PROJECT PERIODIC REPORT

Grant Agreement number: 288956

Project acronym: NADINE

Project title: New tools and Algorithms for Directed Network analysis

Funding Scheme: Small or medium-scale focused research project (STREP)

Periodic report: **1st 2nd X**

Period covered: **from 1.11.2013 to 30.04.2015**

Name, title and organisation of the scientific representative of the project's coordinator[1]:

Dr. Dima Shepelyansky

Directeur de recherche au CNRS

Lab de Phys. Theorique, Universite Paul Sabatier, 31062 Toulouse, France

Tel: +331 5 61556068, Fax: +33 5 61556065, Secr.: +33 5 61557572

E-mail: dima@irsamc.ups-tlse.fr; URL: www.quantware.ups-tlse.fr/dima

Project website address: www.quantware.ups-tlse.fr/FETNADINE/

[1] Usually the contact person of the coordinator as specified in Art. 8.1. of the grant agreement

NADINE DELIVERABLE D5.2.

It is based on milestones M7[WP4.1-WP5.2], M9[WP5.1], M14[WP5.3] with deliverable publications:

[47] P2.10 N.Chen, N.Litvak and M.Olvera-Cravioto, **"PageRank in scale-free random graphs"**, Proceedings 11th International Workshop Algorithms and Models for the Web Graph, WAW 2014, 17-18 Dec 2014, Beijing, China pp. 120-131, Lecture Notes in Computer Science 2014 (8882), Springer (2014). (arXiv:1408.3610[math.PR], 2014 [M7-WP5.2]

[48] P2.11 N.Chen, N.Litvak and M.Olvera-Cravioto, **"Ranking algorithms on directed configuration networks"**, Submitted to Random Structures and Algorithms (2014) (arXiv:1409.7443v2[math.PR], 2014) [M7-WP5.2]

[51] P3.7 Marton Balassi, Robert Palovics and Andras A. Benczur, **"Distributed Frameworks for Alternating Least Squares (Poster presentation)"**, Large-Scale Recommender Systems in conjunction with RecSys, Foster City, Silicon Valley, USA, 6th-10th October 2014 [M7-WP5.2]

[52] P3.8 Balint Daroczy, Krisztian Buza, Andras A. Benczur, **"Similarity Kernel Learning"**, preprint (2015) [M7-WP5.2]

[55] P3.11 R.Palovics, A.A.Benczur, L.Kocsis, T.Kiss, E.Frigo, **"Exploiting temporal influence in online recommendation"**, Proceedings of the 8th ACM Conference on Recommender systems (pp. 273-280), ACM (2015) [M14-WP5.3]

[56] P3.12 Balint Daroczy, David Siklosi, Robert Palovics, Andras A. Benczur, **"Text Classification Kernels for Quality Prediction over the C3 Data Set"**, preprint, WebQuality 2015 in conjunction with WWW 2015 [M14-WP5.3]

[57] P3.13 Frederick Ayala, Robert Palovics, Andras A. Benczur, **"Temporally Evolving Models for Dynamic Networks"**, accepted poster presentation at the International Conference on Computational Social Science, Helsinki, June 2015 [M14-WP5.3]

[58] P3.14 Balint Daroczy, Robert Palovics, Vilmos Wieszner, Richard Farkas, Andras A. Benczur, **"Temporal Twitter prediction by content and network"**, preprint (2015) [M14-WP5.3]

[59] P3.15 Robert Palovics, Andras A. Benczur, **"Modeling Community Growth: densifying graphs or sparsifying subgraphs? "**, preprint (2015) [M14-WP5.3]

[66] P4.13 Sebastiano Vigna, "**Supremum-norm convergence for step-asynchronous successive overrelaxation on M-matrices**", submitted to CoRR (2014) (arXiv:1404.3327[cs.DS], 2014) [M7-WP5.2]

[67] P4.14 Sebastiano Vigna, "**An experimental exploration of Marsaglia's xorshift generators, scrambled**", submitted to CoRR (2014) (arXiv:1402.6246v2[cs.DS], 2014)
[M7-WP5.2]

[68] P4.15 Sebastiano Vigna, "**Further scramblings of Marsaglia's xorshift generators**", submitted to CoRR (2014) (arXiv:1403.0930[cs.NI], 2014) [M7-WP5.2]

[69] P4.16 Young Ho Eom, Pablo Aragon, David Laniado, Andreas Kaltenbrunner, Sebastiano Vigna, and Dima L. Shepelyansky, **"Interactions of cultures and top people of Wikipedia from ranking of 24 language editions"**, PLoS ONE v.10(3), p.e0114825 (2015) (arXiv:1405.7183[cs.SI], 2014) [M13-WP4.3-WP5.2]

[70] P4.17 Sebastiano Vigna, **"A weighted correlation index for rankings with ties"**, Proceedings of the 24th international conference on World Wide Web, ACM (2015) (arXiv:1404.3325[cs.SI], 2014) [M13-WP4.3-WP5.2]

[73] P4.20 Michele Trevisio, Luca Maria Aiello, Paolo Boldi and Roi Blanco, **"Local Ranking Problem on the BrowseGraph"**, accepted for publication in SIGIR (2015) [M13-WP4.3-WP5.2]

PageRank in scale-free random graphs*

Ningyuan Chen[1], Nelly Litvak[2], Mariana Olvera-Cravioto[1]

[1] Columbia University, 500 W. 120th Street, 3rd floor, New York, NY 10027
[2] University of Twente, P.O.Box 217, 7500AE, Enschede, The Netherlands

Abstract. We analyze the distribution of PageRank on a directed configuration model and show that as the size of the graph grows to infinity it can be closely approximated by the PageRank of the root node of an appropriately constructed tree. This tree approximation is in turn related to the solution of a linear stochastic fixed point equation that has been thoroughly studied in the recent literature.

1 Introduction

Google's PageRank proposed by Brin and Page [5] is arguably the most influential technique for computing centrality scores of nodes in a network. Numerous applications include graph clustering [3], spam detection [9], and citation analysis [8, 20]. In this paper we analyze the power law behavior of PageRank scores in scale-free directed random graphs.

In real-world networks, it is often found that the fraction of nodes with (in- or out-) degree k is $\approx c_0 k^{-\alpha-1}$, usually $\alpha \in (1,3)$, see e.g. [17] for an excellent review of the mathematical properties of complex networks. More than ten years ago Pandurangan et al. [16] discovered the interesting fact that PageRank scores also exhibit power laws, with the same exponent as the in-degree. This property holds for a broad class of real-life networks [19]. In fact, the hypothesis that this always holds in power-law networks is plausible.

However, analytical mathematical evidence supporting this hypothesis is surprisingly scarce. As one of the few examples, Avrachenkov and Lebedev [4] obtained the power law behavior of average PageRank scores in a preferential attachment graph by using Polya's urn scheme and advanced numerical methods.

In a series of papers, Volkovich et al. [14, 19, 18] suggested an analytical explanation for the power law behavior of PageRank by comparing it to the endogenous solution of a stochastic fixed point equation (SFPE). The properties of this equation and the study of its multiple solutions has itself been an interesting topic in the recent literature [2, 10, 11, 12, 15, 1], and is related to the broader study of weighted branching processes. The tail behavior of the endogenous solution, the one more closely related to PageRank, was given in [10, 11, 12, 15], where it was shown to have a power law under many different sets of assumptions. However, the SFPE does not fully explain the behavior of PageRank in

* This research is partially funded by the EU-FET Open grant NADINE (288956).

networks since it implicitly assumes that the underlying graph is an infinite tree, an assumption that is not in general satisfied in real-world networks.

This paper makes a fundamental step further by extending the analysis of PageRank to graphs that are not necessarily trees. Specifically, we assume that the underlying graph is a directed configuration model (DCM) with given degree distributions, as developed by Chen and Olvera-Cravioto [7]. We present numerical evidence that in this type of graphs the behavior of PageRank is very close to the one on trees. Intuitively, this is true for two main reasons: 1) the influence of remote nodes on the PageRank of an arbitrary node decreases exponentially fast with the graph distance; and 2) the DCM is asymptotically tree-like, that is, when we explore a graph starting from a given node, then with high probability the first loop is observed at a distance of order $\log n$, where n is the size of the graph (see Figure 1). Our main result establishes analytically that PageRank in a DCM is well approximated by the PageRank of the root node of a suitably constructed tree as the graph size goes to infinity.

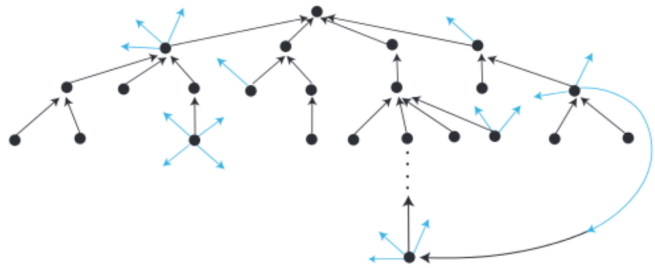


Fig. 1. Graph construction process. Unpaired outbound stubs are in blue.

Section 2 below describes the DCM as presented in [7]. Then, in Section 3 we analytically compare the PageRank scores in the DCM to their approximate value obtained after a finite number of power iterations. Next, in Section 4 we explain how to couple the PageRank of a randomly chosen node with the root node of a suitable branching tree, and give our main analytical results. Finally, in Section 5 we give numerical results validating our analytical work. The complete proofs for more general stochastic recursions, that also cover the PageRank case considered here, will be given in our upcoming paper [6].

2 Directed Random Graphs

We will give below an algorithm, taken from [7], that can be used to generate a scale-free directed graph. Formally, power law distributions are modeled using the mathematical notion of regular variation. A nonnegative random variable X is said to be regularly varying, if $\overline{F}(x) := P(X > x) = L(x)x^{-\alpha}$, $x > 0$, where $L(\cdot)$ is a slowly varying function, that is, $\lim_{x\to\infty} L(tx)/L(x) = 1$, for all $t > 0$.

Our goal now is to create a directed graph $\mathcal{G}(n)$ with the property that the in-degrees and out-degrees will be approximately distributed, for large sizes of

the graph, according to distributions $f_k^{\text{in}} = P(\mathsf{N} = k)$, and $f_k^{\text{out}} = P(\mathsf{D} = k)$, $k = 0, 1, 2, 3, \ldots$, respectively, where $E[\mathsf{N}] = E[\mathsf{D}]$. The only condition needed is that these distributions satisfy

$$\overline{F^{\text{in}}}(x) = \sum_{k>x} f_k^{\text{in}} \le x^{-\alpha} L_{\text{in}}(x) \qquad \text{and} \qquad \overline{F^{\text{out}}}(x) = \sum_{k>x} f_k^{\text{out}} \le x^{-\beta} L_{\text{out}}(x),$$

for some slowly varying functions $L_{\text{in}}(\cdot)$ and $L_{\text{out}}(\cdot)$, and $\alpha, \beta > 1$.

The first step in our procedure is to generate an appropriate bi-degree sequence

$$(\mathsf{N}_n, \mathsf{D}_n) = \{(N_i, D_i) : 1 \le i \le n\}$$

representing the n nodes in the graph. The algorithm given below will ensure that the in- and out-degrees follow closely the desired distributions and also that the sums of in- and out-degrees are the same:

$$L_n := \sum_{i=1}^{n} N_i = \sum_{i=1}^{n} D_i.$$

Denote

$$\kappa_0 = \min\{1 - \alpha^{-1}, 1 - \beta^{-1}, 1/2\}.$$

Algorithm 1. *Generation of a bi-degree sequence with given in-/out-degree distributions.*

1. Fix $0 < \delta_0 < \kappa_0$.
2. Sample an i.i.d. sequence $\{\mathsf{N}_1, \ldots, \mathsf{N}_n\}$ from distribution F^{in}.
3. Sample an i.i.d. sequence $\{\mathsf{D}_1, \ldots, \mathsf{D}_n\}$ from distribution F^{out}, independent of $\{\mathsf{N}_i\}$.
4. Define $\Delta_n = \sum_{i=1}^{n} (\mathsf{N}_n - \mathsf{D}_n)$. If $|\Delta_n| \le n^{1-\kappa_0+\delta_0}$ proceed to step 5; otherwise repeat from step 2.
5. Choose randomly $|\Delta_n|$ nodes $\{i_1, i_2, \ldots, i_{|\Delta_n|}\}$ without replacement and let

$$N_i = \begin{cases} \mathsf{N}_i + 1 & \text{if } \Delta_n < 0 \text{ and } i \in \{i_1, i_2, \ldots, i_{\Delta_n}\}, \\ \mathsf{N}_i & \text{otherwise,} \end{cases}$$

$$D_i = \begin{cases} \mathsf{D}_i + 1 & \text{if } \Delta_n \ge 0 \text{ and } i \in \{i_1, i_2, \ldots, i_{\Delta_n}\}, \\ \mathsf{D}_i & \text{otherwise.} \end{cases}$$

Remark: It was shown in [7] that

$$P\left(|\Delta_n| > n^{1-\kappa_0+\delta_0}\right) = O\left(n^{-\delta_0(\kappa_0-\delta_0)/(1-\kappa_0)}\right) \tag{1}$$

as $n \to \infty$, and therefore the Algorithm 1 will always terminate after a finite number of steps (i.e., it will eventually proceed to step 5).

Having obtained a realization of the bi-degree sequence $(\mathsf{N}_n, \mathsf{D}_n)$, we now use the configuration model to construct the random graph. The idea in the directed case is essentially the same as for undirected graphs. To each node v_i

we assign N_i inbound half-edges and D_i outbound half-edges; then, proceed to match inbound half-edges to outbound half-edges to form directed edges. To be more precise, for each unpaired inbound half-edge of node v_i choose randomly from all the available unpaired outbound half-edges, and if the selected outbound half-edge belongs to node, say, v_j, then add a directed edge from v_j to v_i to the graph; proceed in this way until all unpaired inbound half-edges are matched. Note that the resulting graph is not necessarily simple, i.e., it may contain self-loops and multiple edges in the same direction.

We point out that conditional on the graph being simple, it is uniformly chosen among all simple directed graphs having bi-degree sequence $(\mathsf{N}_n, \mathsf{D}_n)$ (see [7]). Moreover, it was also shown in [7] that, provided $\alpha, \beta > 2$, the probability of obtaining a simple graph through this procedure is bounded away from zero, and therefore one can obtain a simple graph having $(\mathsf{N}_n, \mathsf{D}_n)$ as its bi-degree sequence by simply repeating the algorithm enough times. When we can only ensure that $\alpha, \beta > 1$, then a simple graph can still be obtained without loosing the distributional properties of the in- and out-degrees by erasing the self-loops and merging multiple edges in the same direction. These considerations about the graph being simple are nonetheless irrelevant to the current paper.

3 PageRank iterations in the DCM

Although PageRank can be thought of as the solution to a system of linear equations, we will show in this section how it is sufficient to consider only a finite number of power iterations to obtain an accurate approximation for the PageRank of all the nodes in the graph. We first introduce some notation.

Let $M = M(n) \in \mathsf{R}^{n \times n}$ be matrix constructed as follows:

$$M_{i,j} = \begin{cases} s_{ij}c/D_i, & \text{if there are } s_{ij} \text{ edges from } i \text{ to } j, \\ 0, & \text{otherwise,} \end{cases}$$

and let $\mathbf{1}$ be the row vector of ones. In the classical definition [13], PageRank $\pi = (\pi_1, \ldots, \pi_n)$ is the unique solution to the following equation:

$$\pi = \pi(cM) + \frac{1-c}{n}\mathbf{1}, \tag{2}$$

where $c \in (0,1)$ is a constant known as the damping factor. Rather than analyzing π directly, we consider instead its scale-free version

$$n\pi =: \mathsf{R} = \mathsf{R}\,(cM) + (1-c)\mathbf{1} \tag{3}$$

obtained by multiplying (2) by the size of the graph n. Moreover, whereas π_i is a probability distribution ($\pi_i \geq 0$ for all i and $\pi \mathbf{1}^T = 1$), its scale-free version $\mathsf{R} = (R_1, \ldots, R_n)$ has components that are essentially unbounded for large n and that satisfy $E[R_i] = 1$ for all n and all $1 \leq i \leq n$ (hence the name scale-free).

One way to solve the system of linear equations given in (3) is via power iterations. We define the kth iteration of PageRank on the graph as follows.

First initialize PageRank with a vector $\mathsf{r}_0 = r_0\mathsf{1}$, $r_0 \geq 0$, and then iterate according to $\mathsf{R}^{(n,0)} = \mathsf{r}_0$ and

$$\mathsf{R}^{(n,k)} = \mathsf{R}^{(n,k-1)}M + (1-c)\mathsf{1} = (1-c)\mathsf{1}\sum_{i=0}^{k-1} M^i + \mathsf{r}_0 M^k$$

for $k \geq 1$. In this notation, $\mathsf{R} = \mathsf{R}^{(n,\infty)}$, and our main interest is to analyze the distribution of the PageRank of a randomly chosen node in the DCM, say $R_1^{(n,\infty)}$. The first step of the analysis is to compare $\mathsf{R}^{(n,\infty)}$ to its kth iteration $\mathsf{R}^{(n,k)}$. To this end, note that $\mathsf{R}^{(n,\infty)} = (1-c)\mathsf{1}\sum_{i=0}^{\infty} M^i$, and therefore,

$$\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,\infty)} = \mathsf{r}_0 M^k - (1-c)\mathsf{1}\sum_{i=k}^{\infty} M^i.$$

Moreover,

$$\begin{aligned}\left\|\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,\infty)}\right\|_1 &\leq \left\|\mathsf{r}_0 M^k\right\|_1 + (1-c)\sum_{i=0}^{\infty}\left\|\mathsf{1}M^{k+i}\right\|_1 \\ &\leq r_0 n \left\|M^k\right\|_\infty + (1-c)n\sum_{i=0}^{\infty}\left\|M^{k+i}\right\|_\infty,\end{aligned}$$

where for the last inequality we used the observation that

$$\left\|\mathsf{1}M^r\right\|_1 = \sum_{j=1}^{n}\sum_{i=1}^{n}(M^r)_{ij} = \sum_{i=1}^{n}\left\|(M^r)_{i\bullet}\right\|_1 \leq n\left\|M^r\right\|_\infty,$$

where $A_{i\bullet}$ denotes the ith row of matrix A. Furthermore, since M is equal to c times an adjacency matrix, we have

$$\left\|M^r\right\|_\infty \leq \|M\|_\infty^r = c^r.$$

It follows that

$$\left\|\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,\infty)}\right\|_1 \leq r_0 n c^k + (1-c)n\sum_{i=0}^{\infty} c^{k+i} = (r_0+1)nc^k. \tag{4}$$

In general networks, the inequality for the L_1-norm (4) does not provide information on convergence of specific coordinates and does not give a good upper bound for the quantity $|R_1^{(n,k)} - R_1^{(n,\infty)}|$ that we are interested in. However, the DCM has the additional property that all coordinates of the vector $\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,\infty)}$ have the same distribution, since by construction, the DCM makes all permutations of the nodes' labels equally likely. This leads to the following observation.

Let $\mathcal{F}_n = \sigma((\mathsf{N}_n, \mathsf{D}_n))$ denote the sigma-algebra generated by the bi-degree sequence, which does not include information about the pairing process. Then, conditional on $\mathcal{F}_n$,

$$E\left[\left|R_1^{(n,k)} - R_1^{(n,\infty)}\right| \middle| \mathcal{F}_n\right] = \frac{1}{n}E\left[\left\|\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,\infty)}\right\|_1 \middle| \mathcal{F}_n\right] \leq (r_0+1)\,c^k,$$

and Markov's inequality gives,

$$P\left(\left|R_1^{(n,\infty)} - R_1^{(n,k)}\right| > \epsilon\right) \leq E\left[\epsilon^{-1} E\left[\left|R_1^{(n,k)} - R_1^{(n,\infty)}\right| \middle| \mathcal{F}_n\right]\right]$$
$$\leq (r_0 + 1)\,\epsilon^{-1} c^k \tag{5}$$

for any $\epsilon > 0$.

Note that (5) is a probabilistic statement, which is not completely analogous to (4). In fact, (5) states that we can achieve any level of precision with a pre-specified high probability by simply increasing the number of iterations k. This leads to the following heuristic, that if the DCM looks locally like a tree for k generations, where k is the number of iterations needed to achieve the desired precision in (5), then the PageRank of node 1 in the DCM will be essentially the same as the PageRank of the root node of a suitably constructed tree. The precise result and a sketch of the arguments will be given in the next section.

4 Main Result: Coupling with a thorny branching tree

As mentioned in the previous section, we will now show how to identify $R_1^{(n,k)}$ with the PageRank of the root node of a tree. To start, we construct a variation of a branching tree where each node has an edge pointing to its parent but also has number of outbound stubs or half-edges that are pointing outside of the tree (i.e., to some auxiliary node). We will refer to this tree as a Thorny Branching Tree (TBT), the name "thorny" referring to the outbound stubs (see Figure 1).

To construct simultaneously the graph $\mathcal{G}(n)$ and the TBT, denoted by $\mathcal{T}$, we start by choosing a node uniformly at random, and call it node 1 (the root node). This first node will have N_1 inbound stubs which we will proceed to match with randomly chosen outbound stubs. These outbound stubs are sampled independently and with replacement from all the possible $L_n = \sum_{i=1}^{n} D_i$ outbound stubs, discarding any outbound stub that has already been matched. This corresponds to drawing independently at random from the distribution

$$\begin{aligned} f_n(i,j) &= P(\text{node has } i \text{ offspring, } j \text{ outbound links } |\mathcal{F}_n) \\ &= \sum_{k=1}^{n} 1(N_k = i, D_k = j) P(\text{an outbound stub of node } k \text{ is sampled } |\mathcal{F}_n) \\ &= \sum_{k=1}^{n} 1(N_k = i, D_k = j) \frac{D_k}{L_n}. \end{aligned} \tag{6}$$

This is a so-called size-biased distribution, since nodes with more outbound stubs are more likely to be chosen.

To keep track of which outbound stubs have already been matched we will label them 1, 2, or 3 according to the following rule:

1. Outbound stubs with label 1 are stubs belonging to a node that is not yet attached to the graph.

2. Outbound stubs with label 2 belong to nodes that are already part of the graph but that have not yet been paired with an inbound stub.
3. Outbound stubs with label 3 are those which have already been paired with an inbound stub and now form an edge in the graph.

Let Z_r, $r \geq 0$, denote the number of inbound stubs of all the nodes in the graph at distance r of the first node. Note that $Z_0 = N_1$ and Z_r is also the number of nodes at distance $(r+1)$ of the first node.

To draw the graph we initialize the process by labeling all outbound stubs with a 1, except for the D_1 outbound stubs of node 1 that receive a 2. We then start by pairing the first of the N_1 inbound stubs with a randomly chosen outbound stub, say belonging to node j. Then node j is attached to the graph by forming an edge with node 1, and all the outbound stubs from the new node are now labeled 2. In case that $j = 1$ the pairing forms a self-loop and no new nodes are added to the graph. Next, we label the chosen outbound stub with a 3, since it has already been paired, and in case $j \neq 1$, give all the other outbound stubs of node j a label 2. We continue in this way until all N_1 inbound stubs of node 1 have been paired, after which we will be left with Z_1 unmatched inbound stubs that will determine the nodes at distance 2 from node 1. In general, the kth iteration of this process is completed when all Z_{k-1} inbound stubs have been matched with an outbound stub, and the process ends when all L_n inbound stubs have been paired. Note that whenever an outbound stub with label 2 is chosen a cycle or double edge is formed in the graph. If at any point we sample an outbound stub with label 3 we simply discard it and do a redraw until we obtain an outbound stub with labels 1 or 2.

We now explain the coupling with the TBT. We start with the root node (node 1, generation 0) that has $\hat{N}_1 = N_1$ offspring. Let $\hat{Z}_k$ denote the number of individuals in generation $k+1$ of the tree, $\hat{Z}_0 = \hat{N}_1$. For $k \geq 1$, each of the $\hat{Z}_{k-1}$ individuals in the kth generation will independently have offspring and outbound stubs according to the random joint distribution $f_n(i,j)$ given in (6).

The coupling of the graph and the TBT is done according to the following rules:

1. If an outbound stub with label 1 is chosen, then both the graph and the TBT will connect the chosen outbound stub to the inbound stub being matched, resulting in a node being added to the graph and an offspring being born to its parent. In particular, if the chosen outbound stub corresponds to node j, then the new offspring in the TBT will have $D_j - 1$ outbound stubs (pointing to the auxiliary node) and N_j inbound stubs (number of offspring). We then update the labels by giving a 2 label to all the 'sibling' outbound stubs of the chosen outbound stub, and a 3 label to the chosen outbound stub itself.
2. If an outbound stub with label 2 is sampled it means that its corresponding node already belongs to the graph, and a cycle, self-loop, or multiple edge is created. In $\mathcal{T}$, we proceed as if the outbound stub had label 1 and create a new node, which is a copy of the drawn node. The coupling between DCM and TBT breaks at this point.

3. If an outbound stub with label 3 is drawn it means that this stub has already been matched, and the coupling breaks as well. In $\mathcal{T}$, we again proceed as if the outbound stub had had a label 1. In the graph we do a redraw.

Note that the processes Z_k and $\hat{Z}_k$ are identical as long as the coupling holds. Showing that the coupling holds for a sufficient number of generations is the essence of our main result.

Definition 1 Let τ be the number of generations in the TBT that can be completed before the first outbound stub with label 2 or 3 is drawn, i.e., $\tau = k$ if the first inbound stub to draw an outbound stub with label 2 or 3 belonged to a node i, such that the graph distance between i and the root node is exactly k.

The following result gives us an estimate as to when the coupling between the exploration process of the graph and the construction of the tree is expected to break.

Lemma 1. *Suppose $(\mathsf{N}_n, \mathsf{D}_n)$ are constructed using Algorithm 1 with $\alpha > 1$, and $\beta > 2$. Let $\mu = E[\mathsf{N}] = E[\mathsf{D}] > 1$. Then, for any $1 \le k \le h \log n$ with $0 < h < 1/(2\log\mu)$ there exists a $\delta > 0$ such that,*

$$P(\tau \le k) = O\left(n^{-\delta}\right) \qquad \text{as } n \to \infty.$$

The proof of Lemma 1 is rather technical, so we will only provide a sketch in this paper. The detailed proof will be given in [6].

Proof (Qualitative argument). Let $\hat{V}_s$ be the number of outbound stubs of all nodes in generation s of the tree. The intuition behind the proof is that for all $s = 1, 2, \ldots$, neither $\hat{Z}_s$, nor $\hat{V}_s$ are expected to be much larger than their means:

$$E\left[\hat{Z}_s \middle| \mathcal{F}_n\right] \approx \mu^{s+1} \qquad \text{and} \qquad E\left[\hat{V}_s \middle| \mathcal{F}_n\right] \approx \lambda\mu^s,$$

where $\lambda = E[\mathsf{D}^2]/\mu$. Next, note that an inbound stub of a node in the rth generation will be the first one to be paired with an outbound stub having label 2 or 3 with a probability bounded from above by

$$P_r := \frac{1}{L_n} \sum_{s=0}^{r} \hat{V}_s \approx \frac{\lambda\mu^r}{n(\mu-1)}.$$

Furthermore, for event $\{\tau = r\}$ to occur one of the $\hat{Z}_r$ inbound stubs must have been paired with an outbound stub with labels 2 or 3, which is bounded by the probability that a Binomial random variable with parameters $(\hat{Z}_r, P_r)$ is greater or equal than 1. Since $P_r = o(1)$ for $r \le k$, this probability is $1 - (1-P_r)^{\hat{Z}_r} = P_r\hat{Z}_r(1 + O(P_r)) = O\left(\mu^{2r}n^{-1}\right)$.

Formally, to ensure that the approximations given above are valid, we first show that the event

$$E_k = \left(\max_{0\le r\le k} \mu^{-r}\hat{Z}_r\mu^r \le x_n, \max_{0\le r\le k} \mu^{-(r+1)} \sum_{s=0}^{r} \hat{V}_s \le x_n \right)$$

occurs with high probability as $n \to \infty$ for a suitably chosen $x_n \to \infty$. Then, sum over $r = 0, 1, \ldots, k$ the events $\{\tau = r, E_k\}$ to obtain that $P(\tau \leq k, E_k) = O\left(\mu^{2k} n^{-1}\right)$, which goes to zero for $k \leq h \log n$.

Our main result is now a direct consequence of the bound derived in (5) and Lemma 1 above, since before the coupling breaks $R_1^{(n,k)}$ and the PageRank, computed after k iterations, of the root node of the coupled tree coincide.

Theorem 1. *Suppose $(\mathsf{N}_n, \mathsf{D}_n)$ are constructed using Algorithm 1 with $\alpha > 1$, and $\beta > 2$. Let $\mu = E[\mathsf{N}\,] = E[\mathsf{D}] > 1$ and $c \in (0,1)$. Then, for any $\epsilon > 0$ and any $1 \leq k \leq h \log n$ with $0 < h < 1/(2 \log \mu)$ there exists a $\delta > 0$ such that,*

$$P\left(\left|R_1^{(n,\infty)} - \hat{R}_1^{(n,k)}\right| > \epsilon\right) \leq (r_0 + 1)\epsilon^{-1} c^k + O\left(n^{-\delta}\right),$$

as $n \to \infty$, where $\hat{R}_1^{(n,k)}$ is the PageRank, after k iterations, of the root node of the TBT described above.

In the forthcoming paper [6] we explore further the distribution of the PageRank of the root node of $\mathcal{T}$ and show that $\hat{R}_1^{(n,k)}$ converges to the endogenous solution of a SFPE on a weighted branching tree, as originally suggested in [14, 19, 18]. Moreover, the tail behavior of this solution has been fully described in [18, 10, 11].

5 Numerical Results

In this last section we give some numerical results showing the accuracy of the TBT approximation to the PageRank in the DCM. To generate the bi-degree sequence we use as target distributions two Pareto-like distributions. More precisely, we set

$$\mathsf{N}_i = \lfloor X_{1,i} + Y_{1,i} \rfloor, \quad \mathsf{D}_i = \lfloor X_{2,i} + Y_{2,i} \rfloor,$$

where the $\{X_{1,i}\}$ and the $\{X_{2,i}\}$ are independent sequences of i.i.d. Pareto random variables with shape parameters $\alpha > 1$ and $\beta > 2$, respectively, and scale parameters $x_1 = (\alpha - 1)/\alpha$ and $x_2 = (\beta - 1)/\beta$, respectively (note that $E[X_{1,i}] = E[X_{2,i}] = 1$ for all i). The sequences $\{Y_{1,i}\}$ and $\{Y_{2,i}\}$ are independent sequences, each consisting of i.i.d. exponential random variables with means $1/\lambda_1 > 0$ and $1/\lambda_2$, respectively. The addition of the exponential random variables allows more flexibility in the modeling of the in- and out-degree distributions while preserving a power law tail behavior; the parameters λ_1, λ_2 are also used to match the means $E[\mathsf{N}\,]$ and $E[\mathsf{D}]$.

Once the sequences $\{\mathsf{N}_i\}$ and $\{\mathsf{D}_i\}$ are generated, we use Algorithm 1 to obtain a valid bi-degree sequence $(\mathsf{N}_n, \mathsf{D}_n)$. Given this bi-degree sequence we next proceed to construct the graph and the TBT simultaneously, according to the rules described in Section 4. To compute $\mathsf{R}^{(n,\infty)}$ we perform power iterations with $r_0 = 1$ until $\|\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,k-1)}\|_2 < \epsilon_0$ for some tolerance ϵ_0. We only

generate the TBT for the required number of generations in each of the examples; the computation of $\hat{R}_1^{(n,k)}$ can be done recursively starting from the leaves using

$$\hat{R}_i^{(n,0)} = 1, \quad \hat{R}_i^{(n,k)} = \sum_{j \to i} c\hat{R}_j^{(n,k-1)} + (1-c), \quad k > 0, \tag{7}$$

where $j \to i$ means that node j is an offspring of node i.

Tables 1-3 below compare the PageRank of node 1 in the graph, $R_1^{(n,\infty)}$, the PageRank of node 1 only after k power iterations, $R_1^{(n,k)}$, and the PageRank of the root node of the coupled tree after the same k generations, $\hat{R}_1^{(n,k)}$. The magnitude of the mean squared errors (MSEs), computed using $R_1^{(n,\infty)}$ as the true value, is also given in each table. The tolerance for computing $R_1^{(n,\infty)}$ is set to $\varepsilon_0 = 10^{-6}$. For each n, we generate 100 realizations of $\mathcal{G}(n)$ as well as of the corresponding TBTs and take the empirical average of the PageRank values and of the MSEs. Table 1 includes results for different sizes of the graph, and uses $k_n = \lfloor \log n \rfloor$ iterations for the finite approximations. We note that all the MSEs clearly decrease as n increases since k_n also increases with n.

n	$R_1^{(n,1)}$	$R_1^{(n,k_n)}$	$\hat{R}_1^{(n,k_n)}$	MSE for $R_1^{(n,k_n)}$	MSE for $\hat{R}_1^{(n,k_n)}$
10	0.931	0.946	0.983	3.90E-03	4.20E-02
100	1.023	1.027	1.068	1.80E-04	3.70E-02
1000	1.000	1.002	1.010	1.20E-05	8.00E-04
10000	0.964	0.965	0.962	1.00E-06	7.50E-04

Table 1. $\delta = 2$, $\bullet = 2{:}5$, $\blacklozenge_1 = 1$, $c = 0{:}5$, $k_n = \lfloor \log n \rfloor$.

Table 2 illustrates the impact of using different values of k, with the error between $R_1^{(n,k)}$ and $R_1^{(n,\infty)}$ clearly decreasing as k increases. The simulations were run on a graph with $n = 10,000$ nodes. We also point out that although the accuracy of finitely many PageRank iterations improves as k gets larger, the MSE of the tree approximation seems to plateau after a certain point. In order to obtain a higher level of precision we also need to increase the size of the graph (as suggested by Theorem 1).

k_n	$R_1^{(n,1)}$	$R_1^{(n,k_n)}$	$\hat{R}_1^{(n,k_n)}$	MSE for $R_1^{(n,k_n)}$	MSE for $\hat{R}_1^{(n,k_n)}$
2	0.908	0.933	0.928	7.1E-03	8.59E-03
4	0.929	0.933	0.933	1.5E-04	2.20E-04
6	0.908	0.909	0.910	5.4E-06	5.08E-05
8	0.883	0.884	0.884	8.8E-08	1.20E-06
10	0.948	0.949	0.950	7.6E-09	8.16E-05
15	0.932	0.932	0.932	7.9E-13	2.89E-05

Table 2. $n = 10000$, $\delta = 2$, $\bullet = 2{:}5$, $\blacklozenge_1 = 1$, $c = 0{:}5$.

Table 3 shows the same comparison as in Table 2, for fixed n, for different values of the damping factor c. As c gets larger, the approximations provided by both $R_1^{(n,k_n)}$ and $\hat{R}_1^{(n,k_n)}$ get worse due to the slower convergence of PageRank.

c	$R_1^{(n,1)}$	$R_1^{(n,k_n)}$	$\hat{R}_1^{(n,k_n)}$	MSE for $R_1^{(n,k_n)}$	MSE for $\hat{R}_1^{(n,k_n)}$
0.1	1.011	1.011	1.011	3.8E-22	3.33E-09
0.3	0.958	0.958	0.958	9.8E-13	1.91E-07
0.5	0.898	0.898	0.899	2.7E-08	2.63E-06
0.7	0.755	0.757	0.760	2.4E-05	2.03E-04
0.9	0.663	0.764	0.799	8.3E-02	1.25E-01

Table 3. n = 10000, = 2, = 2:5, = 1, k_n = blog nc = 9.

Our last numerical result shows how the distribution of PageRank on the TBT approximates the distribution of PageRank on the DCM. To illustrate this we generated a graph with $n = 100$ nodes and parameters $\alpha = 2$, $\beta = 2.5$, $\mu = 3$ and $c = 0.5$. We set the number of PageRank iterations (number of generations in the TBT) to be $k = 4$. We then computed the empirical CDFs of the PageRank of all nodes in the graph and that of the PageRank after only k iterations. We also generated the coupled TBT 1000 times based on the same graph; each time by randomly choosing some node i to be the root and computing $\hat{R}_i^{(n,k)}$ according to (7). Figure 2 plots the empirical CDF of PagerRank on $\mathcal{G}(n)$, the empirical CDF of PageRank on $\mathcal{G}(n)$ after only k iterations, and the empirical CDF of the PageRank of the 1000 root nodes after the same k iterations. We can see that the CDFs of PageRank on $\mathcal{G}(n)$ after a finite number of iterations and that of the true PageRank on $\mathcal{G}(n)$ are almost indistinguishable. The PageRank on the TBT also approximates this distribution quite well, especially considering that $n = 100$ is not particularly large.

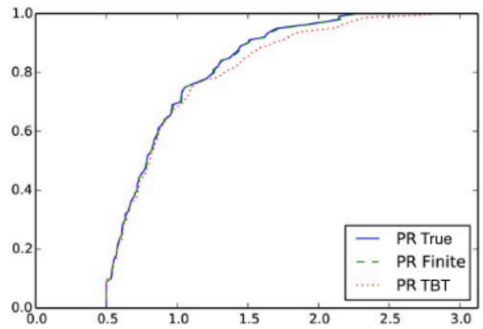


Fig. 2. The empirical distributions of PageRank on G(n) (true and after finitely many iterations) and the empirical distribution of the PageRank of the root in the TBT.

Bibliography

[1] G. Alsmeyer, E. Damek, and S. Mentemeier. Tails of fixed points of the two-sided smoothing transform. In Springer Proceedings in Mathematics & Statistics: Random Matrices and Iterated Random Functions, 2012.

[2] G. Alsmeyer and M. Meiners. Fixed points of the smoothing transform: Two-sided solutions. Probab. Theory Relat. Fields, 155(1-2):165–199, 2013.

[3] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using PageRank vectors. In Proceedings of FOCS2006, pages 475–486, 2006.

[4] K. Avrachenkov and D. Lebedev. PageRank of scale-free growing networks. Internet Mathematics, 3(2):207–231, 2006.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 33:107–117, 1998.

[6] N. Chen, N. Litvak, and M. Olvera-Cravioto. Ranking algorithms on directed configuration networks. Technical report, 2014.

[7] N. Chen and M. Olvera-Cravioto. Directed random graphs with given degree distributions. Stochastic Systems, 3(1):147–186, 2013.

[8] P. Chen, H. Xie, S. Maslov, and S. Redner. Finding scientific gems with Googles PageRank algorithm. Journal of Informetrics, 1(1):8–15, 2007.

[9] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating Web spam with TrustRank. In Proceeding of VLDB2004, pages 576–587, 2004.

[10] P.R. Jelenković and M. Olvera-Cravioto. Information ranking and power laws on trees. Adv. Appl. Prob., 42(4):1057–1093, 2010.

[11] P.R. Jelenković and M. Olvera-Cravioto. Implicit renewal theory and power tails on trees. Adv. Appl. Prob., 44(2):528–561, 2012.

[12] P.R. Jelenković and M. Olvera-Cravioto. Implicit renewal theory for trees with general weights. Stochastic Process. Appl., 122(9):3209–3238, 2012.

[13] A.N. Langville and C.D. Meyer. Google PageRank and beyond. Princeton University Press, 2006.

[14] N. Litvak, W.R.W. Scheinhardt, and Y. Volkovich. In-degree and PageRank: Why do they follow similar power laws? Internet mathematics, 4(2):175–198, 2007.

[15] M. Olvera-Cravioto. Tail behavior of solutions of linear recursions on trees. Stochastic Process. Appl., 122(4):1777–1807, 2012.

[16] G. Pandurangan, P. Raghavan, and E. Upfal. Using PageRank to characterize Web structure. Internet Mathematics, 3(1):1–20, 2006.

[17] R. van der Hofstad. Random graphs and complex networks, 2009.

[18] Y. Volkovich and N. Litvak. Asymptotic analysis for personalized web search. Adv. Appl. Prob., 42(2):577–604, 2010.

[19] Y. Volkovich, N. Litvak, and D. Donato. Determining factors behind the pagerank log-log plot. In Proceedings of the 5th International Conference on Algorithms and Models for the Web-graph, pages 108–123, 2007.

[20] L. Waltman and N.J. van Eck. The relation between eigenfactor, audience factor, and influence weight. J. Am. Soc. Inf. Sci., 61(7):1476–1486, 2010.

Ranking algorithms on directed configuration networks

Ningyuan Chen
Columbia University

Nelly Litvak
University of Twente

Mariana Olvera-Cravioto
Columbia University

October 14, 2014

Abstract

This paper studies the distribution of a family of rankings, which includes Google's PageRank, on a directed configuration model. In particular, it is shown that the distribution of the rank of a randomly chosen node in the graph converges in distribution to a finite random variable $\mathcal{R}^*$ that can be written as a linear combination of i.i.d. copies of the endogenous solution to a stochastic fixed point equation of the form

$$\mathcal{R} \stackrel{\mathcal{D}}{=} \sum_{i=1}^{\mathcal{N}} \mathcal{C}_i \mathcal{R}_i + \mathcal{Q},$$

where $(\mathcal{Q}, \mathcal{N}, \{\mathcal{C}_i\})$ is a real-valued vector with $\mathcal{N} \in \{0, 1, 2, \dots\}$, $P(|\mathcal{Q}| > 0) > 0$, and the $\{\mathcal{R}_i\}$ are i.i.d. copies of $\mathcal{R}$, independent of $(\mathcal{Q}, \mathcal{N}, \{\mathcal{C}_i\})$. Moreover, we provide precise asymptotics for the limit $\mathcal{R}^*$, which when the in-degree distribution in the directed configuration model has a power law imply a power law distribution for $\mathcal{R}^*$ with the same exponent.

Kewywords: PageRank, ranking algorithms, directed configuration model, complex networks, stochastic fixed-point equations, weighted branching processes, power laws.
2000 MSC: Primary: 05C80, 60J80, 68P20. Secondary: 41A60, 37A30, 60B10.

1 Introduction

Ranking of nodes according to their centrality, or importance, in a complex network such as the Internet, the World Wide Web, and other social and biological networks, has been a hot research topic for several years in physics, mathematics, and computer science. For a comprehensive overview of the vast literature on rankings in networks we refer the reader to [27], and more recently to [7] for a thorough up-to-date mathematical classification of centrality measures.

In this paper we analyze a family of ranking algorithms which includes Google's PageRank, the algorithm proposed by Brin and Page [10], and which is arguably the most influential technique for computing rankings of nodes in large directed networks. The original definition of PageRank is the following. Let $\mathcal{G}_n = (V_n, E_n)$ be a directed graph, with a set of (numbered) vertices $V_n = \{1, \dots, n\}$, and a set of directed edges E_n. Choose a constant $c \in (0, 1)$, which is called a *damping factor*, and let $\mathbf{q} = (q_1, q_2, \dots, q_n)$ be a *personalization* probability vector, i.e., $q_i \geq 0$ and $\sum_{i=1}^n q_i = 1$. Denote

by $d_i = |\{j : (i,j) \in E_n\}|$ the out-degree of node $i \in V_n$. Then the PageRank vector $\mathbf{r} = (r_1, \ldots, r_n)$ is the unique solution to the following system of linear equations:

$$r_i = \sum_{j:(j,i)\in E_n} \frac{c}{d_j}\, r_j + (1-c)q_i, \quad i = 1, \ldots, n. \tag{1.1}$$

Google's PageRank was designed to rank Web pages based on the network's structure, rather than their content. The idea behind (1.1) is that a page is important if many important pages have a hyperlink to it. Furthermore, by tuning the personalization values, q_i's, one can, for instance, give preference to specific topics [20] or penalize spam pages [19].

In the original definition, $\mathbf{r}$ is normalized so that $||\mathbf{r}||_1 = 1$, where the norm $||\mathbf{x}||_1 = \sum_{i=1}^n |x_i|$ denotes the l_1 norm in $\mathbb{R}^n$. Since the average PageRank in $\mathbf{r}$ scales as $O(1/n)$, it is more convenient for our purposes to work with a scaled version of PageRank:

$$n\mathbf{r} =: \mathbf{R} = (R_1, R_2, \ldots, R_n).$$

Then, also using the notation C_j for c/d_j, and notation Q_i for $n(1-c)q_i$, we rewrite (1.1) to obtain

$$R_i = \sum_{j:(j,i)\in E_n} C_j\, R_j + Q_i, \quad i = 1, \ldots, n. \tag{1.2}$$

Throughout the paper, we will refer to $\mathbf{R}$ as the PageRank vector and to $\mathbf{Q} = (Q_1, Q_2, \ldots, Q_n)$ as the personalization vector.

The basic definition (1.1) has many modifications and generalizations. The analysis in this paper will cover a wide range of them by allowing a general form of the coefficients in (1.2). For example, our model admits a random damping factor as studied in [15]. Numerous applications of PageRank and its modifications include graph clustering [5], spam detection [19], and citation analysis [13, 43].

In real-world networks, it is often found that the fraction of nodes with (in- or out-) degree k is $\approx c_0 k^{-\alpha-1}$, usually $\alpha \in (1,3)$, see e.g., [10, 29]. Thus, a lot of research has been devoted to the study of random graph models with highly skewed, or scale-free, degree distributions. By now, classical examples are the Chung-Lu model [14], the Preferential Attachment model [9], and the Configuration Model [35, Chapter 7]. New models continue to appear, tuned to the properties of specific networks. For example, an interesting "super-star" model was recently developed to describe retweet graphs [6]. We refer to [35, 17, 29] for a more detailed discussion of random graph models for complex networks. In this paper we focus on the Directed Configuration Model as studied in [11]. Originally, an (undirected) Configuration Model is defined as a graph, randomly sampled from the set of graphs with a given degree sequence [8]. We emphasize that, to the best of our knowledge, [11] is the only paper that formally addresses the directed version of the Configuration Model and obtains its exact mathematical properties. We will provide more details in Section 3.

From the work of Pandurangan et al. [31], and many papers that followed, the following hypothesis has always been confirmed by the data.

The power law hypothesis: *If the in-degree distribution in a network follows a power law then the PageRank scores in this network will also follow a power law with the same exponent.*

The power law hypothesis is plausible because in (1.1) the number of terms in the summation on the right-hand side is just the in-degree of i, so the in-degree provides a 'mean-field' approximation

for PageRank [18]. However, this argument is not exact nor accurate enough, which is confirmed by the fact that the top-ranked nodes in PageRank are not exactly those with the largest in-degrees [13, 42, 38]. Exact mathematical evidence supporting the power law hypothesis is surprisingly scarce. As one of the few examples, [26] obtains the power law behavior of average PageRank scores in a preferential attachment graph by using Polya's urn scheme and advanced numerical methods.

In a series of papers, Volkovich et al. [28, 41, 40] suggested an analytical explanation for the power law behavior of PageRank by comparing the PageRank of a randomly chosen node to the endogenous solution of a stochastic fixed point equation (SFPE) that mimics (1.2):

$$R \stackrel{\mathcal{D}}{=} \sum_{i=1}^{N} C_i R_i + Q. \tag{1.3}$$

Here N (in-degree) is a nonnegative integer random variable having a power law distribution with exponent α, Q (personalization) is an arbitrary positive random variable, and the C_i's are random coefficients that in [40] equal c/D_i, with D_i being the out-degree of a node provided $D_i \geq 1$. The symbol $\stackrel{\mathcal{D}}{=}$ denotes equality in distribution. Assuming that N is regularly varying and using Laplace transforms, it was proved in [40] that R has a power law with the same exponent as N if N has a heavier tail than Q, whereas the tail of R is determined by Q if it is heavier than N. The same result was also proved independently in [22] using a sample-path approach.

The properties of equation (1.3) and the study of its multiple solutions has itself been an interesting topic in the recent literature [4, 22, 24, 23, 30, 2], and is related to the broader study of weighted branching processes (WBPs) [32, 33, 34]. The tail behavior of the endogenous solution, the one relevant to PageRank, was given in [22, 24, 23, 30]. In particular, in [22] it was discovered that when the C_i's are not bounded by one and there exists a positive root to the equation $E\left[\sum_{i=1}^{N} |C_i|^{\alpha}\right] = 1$ with $0 < E\left[\sum_{i=1}^{N} |C_i|^{\alpha} \log |C_i|\right] < \infty$, then R will have a power law tail with exponent α; the main tool for this type of analysis is the implicit renewal theory on trees developed there and later extended in [24, 23] to study (1.3) in its full generality.

However, the SFPE does not fully explain the behavior of PageRank in networks since it implicitly assumes that the underlying graph is an infinite tree, a condition that is never true in real-world networks. In this work we complete the argument when the underlying network is a Directed Configuration Model by showing that the distribution of the PageRank in the graph converges to the endogenous solution of a SFPE. Our techniques are likely to be useful in the analysis of PageRank in other locally tree-like graphs.

The essential theoretical contribution of this work is two-fold. First, we prove that the PageRank in the Directed Configuration Model is well approximated by the endogenous solution to a specific SFPE of the same type as (1.3). Second, we develop a methodology to analyze processes on graphs based on a coupling with a new type of stochastic process: a *weighted* branching process. Due to the presence of weights, couplings with weighted branching processes are more complex compared to traditional couplings with standard branching processes, and therefore, our approach may be of independent interest.

In Section 2 we describe our main results, outline the methodology, and provide an overview of the rest of the paper.

2 Overview of the paper

Although a rigorous presentation of the main result in the paper requires a significant amount of notation, we provide here a somewhat imprecise version that still captures the essence of our work. The paper is written according to the different steps needed in the proof of the main result, outlined in Section 2.2, and the precise statement can be found in Section 6.2.

2.1 An overview of the main result

Let $\mathcal{G}_n = (V_n, E_n)$ be a directed graph. We number the nodes $V_n = \{1, 2, \ldots, n\}$ in an arbitrary fashion and let $R_1 =: R_1^{(n)}$ denote the PageRank of node 1, as defined by (1.2). The in-degree of node 1 is then a random variable N_1 picked uniformly at random from the in-degrees of all n nodes in the graph (i.e., from the empirical distribution). Next, we use the notation N_{i+1} to denote the in-degree of the ith inbound neighbor of node 1 (i.e., $(i+1, 1) \in E_n$), and note that although the $\{N_i\}_{i\geq 2}$ have the same distribution, it is not necessarily the same of N_1 since their corresponding nodes implicitly have one or more out-degrees. More precisely, the distribution of the $\{N_i\}_{i\geq 2}$ is an empirical *size-biased* distribution where nodes with high out-degrees are more likely to be chosen. The two distributions can be significantly different when the number of dangling nodes (nodes with zero out-degrees) is a positive fraction of n and their in-degree distribution is different than that of nodes with one or more out-degrees. Similarly, let Q_1 and $\{Q_i\}_{i\geq 2}$ denote the personalization values of node 1 and of its neighbors, respectively, and let $\{C_i\}_{i\geq 2}$ denote the coefficients, or weights, of the neighbors.

As already mentioned, we will assume throughout the paper that $\mathcal{G}_n$ is constructed according to the Directed Configuration Model (DCM). To briefly explain the construction of the DCM consider a bi-degree sequence $(\mathbf{N}_n, \mathbf{D}_n) = \{(N_i, D_i) : 1 \leq i \leq n\}$ of nonnegative integers satisfying $\sum_{i=1}^n N_i = \sum_{i=1}^n D_i$. To draw the graph think of each node, say node i, as having N_i inbound and D_i outbound half-edges or stubs, then pair each of its inbound stubs with a randomly chosen outbound stub from the set of unpaired outbound stubs (see Section 3 for more details). The resulting graph is in general what is called a multigraph, i.e., it can have self-loops and multiple edges in the same direction.

Our main result requires us to make some assumptions on the bi-degree sequence used to construct the DCM, as well as on the coefficients $\{C_i\}$ and the personalization values $\{Q_i\}$, which we will refer to as the extended bi-degree sequence. The first set of assumptions (see Assumption 5.1) requires the existence of certain limits in the spirit of the weak law of large numbers, including $\frac{1}{n}\sum_{i=1}^n D_i^2$ to be bounded in probability (which essentially imposes a finite variance on the out-degrees). This first assumption will ensure the local tree-like structure of the graph. The second set of assumptions (see Assumption 6.2) requires the convergence of certain empirical distributions, derived from the extended bi-degree sequence, to proper limits as the graph size goes to infinity. This type of weak convergence assumption is typical in the analysis of random graphs [35]. We point out that the two sets of assumptions mentioned above are rather weak, and therefore our result is very general. Moreover, as an example, we provide in Section 7 an algorithm to generate an extended bi-degree sequence from a set of prescribed distributions that satisfies both assumptions.

To state our main result let $(\mathcal{N}_0, \mathcal{Q}_0)$ and $(\mathcal{N}, \mathcal{Q}, \mathcal{C})$ denote the weak limits of the joint random

distributions of (N_1, Q_1) and (N_2, Q_2, C_2), respectively, as defined in Assumption 6.2. Let $\mathcal{R}$ denote the endogenous solution to the following SFPE:

$$\mathcal{R} \stackrel{\mathcal{D}}{=} \sum_{j=1}^{\mathcal{N}} \mathcal{C}_j \mathcal{R}_j + \mathcal{Q}, \tag{2.1}$$

where $\{\mathcal{R}_i\}$ are i.i.d. copies of $\mathcal{R}$, independent of $(\mathcal{N}, \mathcal{Q}, \{\mathcal{C}_i\})$, and with $\{\mathcal{C}_i\}$ i.i.d. and independent of $(\mathcal{N}, \mathcal{Q})$. Our main result establishes that under the assumptions mentioned above, we have that

$$R_1^{(n)} \Rightarrow \mathcal{R}^*, \qquad n \to \infty,$$

where $\Rightarrow$ denotes weak convergence and $\mathcal{R}^*$ is given by

$$\mathcal{R}^* := \sum_{j=1}^{\mathcal{N}_0} \mathcal{C}_j \mathcal{R}_j + \mathcal{Q}_0, \tag{2.2}$$

where the $\{\mathcal{R}_i\}$ are again i.i.d. copies of $\mathcal{R}$, independent of $(\mathcal{N}_0, \mathcal{Q}_0, \{\mathcal{C}_i\})$, and with $\{\mathcal{C}_i\}$ independent of $(\mathcal{N}_0, \mathcal{Q}_0)$. Thus, $R_1^{(n)}$ is well approximated by a linear combination of endogenous solutions of a SFPE. Here $\mathcal{R}^*$ represents the PageRank of node 1, and the $\mathcal{R}_i$'s represent the PageRank of its inbound neighbors. We give more details on the explicit construction of $\mathcal{R}$ and comment on why it is called the "endogenous" solution in Section 6. Furthermore, since $\mathcal{R}$ has been thoroughly studied in the weighted branching processes literature, we can establish the power law behavior of PageRank in a wide class of DCM graphs.

2.2 Methodology

As mentioned earlier, the proof of our main result is given in several steps, each of them requiring a very different type of analysis. For the convenience of the reader, we include in this section a map of these steps.

We start in Section 3 by describing the DCM, which on its own does not require any assumptions on the bi-degree sequence. Then, in Section 4 we define a class of ranking algorithms, of which PageRank and its various modifications are special cases. These algorithms produce a vector $\mathbf{R}^{(n)}$ that is a solution to a linear system of equations, where the coefficients are the *weights* $\{C_i\}$ assigned to the nodes. For example, in the classical PageRank scenario, we have $C_i = c/D_i$, if $D_i \neq 0$.

The proof of the main result consists of the following three steps:

1. *Finite approximation* (Section 4.2). Show that the class of rankings that we study can be approximated in the DCM with any given accuracy by a finite (independent of the graph size n) number of matrix iterations. The DCM plays a crucial role in this step since it implies that the ranks of all the nodes in the graph have the same distribution. A uniform bound on the sequence $\{C_i D_i\}$ is required to provide a suitable rate of convergence.

2. *Coupling with a tree* (Section 5). Construct a coupling of the DCM graph and a "thorny branching tree" (TBT). In a TBT each node with the exception of the root has one outbound link to its parent and possibly several other unpaired outbound links. During the construction,

all nodes in both the graph and the tree are also assigned a weight C_i. The main result in this section is the Coupling Lemma 5.4, which states that the coupling between the graph and the tree will hold for a number of generations in the tree that is logarithmic in n. The locally tree-like property of the DCM and our first set of assumptions (Assumption 5.1) on the bi-degree sequence are important for this step.

3. *Convergence to a weighted branching process* (Section 6). Show that the rank of the root node of the TBT converges weakly to (2.2). This last step requires the weak convergence of the random distributions that define the TBT in the previous step (Assumption 6.2).

Finally, Section 7 gives an algorithm to construct an extended bi-degree sequence satisfying the two main assumptions. The technical proofs are postponed to Section 8.

3 The directed configuration model

The Configuration Model (CM) was originally defined as an undirected graph sampled uniformly at random from the collection of graphs with a given degree sequence [8]. In order to ensure a desired degree distribution, one may generate an i.i.d. degree sequence sampled from this distribution, see [35, Section 7.6]. In this case each node receives a random number of half-edges, or stubs, and then the stubs are paired uniformly at random. The resulting graph is, in general, a multi-graph, because two stubs of the same node may form an edge (self-loop), or a node may have two or more stubs connected to the same other node (multiple edges). There are two ways to create a simple graph. In the *repeated* CM, the pairing is repeated until a simple graph is obtained. This will occur with positive probability if the degrees have finite variance, see [35, Section 7.6.]. In the *erased* CM self-loops and double-edges are removed. In the erased CM, the degree sequence is altered because of edge removal, but the distribution of the original degree sequence is preserved asymptotically under very general conditions, see again [35, Section 7.6]. A literature review and discussion of the undirected CM is provided in [35, Section 7.9].

While the undirected CM has been thoroughly studied, a formal analysis of the Directed Configuration Model (DCM) with given in- and out-degree distributions has only been recently presented by Chen and Olvera-Cravioto [11]. The crucial difference compared to the undirected case is that now we have a *bi-degree* sequence, i.e., a pair of sequences of nonnegative integers determining the in- and out-degrees of the nodes. Note that the sums of the in-degrees must be equal to that of the out-degrees for one to be able to draw a graph. The difficulty and originality of the DCM is that sums of i.i.d. in- and out-degrees will only be equal with a probability converging to zero as the size of the graph grows. To circumvent this problem, the algorithm given in [11], and included in Section 7 in this paper, forces the sums to match by adding the necessary half-edges in such a way that the degree distributions are essentially unchanged.

In order to analyze the distribution of ranking scores on the DCM we also need other node attributes besides the in- and out-degrees, such as the coefficients and the personalization values. With this in mind we give the following definition.

Definition 3.1 *We say that the sequence* $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n) = \{(N_i, D_i, C_i, Q_i) : 1 \leq i \leq n\}$ *is an* extended bi-degree sequence *if for all* $1 \leq i \leq n$ *it satisfies* $N_i, D_i \in \mathbb{N} = \{0, 1, 2, 3, \dots\}$, $Q_i, C_i \in \mathbb{R}$,

and is such that

$$L_n := \sum_{i=1}^{n} N_i = \sum_{i=1}^{n} D_i.$$

In this case, we call $(\mathbf{N}_n, \mathbf{D}_n)$ *a* bi-degree sequence.

Formally, the DCM can be defined as follows.

Definition 3.2 *Let* $(\mathbf{N}_n, \mathbf{D}_n)$ *be a bi-degree sequence and let* $V_n = \{1, 2, \ldots, n\}$ *denote the nodes in the graph. To each node* i *assign* N_i *inbound half-edges and* D_i *outbound half-edges. Enumerate all* L_n *inbound half-edges, respectively outbound half-edges, with the numbers* $\{1, 2, \ldots, L_n\}$*, and let* $\mathbf{x}_n = (x_1, x_2, \ldots, x_{L_n})$ *be a random permutation of these* L_n *numbers, chosen uniformly at random from the possible* $L_n!$ *permutations. The DCM with bi-degree sequence* $(\mathbf{N}_n, \mathbf{D}_n)$ *is the directed graph* $\mathcal{G}_n = (V_n, E_n)$ *obtained by pairing the* x_i*th outbound half-edge with the* i*th inbound half-edge.*

We point out that instead of generating the permutation $\mathbf{x}_n$ of the outbound half-edges up front, one could alternatively construct the graph in a breadth-first fashion, by pairing each of the inbound half-edges, one at a time, with an outbound half-edge, randomly chosen with equal probability from the set of unpaired outbound half-edges. In Section 5 we will follow this approach while simultaneously constructing a coupled TBT.

We emphasize that the DCM is, in general, a multi-graph. It was shown in [11] that the random pairing of inbound and outbound half-edges results in a simple graph with positive probability provided both the in-degree and out-degree distributions possess a finite variance. In this case, one can obtain a simple realization after finitely many attempts, a method we refer to as the *repeated* DCM, and this realization will be chosen uniformly at random from all simple directed graphs with the given bi-degree sequence. Furthermore, if the self-loops and multiple edges in the same direction are simply removed, a model we refer to as the *erased* DCM, the degree distributions will remain asymptotically unchanged.

For the purposes of this paper, self-loops and multiple edges in the same direction do not affect the main convergence result for the ranking scores, and therefore we do not require the DCM to result in a simple graph. A similar observation was made in the paper by van der Hofstad et al. [36] when analyzing distances in the undirected CM.

Throughout the paper, we will use $\mathscr{F}_n = \sigma((\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n))$ to denote the sigma algebra generated by the extended bi-degree sequence, which does not include information about the random pairing. To simplify the notation, we will use $\mathbb{P}_n(\cdot) = P(\cdot|\mathscr{F}_n)$ and $\mathbb{E}_n[\cdot] = E[\cdot|\mathscr{F}_n]$ to denote the conditional probability and conditional expectation, respectively, given $\mathscr{F}_n$.

4 Spectral ranking algorithms

In this section we introduce the class of ranking algorithms that we analyze in this paper. Following the terminology from [7], these algorithms belong to the class of *spectral centrality measures*, which 'compute the left dominant eigenvector of some matrix derived from the graph'. We point out that the construction of the matrix of weights and the definition of the rank vector that we give in Section 4.1 is not particular to the DCM.

4.1 Definition of the rank vector

The general class of spectral ranking algorithms we consider are determined by a matrix of weights $M = M(n) \in \mathbb{R}^{n\times n}$ and a personalization vector $\mathbf{Q} \in \mathbb{R}^n$. More precisely, given a directed graph with $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$ as its extended bi-degree sequence, we define the (i,j)th component of matrix M as follows:

$$M_{i,j} = \begin{cases} s_{ij}C_i, & \text{if there are } s_{ij} \text{ edges from } i \text{ to } j, \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

The rank vector $\mathbf{R} = (R_1, \ldots, R_n)$ is then defined to be the solution to the system of equations

$$\mathbf{R} = \mathbf{R}M + \mathbf{Q}. \tag{4.2}$$

Remark 4.1 *In the case of the PageRank algorithm, $C_i = c/D_i$, $Q_i = 1 - c$ for all i, and the constant $0 < c < 1$ is the so-called damping factor.*

4.2 Finitely many iterations

To solve the system of equations given in (4.2) we proceed via matrix iterations [27]. To initialize the process let $\mathbf{1}$ be the (row) vector of ones in $\mathbb{R}^n$ and let $\mathbf{r}_0 = r_0\mathbf{1}$, with $r_0 \in \mathbb{R}$. Define

$$\mathbf{R}^{(n,0)} = \mathbf{r}_0,$$

and for $k \geq 1$,

$$\mathbf{R}^{(n,k)} = \mathbf{r}_0 M^k + \sum_{i=0}^{k-1} \mathbf{Q}M^i.$$

With this notation, we have that the solution $\mathbf{R}$ to (4.2), provided it exists, can be written as

$$\mathbf{R} = \mathbf{R}^{(n,\infty)} = \sum_{i=0}^{\infty} \mathbf{Q}M^i.$$

We are interested in analyzing a randomly chosen coordinate of the vector $\mathbf{R}^{(n,\infty)}$. The first step, as described in Section 2.2, is to show that we can do so by using only finitely many matrix iterations. To this end note that

$$\mathbf{R}^{(n,k)} - \mathbf{R}^{(n,\infty)} = \mathbf{r}_0 M^k - \sum_{i=k}^{\infty} \mathbf{Q}M^i = \left(\mathbf{r}_0 - \sum_{i=0}^{\infty} \mathbf{Q}M^i\right) M^k.$$

Moreover,

$$\left|\left|\mathbf{R}^{(n,k)} - \mathbf{R}^{(n,\infty)}\right|\right|_1 \leq \left|\left|\mathbf{r}_0 M^k\right|\right|_1 + \sum_{i=0}^{\infty} \left|\left|\mathbf{Q}M^{k+i}\right|\right|_1.$$

Next, note that for any row vector $\mathbf{y} = (y_1, y_2, \ldots, y_n)$,

$$\begin{aligned} ||\mathbf{y}M^r||_1 &\leq \sum_{j=1}^{n} |\mathbf{y}(M^r)_{\bullet j}| \leq \sum_{j=1}^{n}\sum_{i=1}^{n} |y_i (M^r)_{ij}| \\ &= \sum_{i=1}^{n} |y_i| \sum_{j=1}^{n} |(M^r)_{ij}| = \sum_{i=1}^{n} |y_i| \cdot ||M^r_{i\bullet}||_1 \\ &\leq ||\mathbf{y}||_1 \, ||M^r||_\infty \,, \end{aligned}$$

where $A_{i\bullet}$ and $A_{\bullet j}$ are the ith row and jth column, respectively, of matrix A, and $||A||_\infty = \max_{1\leq i\leq n} ||A_{i\bullet}||_1$ is the operator infinity norm. It follows that if we assume that $\max_{1\leq i\leq n} |C_i|D_i \leq c$ for some $c \in (0,1)$, then we have

$$||M^r||_\infty \leq ||M||_\infty^r = \left(\max_{1\leq i\leq n} |C_i|D_i \right)^r \leq c^r.$$

In this case we conclude that

$$\begin{aligned} \left|\left| \mathbf{R}^{(n,k)} - \mathbf{R}^{(n,\infty)} \right|\right|_1 &\leq ||\mathbf{r}_0||_1 c^k + \sum_{i=0}^{\infty} ||\mathbf{Q}||_1 c^{k+i} \\ &= |r_0| n c^k + ||\mathbf{Q}||_1 \frac{c^k}{1-c}. \end{aligned}$$

Now note that all the coordinates of the vector $\mathbf{R}^{(n,k)} - \mathbf{R}^{(n,\infty)}$ have the same distribution, since by construction, the configuration model makes all permutations of the nodes' labels equally likely. Hence, the randomly chosen node may as well be the first node, and the error that we make by considering only finitely many iterations in its approximation is bounded in expectation by

$$\begin{aligned} \mathbb{E}_n \left[\left| R_1^{(n,k)} - R_1^{(n,\infty)} \right| \right] &= \frac{1}{n} \mathbb{E}_n \left[\left|\left| \mathbf{R}^{(n,k)} - \mathbf{R}^{(n,\infty)} \right|\right|_1 \right] \\ &\leq |r_0| c^k + \mathbb{E}_n \left[||\mathbf{Q}||_1 \right] \frac{c^k}{n(1-c)} \\ &= \left(|r_0| + \frac{1}{n(1-c)} \sum_{i=1}^{n} |Q_i| \right) c^k. \end{aligned}$$

It follows that if we let

$$B_n = \left\{ \max_{1\leq i\leq n} |C_i|D_i \leq c, \, \frac{1}{n}\sum_{i=1}^{n} |Q_i| \leq H \right\} \tag{4.3}$$

for some constants $c \in (0,1)$ and $H < \infty$, then Markov's inequality yields

$$\begin{aligned}
&P\left(\left|R_1^{(n,k)} - R_1^{(n,\infty)}\right| > n^{-\epsilon} \middle| B_n\right) \\
&= \frac{1}{P(B_n)} E\left[1(B_n)\mathbb{E}_n\left[1\left(\left|R_1^{(n,k)} - R_1^{(n,\infty)}\right| > n^{-\epsilon}\right)\right]\right] \\
&\leq \frac{1}{P(B_n)} E\left[1(B_n) n^{\epsilon}\mathbb{E}_n\left[\left|R_1^{(n,k)} - R_1^{(n,\infty)}\right|\right]\right] \\
&\leq \left(|r_0| + \frac{1}{1-c} E\left[\frac{1}{n}\sum_{i=1}^{n}|Q_i| \middle| B_n\right]\right) n^{\epsilon} c^k \\
&\leq \left(|r_0| + \frac{H}{1-c}\right) n^{\epsilon} c^k. \qquad (4.4)
\end{aligned}$$

We have thus derived the following result.

Proposition 4.2 *Consider the directed configuration graph generated by the extended bi-degree sequence* $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$ *and let* B_n *be defined according to* (4.3)*. Then, for any* $x_n \to \infty$ *and any* $k \geq 1$*, we have*

$$P\left(\left|R_1^{(n,\infty)} - R_1^{(n,k)}\right| > x_n^{-1} \middle| B_n\right) = O\left(x_n c^k\right)$$

as $n \to \infty$*.*

This completes the first step of our approach. In the next section we will explain how to couple the graph, as seen from a randomly chosen node, with an appropriate branching tree.

5 Construction of the graph and coupling with a branching tree

The next step in our approach is to approximate the distribution of $R_1^{(n,k)}$ with the rank of the root node of a suitably constructed branching tree. To ensure that we can construct such a tree we require the extended bi-degree sequence to satisfy some further properties with high probability. These properties are summarized in the following assumption.

Assumption 5.1 *Let* $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$ *be an extended bi-degree sequence for which there exists constants* $H, \nu_i > 0$*,* $i = 1, \ldots, 5$*, with*

$$\mu := \nu_2/\nu_1, \quad \lambda := \nu_3/\nu_1 \qquad \text{and} \qquad \rho := \nu_5\mu/\nu_1 < 1,$$

$0 < \kappa \leq 1$, *and* $0 < c, \gamma, \epsilon < 1$ *such that the events*

$$\begin{aligned}
\Omega_{n,1} &= \left\{ \left| \sum_{r=1}^{n} D_r - n\nu_1 \right| \leq n^{1-\gamma} \right\}, \\
\Omega_{n,2} &= \left\{ \left| \sum_{r=1}^{n} D_r N_r - n\nu_2 \right| \leq n^{1-\gamma} \right\}, \\
\Omega_{n,3} &= \left\{ \left| \sum_{r=1}^{n} D_r^2 - n\nu_3 \right| \leq n^{1-\gamma} \right\}, \\
\Omega_{n,4} &= \left\{ \left| \sum_{r=1}^{n} D_r^{2+\kappa} - n\nu_4 \right| \leq n^{1-\gamma} \right\}, \\
\Omega_{n,5} &= \left\{ \left| \sum_{r=1}^{n} |C_r| D_r - n\nu_5 \right| \leq n^{1-\gamma}, \max_{1 \leq r \leq n} |C_r| D_r \leq c \right\}, \\
\Omega_{n,6} &= \left\{ \sum_{r=1}^{n} |Q_r| \leq Hn \right\},
\end{aligned}$$

satisfy as $n \to \infty$,

$$P\left(\Omega_n^c\right) = P\left(\left(\bigcap_{i=1}^{6} \Omega_{n,i}\right)^c\right) = O\left(n^{-\epsilon}\right).$$

It is clear from (4.3) that $\Omega_n \subseteq B_n$, hence Proposition 4.2 holds under Assumption 5.1. We also point out that all six conditions in the assumption are in the spirit of the Weak Law of Large Numbers, and are therefore general enough to be satisfied by many different constructions of the extended bi-degree sequence. As an example, we give in Section 7 an algorithm based on sequences of i.i.d. random variables that satisfies Assumption 5.1.

In Sections 5.1–5.4 we describe in detail how to construct a coupling of the directed graph $\mathcal{G}_n$ and its approximating branching tree. We start by explaining the terminology and notation in Section 5.1, followed by the construction itself in Section 5.2. Then, in Section 5.3 we present the Coupling Lemma 5.4, which is the main result of Section 5. Finally, Section 5.4 explains how to compute the rank of the root node in the coupled tree.

5.1 Terminology and notation

Throughout the remainder of the paper we will interchangeably refer to the $\{N_i\}$ as the in-degrees/number of offspring/number of inbound stubs, to the $\{D_i\}$ as the out-degrees/number of outbound links/number of outbound stubs, to the $\{C_i\}$ as the weights, and to the $\{Q_i\}$ as the personalization values. We will refer to these four characteristics of a node as the *node attributes*.

The fact that we are working with a directed graph combined with the presence of weights, means that we need to use a more general kind of tree in our coupling than the standard branching process typically used in the random graph literature. To this end, we will define a process we call a Thorny Branching Tree (TBT), where each individual (node) in the tree has a directed edge

pointing towards its parent, and also a certain number of unpaired outbound links (pointing, say, to an artificial node outside of the tree). The name 'thorny' is due to these unpaired outbound links, see Figure 1. We point out that the structure of the tree (i.e., parent-offspring relations) is solely determined by the number of offspring.

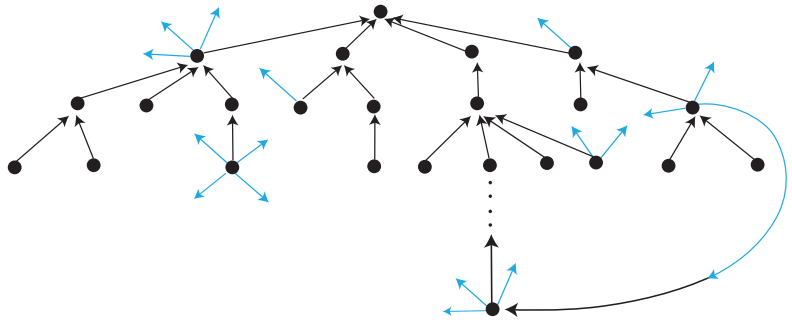


Figure 1: Graph construction process. Unpaired outbound links are in blue.

The simpler structure of a tree compared to a general graph allows for a more natural enumeration of its nodes. As usually in the context of branching processes, we let each node in the TBT have a label of the form $\mathbf{i} = (i_1, i_2, \ldots, i_k) \in \mathcal{U}$, where $\mathcal{U} = \bigcup_{k=0}^{\infty} (\mathbb{N}_+)^k$ is the set of all finite sequences of positive integers. Here, the convention is that $\mathbb{N}_+^0 = \{\emptyset\}$ contains the null sequence $\emptyset$. Also, for $\mathbf{i} = (i_1)$ we simply write $\mathbf{i} = i_1$, that is, without the parenthesis. Note that this form of enumeration gives the complete lineage of each individual in the tree.

We will use the following terminology and notation throughout the paper.

Definition 5.2 *We say that a node i in the graph (resp. TBT) is at distance k of the first (resp. root) node if it can reach the first (resp. root) node in k steps, but not in any less than k steps.*

In addition, for $r \geq 0$, we define on the graph/tree the following processes:

- A_r: set of nodes in the graph at distance r of the first node.
- $\hat{A}_r$: set of nodes in the tree at distance r of the root node ($\hat{A}_r$ is also the set of nodes in the rth generation of TBT, with the root node being generation zero).
- Z_r: number of inbound stubs of all the nodes in the graph at distance r of the first node ($Z_r \geq |A_{r+1}|$).
- $\hat{Z}_r$: number of inbound stubs of all the nodes in generation r of the TBT ($\hat{Z}_r = |\hat{A}_{r+1}|$).
- V_r: number of outbound stubs of all the nodes in the graph at distance r of the first node.
- $\hat{V}_r$: number of outbound stubs of all the nodes in generation r of the TBT.

Finally, given the extended bi-degree sequence $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$, we introduce two empirical distributions that will be used in the construction of the coupling. The first one describes the attributes

of a randomly chosen node:

$$f_n^*(i,j,s,t) = \sum_{k=1}^{n} 1(N_k = i, D_k = j, C_k = s, Q_k = t)\mathbb{P}_n(\text{node } k \text{ is sampled})$$
$$= \frac{1}{n}\sum_{k=1}^{n} 1(N_k = i, D_k = j, C_k = s, Q_k = t). \tag{5.1}$$

The second one, corresponds to the attributes of a node that is chosen by sampling uniformly at random from all the L_n outbound stubs:

$$f_n(i,j,s,t) = \sum_{k=1}^{n} 1(N_k = i, D_k = j, C_k = s, Q_k = t)\mathbb{P}_n(\text{an outbound stub from node } k \text{ is sampled})$$
$$= \sum_{k=1}^{n} 1(N_k = i, D_k = j, C_k = s, Q_k = t)\frac{D_k}{L_n}. \tag{5.2}$$

Note that this is a size-biased distribution, since nodes with more outbound stubs are more likely to be chosen, whereas nodes with no outbound stubs (dangling nodes) cannot be chosen.

5.2 Construction of the coupling

Given an extended bi-degree sequence $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$ we now explain how to construct the graph $\mathcal{G}_n$ and its coupled TBT through a breadth-first exploration process. From this point onwards we will ignore the implicit numbering of the nodes in the definition of the extended bi-degree sequence and rename them according to the order in which they appear in the graph exploration process.

To keep track of which outbound stubs have already been matched we borrow the approach used in [36] and label them 1, 2, or 3 according to the following rules:

1. Outbound stubs with label 1 are stubs belonging to a node that is not yet attached to the graph.
2. Outbound stubs with label 2 belong to nodes that are already part of the graph but that have not yet been paired with an inbound stub.
3. Outbound stubs with label 3 are those which have already been paired with an inbound stub and now form an edge in the graph.

The graph $\mathcal{G}_n$ is constructed as follows. Right before the first node is sampled, all outbound stubs are labeled 1. To start the construction of the graph, we choose randomly a node (all nodes with the same probability) and call it node 1. The attributes of this first node, denoted by (N_1, D_1, C_1, Q_1), are sampled from distribution (5.1).

After the first node is chosen, its D_1 outbound stubs are labeled 2. We then proceed to pair the first of the $Z_0 = N_1$ inbound stubs of the first node with a randomly chosen outbound stub. The corresponding node is attached to the graph by forming an edge pointing to node 1 using the chosen outbound stub, which receives a label 3, and all the remaining outbound stubs from the new node are labeled 2. Note that it is possible that the chosen node is node 1 itself, in which case the pairing forms a self-loop and no new nodes are added to the graph. We continue in this way until

all Z_0 inbound stubs of node 1 have been paired with randomly chosen outbound stubs. Since these outbound stubs are sampled independently and with replacement from all the possible L_n outbound stubs, this corresponds to drawing the node attributes independently from the random distribution (5.2). Note that in the construction of the graph any unfeasible matches will be discarded, and therefore the attributes of nodes in $\mathcal{G}_n$ do not necessarily have distribution (5.2), but rather have the conditional distribution given the pairing was feasible. We will use the vector (N_i, D_i, C_i, Q_i) to denote the attributes of the ith node to be added to the graph.

In general, the kth iteration of this process is completed when all Z_{k-1} inbound stubs have been matched with an outbound stub, and the corresponding node attributes have been assigned. The process ends when all L_n inbound stubs have been paired. Note that whenever an outbound stub with label 2 is chosen a cycle or a double edge is formed in the graph.

Next, we explain how the TBT is constructed. To distinguish the attribute vectors of nodes in the TBT from those of nodes in the graph, we denote them by $(\hat{N}_{\mathbf{i}}, \hat{D}_{\mathbf{i}}, \hat{C}_{\mathbf{i}}, \hat{Q}_{\mathbf{i}})$, $\mathbf{i} \in \mathcal{U}$. We start with the root node (node $\emptyset$) that has the same attributes as node 1 in the graph: $(\hat{N}_\emptyset, \hat{D}_\emptyset, \hat{C}_\emptyset, \hat{Q}_\emptyset) \equiv (N_1, D_1, C_1, Q_1)$, sampled from distribution (5.1). Next, for $k \geq 1$, each of the $\hat{Z}_{k-1}$ individuals in the kth generation will independently have offspring, outbound stubs, weight and personalization value according to the joint distribution $f_n(i, j, s, t)$ given by (5.2).

Now, we explain how the coupling with the graph, i.e., the simultaneous construction of the graph and the TBT, is done.

1) Whenever an outbound stub is sampled randomly in an attempt to add an edge to $\mathcal{G}_n$, then, independently of the stub's label, a new offspring is added to the TBT. This is done to maintain the branching property (i.i.d. node attributes). In particular, if the chosen outbound stub belongs to node j, then the new offspring in the TBT will have $D_j - 1$ outbound stubs (which will remain unpaired), N_j inbound stubs (number of offspring), weight C_j, and personalization value Q_j.

2) If an outbound stub with label 1 is chosen, then both the graph and the TBT will connect the chosen outbound stub to the inbound stub being matched, resulting in a node being added to the graph and an offspring being born to its parent. We then update the labels by giving a 2 label to all the 'sibling' outbound stubs of the chosen outbound stub, and a 3 label to the chosen outbound stub itself.

3) If an outbound stub with label 2 is chosen it means that its corresponding node already belongs to the graph, and a cycle, self-loop, or multiple edge is created. We then relabel the chosen outbound stub with a 3. An offspring is born in the TBT according to 1).

4) If an outbound stub with label 3 is chosen it means that the chosen outbound stub has already been matched. In terms of the construction of the graph, this case represents a failed attempt to match the current inbound stub, and we have to keep sampling until we draw an outbound stub with label 1 or 2. Once we do so, we update the labels according to the rules given above. An offspring is born in the TBT according to 1).

Note that as long as we do not sample any outbound stub with label 2 or 3, the graph $\mathcal{G}_n$ and the TBT are identical. Once we draw the first outbound stub with label 2 or 3 the processes Z_k

and $\hat{Z}_k$ may start to disagree. The moment this occurs we say that the coupling has been broken. Nonetheless, we will continue with the pairing process following the rules given above until all L_n inbound stubs have been paired. The construction of the TBT also continues in parallel by keeping the synchronization of the pairing whenever the inbound stub being matched belongs to a node that is both in the graph and the tree. If the pairing of all L_n inbound stubs is completed after k iterations of the process, then we will have completed k generations in the TBT. Moreover, up to the time the coupling breaks, a node $\mathbf{i} \in \hat{A}_k$ is also the jth node to be added to the graph, where:

$$j = 1 + \sum_{r=0}^{k-2} \hat{Z}_r + \sum_{s=1}^{i_{k-1}-1} \hat{N}_{(i_1,\ldots,i_{k-2},s)} + i_k,$$

with the convention that $\sum_{r=a}^{b} x_r = 0$ if $b < a$.

Definition 5.3 *Let τ be the number of generations in the TBT that can be completed before the first outbound stub with label 2 or 3 is drawn, i.e., $\tau = k$ if and only if the first inbound stub to draw an outbound stub with label 2 or 3 belonged to a node $\mathbf{i} \in \hat{A}_k$.*

The main result in this section consists in showing that provided the extended bi-degree sequence $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$ satisfies Assumption 5.1, the coupling breaks only after a number of generations that is of order $\log n$, which combined with Proposition 4.2 will allow us to approximate the rank of a randomly chosen node in the graph with the rank of the root node of the coupled TBT.

5.3 The coupling lemma

It follows from the construction in Section 5.2 that, before the coupling breaks, the neighborhood of node 1 in $\mathcal{G}_n$ and of the root node in the TBT are identical. Recall also from Proposition 4.2 that we only need a finite number k of matrix iterations to approximate the elements of the rank vector to any desired precision. Furthermore, the weight matrix M is such that the elements $(M^r)_{i,1}$, $1 \le i \le n$, $1 \le r \le k$, depend only on the k-neighborhood of node 1. Hence, if the coupling holds for $\tau > k$ generations, then the rank score of node 1 in $\mathcal{G}_n$ is exactly the same as that of the root node of the TBT restricted to those same k generations. The following coupling lemma will allow us to complete the appropriate number of generations in the tree to obtain the desired level of precision in Proposition 4.2. Its proof is rather technical and is therefore postponed to Section 8.1.

Lemma 5.4 *Suppose $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$ satisfies Assumption 5.1. Then,*

- *for any $1 \le k \le h \log n$ with $0 < h < 1/(2 \log \mu)$, if $\mu > 1$,*
- *for any $1 \le k \le n^b$ with $0 < b < \min\{1/2, \gamma\}$, if $\mu \le 1$,*

we have

$$P(\tau \le k | \Omega_n) = \begin{cases} O\left((n/\mu^{2k})^{-1/2}\right), & \mu > 1, \\ O\left((n/k^2)^{-1/2}\right), & \mu = 1, \\ O\left(n^{-1/2}\right), & \mu < 1, \end{cases}$$

as $n \to \infty$.

Remark 5.5 *The constant μ was defined in Assumption 5.1, and it corresponds to the limiting expected number of offspring that each node in the TBT (with the exception of the root node) will have. The coupling between the graph and the TBT will hold for any $\mu > 0$.*

We conclude from Lemma 5.4 that if $\hat{R}^{(n,k)} := \hat{R}_{\emptyset}^{(n,k)}$ denotes the rank of the root node of the TBT restricted to the first k generations, then, for any $\delta > 0$,

$$P\left(\left|R_1^{(n,k)} - \hat{R}^{(n,k)}\right| > n^{-\delta}\Big|\,\Omega_n\right) \leq P(\tau < k|\Omega_n) := \varphi(k,n).$$

Note that the super index n does not refer to the number of nodes in the tree, and is being used only in the definition of the distributions f_n^* and f_n (given in (5.1) and (5.2), respectively).

This observation, combined with Proposition 4.2, implies that if we let $k_n = \lceil h \log n \rceil$, when $\mu > 1$, and $k_n = n^{\varepsilon}$, when $\mu \leq 1$, where $h = (1-\varepsilon)/(2\log\mu)$ and $0 < \varepsilon < \min\{1/3, \gamma\}$, then

$$\begin{aligned} P\left(\left|R_1^{(n,\infty)} - \hat{R}^{(n,k_n)}\right| > n^{-\delta}\Big|\,\Omega_n\right) &\leq P\left(\left|R_1^{(n,\infty)} - R_1^{(n,k_n)}\right| > n^{-\delta}/2\Big|\,\Omega_n\right) \\ &\quad + P\left(\left|R_1^{(n,k_n)} - \hat{R}^{(n,k_n)}\right| > n^{-\delta}/2\Big|\,\Omega_n\right) \\ &= O\left(n^{\delta} c^{k_n} + \varphi(k_n,n)\right) \\ &= O\left(n^{\delta - h|\log c|} + n^{-\varepsilon/2}\right). \end{aligned} \tag{5.3}$$

In view of (5.3), analyzing the distribution of $R_1^{(n,k)}$ in the graph reduces to analyzing the rank of the root node of the coupled TBT, $\hat{R}^{(n,k)}$. In the next section, we compute $\hat{R}^{(n,k)}$ by relating it to a linear process constructed on the TBT.

5.4 Computing the rank of nodes in the TBT

In order to compute $\hat{R}^{(n,k)}$ we need to introduce a new type of weights. To simplify the notation, for $\mathbf{i} = (i_1, \ldots, i_k)$ we will use $(\mathbf{i}, j) = (i_1, \ldots, i_k, j)$ to denote the index concatenation operation; if $\mathbf{i} = \emptyset$, then $(\mathbf{i}, j) = j$. Each node $\mathbf{i}$ is then assigned a weight $\hat{\Pi}_{\mathbf{i}}$ according to the recursion

$$\hat{\Pi}_{\emptyset} \equiv 1 \qquad \text{and} \qquad \hat{\Pi}_{(\mathbf{i},j)} = \hat{\Pi}_{\mathbf{i}} \hat{C}_{(\mathbf{i},j)}, \quad \mathbf{i} \in \mathcal{U}.$$

Note that the $\hat{\Pi}_{\mathbf{i}}$'s are the products of all the weights $\hat{C}_{\mathbf{j}}$ along the path leading to node $\mathbf{i}$, as depicted in Figure 2.

Next, for each fixed $k \in \mathbb{N}$ and each node $\mathbf{i}$ in the TBT define $\hat{R}_{\mathbf{i}}^{(n,k)}$ to be the rank of node $\mathbf{i}$ computed on the subtree that has $\mathbf{i}$ as its root and that is restricted to having only k generations, with each of the $|\hat{A}_k|$ nodes having rank r_0. In mathematical notation,

$$\hat{R}_{\mathbf{i}}^{(n,k)} = \sum_{j=1}^{\hat{N}_{\mathbf{i}}} \hat{C}_{(\mathbf{i},j)} \hat{R}_{(\mathbf{i},j)}^{(n,k-1)} + \hat{Q}_{\mathbf{i}}, \qquad k \geq 1, \qquad \hat{R}_{\mathbf{j}}^{(n,0)} = r_0. \tag{5.4}$$

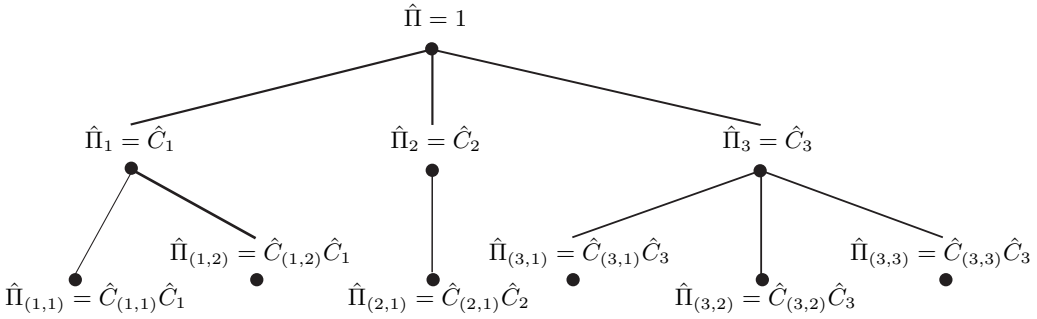


Figure 2: Weighted tree.

Iterating (5.4) gives

$$\begin{aligned}\hat{R}^{(n,k)} &= \sum_{\mathbf{i}\in\hat{A}_1} \hat{\Pi}_{\mathbf{i}}\hat{R}_{\mathbf{i}}^{(n,k-1)} + \hat{Q}_{\emptyset} = \sum_{\mathbf{i}\in\hat{A}_1} \hat{\Pi}_{\mathbf{i}} \left(\sum_{j=1}^{\hat{N}_{\mathbf{i}}} \hat{C}_{(\mathbf{i},j)}\hat{R}_{(\mathbf{i},j)}^{(n,k-2)} + \hat{Q}_{\mathbf{i}} \right) + \hat{Q}_{\emptyset} \\ &= \sum_{\mathbf{i}\in\hat{A}_2} \hat{\Pi}_{\mathbf{i}}\hat{R}_{\mathbf{i}}^{(n,k-2)} + \sum_{\mathbf{i}\in\hat{A}_1} \hat{\Pi}_{\mathbf{i}}\hat{Q}_{\mathbf{i}} + \hat{Q}_{\emptyset} = \cdots = \sum_{\mathbf{i}\in\hat{A}_k} \hat{\Pi}_{\mathbf{i}} r_0 + \sum_{s=0}^{k-1} \sum_{\mathbf{i}\in\hat{A}_s} \hat{\Pi}_{\mathbf{i}}\hat{Q}_{\mathbf{i}}. \end{aligned} \tag{5.5}$$

The last step in our proof of the main result is to identify the limit of $\hat{R}^{(n,k_n)}$ as $n \to \infty$, for a suitable chosen $k_n \to \infty$. This is done in the next section.

6 Coupling with a weighted branching process

The last step in the derivation of our approximation for the rank of a randomly chosen node in the graph $\mathcal{G}_n$ is to substitute the rank of the root node in the TBT, which is defined with respect to empirical distributions based on the extended bi-degree sequence $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$, with a limiting random variable independent of the size of the graph, n.

The appropriate limit will be given in terms of a solution to a certain stochastic fixed-point equation (SFPE). The appeal of having such a representation is that these solutions have been thoroughly studied in the WBPs literature, and in many cases exact asymptotics describing their tail behavior are available [22, 23, 30]. We will elaborate more on this point after we state our main result.

As already mentioned in Section 2, our main result shows that

$$R_1^{(n,\infty)} \Rightarrow \mathcal{R}^*$$

as $n \to \infty$, where $\mathcal{R}^*$ can be written in terms of the so-called endogenous solution to a linear SFPE. Before we write the expression for $\mathcal{R}^*$ we will need to introduce a few additional concepts.

6.1 The linear branching stochastic fixed-point equation

We define the linear branching SFPE according to:

$$\mathcal{R} \stackrel{\mathcal{D}}{=} \sum_{j=1}^{\mathcal{N}} \mathcal{C}_j \mathcal{R}_j + \mathcal{Q}, \tag{6.1}$$

where $(\mathcal{N}, \mathcal{Q}, \mathcal{C}_1, \mathcal{C}_2, \dots)$ is a real-valued random vector with $\mathcal{N} \in \mathbb{N} \cup \{\infty\}$, $P(|\mathcal{Q}| > 0) > 0$, and the $\{\mathcal{R}_i\}$ are i.i.d. copies of $\mathcal{R}$, independent of the vector $(\mathcal{N}, \mathcal{Q}, \mathcal{C}_1, \mathcal{C}_2, \dots)$. The vector $(\mathcal{N}, \mathcal{Q}, \mathcal{C}_1, \mathcal{C}_2, \dots)$ is often referred to as the generic branching vector, and in the general setting is allowed to be arbitrarily dependent with the weights $\{\mathcal{C}_i\}$ not necessarily identically distributed. This equation is also known as the "smoothing transform" [21, 16, 1, 3].

In the context of ranking algorithms, we can identify $\mathcal{N}$ with the in-degree of a node, $\mathcal{Q}$ with its personalization value, and the $\{\mathcal{C}_i\}$ with the weights of the neighboring nodes pointing to it. We now explain how to construct a solution to (6.1).

Similarly as what we did in Section 5.4 and using the same notation introduced there, we construct a weighted tree using a sequence $\{(\mathcal{N}_{\mathbf{i}}, \mathcal{Q}_{\mathbf{i}}, \mathcal{C}_{(\mathbf{i},1)}, \mathcal{C}_{(\mathbf{i},2)}, \dots)\}_{\mathbf{i} \in \mathcal{U}}$ of i.i.d. copies of the vector $(\mathcal{N}, \mathcal{Q}, \mathcal{C}_1, \mathcal{C}_2, \dots)$ to define its structure and its node attributes. This construction is known in the literature as a WBP [32]. Next, let $\mathcal{A}_k$ denote the number of individuals in the kth generation of the tree, and to each node $\mathbf{i}$ in the tree assign a weight $\Pi_{\mathbf{i}}$ according to the recursion

$$\Pi_{\emptyset} \equiv 1 \qquad \text{and} \qquad \Pi_{(\mathbf{i},j)} = \Pi_{\mathbf{i}}\, \mathcal{C}_{(\mathbf{i},j)}, \quad \mathbf{i} \in \mathcal{U}.$$

Then, the random variable formally defined as

$$\mathcal{R} := \sum_{k=0}^{\infty} \sum_{\mathbf{i} \in \mathcal{A}_k} \Pi_{\mathbf{i}} \mathcal{Q}_{\mathbf{i}} \tag{6.2}$$

is called the endogenous solution to (6.1), and provided $E\left[\sum_{i=1}^{\mathcal{N}} |\mathcal{C}_i|^{\beta}\right] < 1$ for some $0 < \beta \leq 1$, it is well defined (see [23], Lemma 4.1). The name "endogenous" comes from its explicit construction in terms of the weighted tree. We point out that equation (6.1) has in general multiple solutions [3, 4], so it is important to emphasize that the one considered here is the endogenous one.

Comparing (5.5) and (6.2) suggests that $\hat{R}^{(n,k_n)}$ should converge to $\mathcal{R}$ provided the distribution of the attribute vectors in the TBT converges to the distribution of the generic branching vector in the WBP, but in order to formalize this heuristic there are two difficulties that we need to overcome. The first one is that the TBT was defined using a sequence of (conditionally) independent vectors of the form $\{(\hat{N}_{\mathbf{i}}, \hat{Q}_{\mathbf{i}}, \hat{C}_{\mathbf{i}})\}_{\mathbf{i} \in \mathcal{U}}$, where by construction (see Assumption 5.1 and (5.2)) the generic attribute vector $(\hat{N}_1, \hat{Q}_1, \hat{C}_1)$ is dependent. Note that this implies that the vectors $(\hat{N}_{\mathbf{i}}, \hat{Q}_{\mathbf{i}}, \hat{C}_{(\mathbf{i},1)}, \hat{C}_{(\mathbf{i},2)}, \dots)$ and $\{(\hat{N}_{(\mathbf{i},j)}, \hat{Q}_{(\mathbf{i},j)}, \hat{C}_{(\mathbf{i},j,1)}, \hat{C}_{(\mathbf{i},j,2)}, \dots)\}_{j \geq 1}$ are dependent through the dependence between $\hat{N}_{(\mathbf{i},j)}$ and $\hat{C}_{(\mathbf{i},j)}$, which destroys the branching property of the WBP. The second problem is that the root node of the TBT has a different distribution from the rest of the nodes in the tree.

It is therefore to be expected that we will need something more than weak convergence of the node attributes to obtain the convergence of $\hat{R}^{(n,k_n)}$ we seek. To solve the first problem we will require that $(\hat{N}_1, \hat{Q}_1, \hat{C}_1)$ converges to $(\mathcal{N}, \mathcal{Q}, \mathcal{C})$ with $\mathcal{C}$ independent of $(\mathcal{N}, \mathcal{Q})$. Note that this will naturally

lead to the $\{\mathcal{C}_i\}$ being i.i.d. in (6.1). To solve the second problem we will allow the attributes of the root node in the TBT to converge to their own limit $(\mathcal{N}_0, \mathcal{Q}_0)$. In view of these observations we can now identify the limit of $\hat{R}^{(n,k_n)}$ to be:

$$\mathcal{R}^* := \sum_{i=1}^{\mathcal{N}_0} \mathcal{C}_i \mathcal{R}_i + \mathcal{Q}_0, \tag{6.3}$$

where the $\{\mathcal{R}_i\}$ are i.i.d. copies of $\mathcal{R}$, as given by (6.2), independent of the vector $(\mathcal{N}_0, \mathcal{Q}_0, \{\mathcal{C}_i\})$ with $\{\mathcal{C}_i\}$ i.i.d. and independent of $(\mathcal{N}_0, \mathcal{Q}_0)$. The appropriate condition ensuring that $\mathcal{R}^*$ is the correct limit is given in terms of the Kantorovich-Rubinstein distance (also known as the minimal l_1 distance or the Wasserstein distance).

Definition 6.1 *Consider the metric space $(\mathbb{R}^d, ||\cdot||_1)$, where $||\mathbf{x}||_1$ is the l_1 norm in $\mathbb{R}^d$. Let $M(\mu,\nu)$ denote the set of joint probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals μ and ν. Then, the Kantorovich-Rubinstein distance between μ and ν is given by*

$$d_1(\mu,\nu) = \inf_{\pi \in M(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} ||\mathbf{x}-\mathbf{y}||_1 \, d\pi(\mathbf{x},\mathbf{y}).$$

We point out that d_1 is only strictly speaking a distance when restricted to the subset of measures

$$\mathscr{P}_1(\mathbb{R}^d) := \left\{ \mu \in \mathscr{P}(\mathbb{R}^d) : \int_{\mathbb{R}^d} ||\mathbf{x}-\mathbf{x}_0||_1 \, d\mu(\mathbf{x}) < \infty \right\},$$

for some $\mathbf{x}_0 \in \mathbb{R}^d$, where $\mathscr{P}(\mathbb{R}^d)$ is the set of Borel probability measures on $\mathbb{R}^d$. We refer the interested reader to [39] for a thorough treatment of this distance, since Definition 6.1 gives only a special case.

An important property of the Kantorovich-Rubinstein distance is that if $\{\mu_k\}_{k\in\mathbb{N}}$ is a sequence of probability measures in $\mathscr{P}_1(\mathbb{R}^d)$, then convergence in d_1 to a limit $\mu \in \mathscr{P}_1(\mathbb{R}^d)$ is equivalent to weak convergence. Furthermore, d_1 satisfies the useful **duality formula**:

$$d_1(\mu,\nu) = \sup_{||\psi||_{\text{Lip}} \le 1} \left\{ \int_{\mathbb{R}^d} \psi(\mathbf{x}) d\mu(\mathbf{x}) - \int_{\mathbb{R}^d} \psi(\mathbf{x}) d\nu(\mathbf{x}) \right\}$$

for all $\mu, \nu \in \mathscr{P}_1(\mathbb{R}^d)$, where the supremum is taken over al Lipschitz continuous functions $\psi : \mathbb{R}^d \to \mathbb{R}$ with Lipschitz constant one (see Remark 6.5 in [39]).

We now give the required assumption. With some abuse of notation, for joint distribution functions $F_n, F \in \mathbb{R}^d$ we write $d_1(F_n, F)$ to denote the Kantorovich-Rubinstein distance between their probability measures μ_n and μ. The symbol $\xrightarrow{P}$ denotes convergence in probability.

Assumption 6.2 *Given the extended bi-degree sequence $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$ define*

$$F_n^*(m,q) := \frac{1}{n}\sum_{k=1}^{n} 1(N_k \le m, Q_k \le q) \quad \textit{and} \quad F_n(m,q,x) := \sum_{k=1}^{n} 1(N_k \le m, Q_k \le q, C_k \le x)\frac{D_k}{L_n}.$$

Suppose there exist random vectors $(\mathcal{N}_0, \mathcal{Q}_0)$ and $(\mathcal{N}, \mathcal{Q})$, and a random variable $\mathcal{C}$, such that

$$d_1(F_n^*, F^*) \xrightarrow{P} 0 \qquad \textit{and} \qquad d_1(F_n, F) \xrightarrow{P} 0,$$

as $n \to \infty$, where

$$F^*(m,q) := P(\mathcal{N}_0 \le m, \mathcal{Q}_0 \le q) \qquad \text{and} \qquad F(m,q,x) := P(\mathcal{N} \le m, \mathcal{Q} \le q)P(\mathcal{C} \le x).$$

Remark 6.3 *Note that Assumption 6.2 and the duality formula imply that*

$$\sup\left\{\mathbb{E}_n\left[\psi(\hat{N}_1, \hat{Q}_1, \hat{C}_1)\right] - E[\psi(\mathcal{N}, \mathcal{Q}, \mathcal{C})] : \psi \text{ is bounded and continuous}\right\}$$

converges to zero in probability, and therefore, by the bounded convergence theorem,

$$E\left[\psi(\hat{N}_1, \hat{Q}_1, \hat{C}_1)\right] \to E[\psi(\mathcal{N}, \mathcal{Q}, \mathcal{C})], \quad n \to \infty,$$

for any bounded and continuous function ψ, or equivalently, $(\hat{N}_1, \hat{Q}_1, \hat{C}_1) \Rightarrow (\mathcal{N}, \mathcal{Q}, \mathcal{C})$; similarly, $(\hat{N}_\emptyset, \hat{Q}_\emptyset) \Rightarrow (\mathcal{N}_0, \mathcal{Q}_0)$. The duality formula, combined with Assumption 5.1, also implies that $E[\mathcal{N}_0] = \nu_1$, $E[\mathcal{N}] = \mu$ and $E[\mathcal{C}] = \nu_5/\nu_1$.

6.2 Main Result

We are now ready to state the main result of this paper, which establishes the convergence of the rank of a randomly chosen node in the DCM to a non-degenerate random variable $\mathcal{R}^*$.

Theorem 6.4 *Suppose the extended bi-degree sequence $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$ satisfies Assumptions 5.1 and 6.2. Then,*

$$R_1^{(n,\infty)} \Rightarrow \mathcal{R}^*$$

as $n \to \infty$, where $\mathcal{R}^*$ is defined as in (6.3) *with the weights $\{\mathcal{C}_i\}$ i.i.d. and independent of $(\mathcal{N}_0, \mathcal{Q}_0)$, respectively of $(\mathcal{N}, \mathcal{Q})$ in* (2.1)*.*

Proof. Define Ω_n according to Assumption 5.1 and note that $P(\Omega_n^c) = O(n^{-\epsilon})$, so it suffices to show that $R_1^{(n,\infty)}$, conditional on Ω_n, converges weakly to $\mathcal{R}^*$. Note that by Assumption 5.1, $\rho = E[\mathcal{N}]E[|\mathcal{C}|] = \nu_5\mu/\nu_1 < 1$, which is a sufficient condition for $\mathcal{R}$ to be well defined (see Lemma 4.1 in [23]). First, when $\mu > 1$, fix $0 < \delta < |\log c|/(2\log\mu)$ and let $k_n = s\log n$, where $\delta/|\log c| < s < 1/(2\log\mu)$. Next, note that by the arguments leading to (5.3),

$$\begin{aligned} P\left(\left|R_1^{(n,\infty)} - \hat{R}^{(n,k_n)}\right| > n^{-\delta}\middle|\,\Omega_n\right) &= O\left(n^\delta c^{k_n} + (\mu^{2k_n}/n)^{1/2}\right) \\ &= O\left(n^{\delta - s|\log c|} + n^{(2s\log\mu - 1)/2}\right) = o(1) \end{aligned}$$

as $n \to \infty$. When $\mu \le 1$ we can take $k_n = n^\varepsilon$, with $\varepsilon < \min\{1/2, \gamma\}$, to obtain that the probability converges to zero. We then obtain that conditionally on Ω_n,

$$\left|R_1^{(n,\infty)} - \hat{R}^{(n,k_n)}\right| \Rightarrow 0.$$

That $\hat{R}^{(n,k_n)} \Rightarrow \mathcal{R}^*$ conditionally on Ω_n will follow from Theorem 4.8 in [12] and Assumption 6.2 once we verify that, as $n \to \infty$,

$$\mathbb{E}_n\left[\hat{N}_1|\hat{C}_1|\right] \xrightarrow{P} E[\mathcal{N}]E[|\mathcal{C}|] \qquad \text{and} \qquad \mathbb{E}_n\left[|\hat{Q}_1\hat{C}_1|\right] \xrightarrow{P} E[|\mathcal{Q}|]E[|\mathcal{C}|]. \tag{6.4}$$

To show that (6.4) holds define $\phi_K(q,x) = (|q| \wedge K)(|x| \wedge 1)$ for $K > 0$, and note that since ϕ_K is bounded and continuous, Assumption 6.2 and Remark 6.3 imply that

$$\mathbb{E}_n\left[\phi_K(\hat{Q}_1, \hat{C}_1)\right] \xrightarrow{P} E[\phi_K(\mathcal{Q}, \mathcal{C})] = E[|\mathcal{Q}| \wedge K]E[|\mathcal{C}|], \qquad n \to \infty.$$

Next, fix $\epsilon > 0$ and choose K such that $E[|\mathcal{Q}|1(|\mathcal{Q}| > K)] < \epsilon/4$. Then,

$$\begin{aligned}\left|\mathbb{E}_n\left[|\hat{Q}_1\hat{C}_1|\right] - E[|\mathcal{Q}\mathcal{C}|]\right| &\leq \left|\mathbb{E}_n\left[\phi_K(\hat{Q}_1, \hat{C}_1)\right] - E[\phi_K(\mathcal{Q}, \mathcal{C})]\right| \\ &\quad + \mathbb{E}_n\left[(|\hat{Q}_1| - K)^+|\hat{C}_1|\right] + E[(|\mathcal{Q}| - K)^+|\mathcal{C}|] \\ &\leq \left|\mathbb{E}_n\left[\phi_K(\hat{Q}_1, \hat{C}_1)\right] - E[\phi_K(\mathcal{Q}, \mathcal{C})]\right| + c\mathbb{E}_n\left[(|\hat{Q}_1| - K)^+\right] + c\epsilon/4,\end{aligned}$$

where we used that both $|\hat{C}_1|$ and $|\mathcal{C}|$ are bounded by $c < 1$. It follows that

$$\lim_{n\to\infty} P\left(\left|\mathbb{E}_n\left[|\hat{Q}_1\hat{C}_1|\right] - E[|\mathcal{Q}\mathcal{C}|]\right| > \epsilon\right) \leq \lim_{n\to\infty} P\left(\mathbb{E}_n\left[(|\hat{Q}_1| - K)^+\right] > \epsilon/2\right).$$

To show that this last limit is zero note that $(|x| - K)^+$ is Lipschitz continuous with Lipschitz constant one, so by the duality formula we obtain

$$\mathbb{E}_n\left[(|\hat{Q}_1| - K)^+\right] \xrightarrow{P} E[(|\mathcal{Q}| - K)^+] < \epsilon/4$$

as $n \to \infty$, which gives the desired limit.

The proof for $\mathbb{E}_n\left[|\hat{N}_1\hat{C}_1|\right]$ follows the same steps and is therefore omitted. ∎

6.3 Asymptotic behavior of the limit

We end this section by giving a limit theorem describing the tail asymptotics of $\mathcal{R}^*$; its proof is given in Section 8.2. This result covers the case where the weights $\{\mathcal{C}_i\}$ are nonnegative and either the limiting in-degree $\mathcal{N}$ or the limiting personalization value $\mathcal{Q}$ have a regularly varying distribution, which in turn implies the regular variation of $\mathcal{R}$. Then, we deduce the asymptotics of $\mathcal{R}^*$ using some results for weighted random sums with heavy-tailed summands. The corresponding theorems can be found in [30, 40].

Definition 6.5 *We say that a function f is regularly varying at infinity with index $-\alpha$, denoted $f \in \mathscr{R}_{-\alpha}$, if $f(x) = x^{-\alpha}L(x)$ for some slowly varying function L; and $L : [0,\infty) \to (0,\infty)$ is slowly varying if $\lim_{x\to\infty} L(\lambda x)/L(x) = 1$ for any $\lambda > 0$.*

We use the notation $f(x) \sim g(x)$ as $x \to \infty$ for $\lim_{x\to\infty} f(x)/g(x) = 1$.

Theorem 6.6 *Suppose the generic branching vector $(\mathcal{N}, \mathcal{Q}, \mathcal{C}_1, \mathcal{C}_2, \dots)$ is such that the weights $\{\mathcal{C}_i\}$ are nonnegative, bounded i.i.d. copies of $\mathcal{C}$, independent of $(\mathcal{N}, \mathcal{Q})$, $\mathcal{N} \in \mathbb{N}$ and $\mathcal{Q} \in \mathbb{R}$. Define $\rho = E[\mathcal{N}]E[\mathcal{C}]$ and $\rho_\alpha = E[\mathcal{N}]E[\mathcal{C}^\alpha]$ and let $\mathcal{R}$ be defined as in (6.2).*

- *If* $P(\mathcal{N} > x) \in \mathscr{R}_{-\alpha}$, $\alpha > 1$, $\rho \vee \rho_\alpha < 1$, $P(\mathcal{N}_0 > x) \sim \kappa P(\mathcal{N} > x)$ *as* $x \to \infty$ *for some* $\kappa > 0$, $E[\mathcal{Q}], E[\mathcal{Q}_0] > 0$, *and* $E\left[|\mathcal{Q}|^{\alpha+\epsilon} + |\mathcal{Q}_0|^{\alpha+\epsilon}\right] < \infty$ *for some* $\epsilon > 0$, *then*

$$P(\mathcal{R}^* > x) \sim \left(E[\mathcal{N}_0]E[\mathcal{C}^\alpha] + \kappa(1-\rho_\alpha)\right) \frac{(E[\mathcal{Q}]E[\mathcal{C}])^\alpha}{(1-\rho)^\alpha(1-\rho_\alpha)} P(\mathcal{N} > x), \qquad x \to \infty.$$

- *If* $P(\mathcal{Q} > x) \in \mathscr{R}_{-\alpha}$, $\alpha > 1$, $\rho \vee \rho_\alpha < 1$, $P(\mathcal{Q}_0 > x) \sim \kappa P(\mathcal{Q} > x)$ *as* $x \to \infty$ *for some* $\kappa > 0$, $E[|\mathcal{Q}|^\beta + |\mathcal{Q}_0|^\beta] < \infty$ *for all* $0 < \beta < \alpha$, *and* $E\left[|\mathcal{N}|^{\alpha+\epsilon} + |\mathcal{N}_0|^{\alpha+\epsilon}\right] < \infty$ *for some* $\epsilon > 0$, *then*

$$P(\mathcal{R}^* > x) \sim \left(E[\mathcal{N}_0]E[\mathcal{C}^\alpha] + \kappa(1-\rho_\alpha)\right)(1-\rho_\alpha)^{-1} P(\mathcal{Q} > x), \qquad x \to \infty.$$

Remark 6.7 *(i) For PageRank we have* $C_i = c/D_i$ *and* $Q_i = 1-c$, *where* $c \in (0,1)$ *is the damping factor. This leads to a limiting weight distribution of the form*

$$P(\mathcal{C} \le x) = \lim_{n\to\infty} \frac{1}{L_n} \sum_{i=1}^{n} 1(c/D_i \le x) D_i,$$

which is not the limiting distribution of the reciprocal of the out-degrees, $\{c/D_i\}$, *but rather a size-biased version of it.*

(ii) Applying Theorem 6.6 to PageRank when $P(\mathcal{N} > x) \in \mathscr{R}_{-\alpha}$ *and* $P(\mathcal{N}_0 > x) \sim \kappa P(\mathcal{N} > x)$ *for some constant* $\kappa > 0$ *gives that*

$$P(\mathcal{R}^* > x) \sim \kappa' P(\mathcal{N} > x) \qquad \text{as } x \to \infty,$$

where $\kappa' > 0$ *is determined by the theorem.*

(iii) The theorem above only includes two possible cases of the relations between $(\mathcal{N}_0, \mathcal{Q}_0)$ *and* $(\mathcal{N}, \mathcal{Q})$. *The exact asymptotics of* $\mathcal{R}^*$ *can be obtained from those of* $\mathcal{R}$ *in more cases than these using the same techniques; we leave the details to the reader.*

(iv) Theorem 6.6 requires the weights $\{\mathcal{C}_i\}$ *to be nonnegative, which is not a condition in Theorem 6.4. The tail asymptotics of* $\mathcal{R}$, *and therefore of* $\mathcal{R}^*$, *in the real-valued case are unknown.*

7 Algorithm to generate bi-degree sequences

As an example of an extended bi-degree sequence satisfying Assumptions 5.1 and 6.2, we give in this section an algorithm based on sequences of i.i.d. random variables. The method for generating the bi-degree sequence $(\mathbf{N}_n, \mathbf{D}_n)$ is taken from [11], where the goal was to generate a directed random graph with prescribed in- and out-degree distributions.

To define the algorithm we need to first specify target distributions for the in- and out-degrees, which we will denote by $f_k^{\text{in}} = P(\mathscr{N} = k)$, and $f_k^{\text{out}} = P(\mathscr{D} = k)$, $k \ge 0$, respectively. Furthermore, we will assume that these target distributions satisfy $E[\mathscr{N}] = E[\mathscr{D}]$,

$$\overline{F^{\text{in}}}(x) = \sum_{k>x} f_k^{\text{in}} \le x^{-\alpha} L_{\text{in}}(x) \qquad \text{and} \qquad \overline{F^{\text{out}}}(x) = \sum_{k>x} f_k^{\text{out}} \le x^{-\beta} L_{\text{out}}(x),$$

for some slowly varying functions L_{in} and L_{out}, and $\alpha > 1, \beta > 2$. To the original construction given in [11] we will need to add two additional steps to generate the weight and personalization sequences

$\mathbf{C}_n$ and $\mathbf{Q}_n$, for which we need two more distributions $F^\zeta(x) = P(\zeta \le x)$ and $F^Q(x) = P(Q \le x)$ with support on the real line and satisfying

$$P(|\zeta| \le c) = 1 \text{ for some } 0 < c < 1, \quad \text{and} \quad E[|Q|^{1+\epsilon_Q}] < \infty \text{ for some } 0 < \epsilon_Q \le 1.$$

Let

$$\kappa_0 = \min\{1 - \alpha^{-1}, 1/2\}.$$

The IID Algorithm:

1. Fix $0 < \delta_0 < \kappa_0$.
2. Sample an i.i.d. sequence $\{\mathscr{N}_1, \ldots, \mathscr{N}_n\}$ from distribution F^{in}; let $\overline{\mathscr{N}}_n = \sum_{i=1}^n \mathscr{N}_i$.
3. Sample an i.i.d. sequence $\{\mathscr{D}_1, \ldots, \mathscr{D}_n\}$ from distribution F^{out}, independent of $\{\mathscr{N}_i\}$; let $\overline{\mathscr{D}}_n = \sum_{i=1}^n \mathscr{D}_i$.
4. Define $\Delta_n = \overline{\mathscr{N}}_n - \overline{\mathscr{D}}_n$. If $|\Delta_n| \le n^{1-\kappa_0+\delta_0}$ proceed to step 5; otherwise repeat from step 2.
5. Choose randomly $|\Delta_n|$ nodes $\{i_1, i_2, \ldots, i_{|\Delta_n|}\}$ without replacement and let

$$N_i = \begin{cases} \mathscr{N}_i + 1 & \text{if } \Delta_n < 0 \text{ and } i \in \{i_1, i_2, \ldots, i_{|\Delta_n|}\}, \\ \mathscr{N}_i & \text{otherwise,} \end{cases}$$

$$D_i = \begin{cases} \mathscr{D}_i + 1 & \text{if } \Delta_n \ge 0 \text{ and } i \in \{i_1, i_2, \ldots, i_{|\Delta_n|}\}, \\ \mathscr{D}_i & \text{otherwise.} \end{cases}$$

6. Sample an i.i.d. sequence $\{Q_1, \ldots, Q_n\}$ from distribution F^Q, independent of $\{\mathscr{N}_i\}$ and $\{\mathscr{D}_i\}$.
7. Sample an i.i.d. sequence $\{\zeta_1, \ldots, \zeta_n\}$ from distribution F^ζ, independent of $\{\mathscr{N}_i\}$, $\{\mathscr{D}_i\}$ and $\{Q_i\}$, and set $C_i = \zeta_i / D_i$ if $D_i \ge 1$ or $C_i = c\,\text{sgn}(\zeta_i)$ otherwise.

Remark 7.1 *Note that since $E[|\mathscr{N} - \mathscr{D}|^{1+a}] < \infty$ for any $0 < a < \min\{\alpha - 1, \beta - 1\}$, then $E[|\mathscr{N} - \mathscr{D}|^{1+(\kappa_0-\delta_0)/(1-\kappa_0)}] < \infty$, and Corollary 8.4 in Section 8 gives*

$$P\left(|\Delta_n| > n^{1-\kappa_0+\delta_0}\right) = O\left(n^{-\delta_0(\kappa_0-\delta_0)/(1-\kappa_0)}\right) \tag{7.1}$$

as $n \to \infty$.

The two propositions below give the desired properties. Their proofs are given in Section 8.3.

Proposition 7.2 *The extended bi-degree sequence $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$ generated by the IID Algorithm satisfies Assumption 5.1 for any $0 < \kappa < \beta - 2$, any $0 < \gamma < \min\{(\kappa_0 - \delta_0)^2/(1 - \delta_0), (\beta - 2 - \kappa)/\beta\}$, $\mu = \nu_1 = E[\mathscr{N}] = E[\mathscr{D}]$, $\nu_2 = (E[\mathscr{D}])^2$, $\nu_3 = E[\mathscr{D}^2]$, $\nu_4 = E[\mathscr{D}^{2+\kappa}]$, $\nu_5 = E[|\zeta|]P(\mathscr{D} \ge 1)$, $H = E[|Q|] + 1$, and some $\varepsilon > 0$.*

Proposition 7.3 *The extended bi-degree sequence $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$ generated by the IID Algorithm satisfies Assumption 6.2 with*

$$F^*(m, q) = P(\mathscr{N} \le m)P(Q \le q) \qquad \text{and}$$

$$F(m, q, x) = P(\mathscr{N} \le m)P(Q \le q)E[1(\zeta/\mathscr{D} \le x)\mathscr{D}]/\mu.$$

7.1 Numerical examples

To complement the theoretical contribution of the paper, we use the IID Algorithm described in the previous section to provide some numerical results showing the accuracy of the WBP approximation to PageRank. To generate the in- and out-degrees we use the zeta distribution. More precisely, we set

$$\mathscr{N}_i = X_{1,i} + Y_{1,i}, \quad \mathscr{D}_i = X_{2,i} + Y_{2,i},$$

where $\{X_{1,i}\}$ and $\{X_{2,i}\}$ are independent sequences of i.i.d. Zeta random variables with parameters $\alpha+1$ and $\beta+1$, respectively; $\{Y_{1,i}\}$ and $\{Y_{2,i}\}$ are independent sequences of i.i.d. Poisson random variables with different parameters chosen so that $\mathscr{N}$ and $\mathscr{D}$ have equal mean. Note that the Poisson distribution has a light tail so that the power law tail behavior of $\mathscr{N}$ and $\mathscr{D}$ is preserved and determined by α and β, respectively.

Once the sequences $\{\mathscr{N}_i\}$ and $\{\mathscr{D}_i\}$ are generated, we use the IID Algorithm to obtain a valid bi-degree sequence $(\mathbf{N}_n, \mathbf{D}_n)$. Note that in PageRank, we have $\zeta_i = c$ and $Q_i = 1-c$. Given this bi-degree sequence we next proceed to construct the graph and the TBT simultaneously, according to the rules described in Section 5. To compute $\mathbf{R}^{(n,\infty)}$ we perform matrix iterations with $r_0 = 1$ until $\|\mathbf{R}^{(n,k)} - \mathbf{R}^{(n,k-1)}\|_2 < \varepsilon_0$ for some tolerance ε_0. We only generate the TBT for as many generations as it takes to construct the graph, with each generation corresponding to a step in the breadth first graph exploration process. The computation of the root node of the TBT, $\hat{R}^{(n,k)}$ is done recursively starting from the leaves using

$$\hat{R}_{\mathbf{i}}^{(n,0)} = 1 \text{ for } \mathbf{i} \in \hat{A}_k, \qquad \hat{R}_{\mathbf{i}}^{(n,r)} = \sum_{j=1}^{\hat{N}_{\mathbf{i}}} \frac{c}{\hat{D}_{(\mathbf{i},j)}} \hat{R}_{(\mathbf{i},j)}^{(n,r-1)} + 1 - c, \text{ for } \mathbf{i} \in \hat{A}_r,\ 0 \le r < k.$$

To draw a sample from $\mathcal{R}^*$, note that by Proposition 7.3, $\mathcal{R}^*$ in the IID Algorithm has the same distribution as $\mathcal{R}$, i.e., the endogenous solution to the SFPE

$$\mathcal{R} \stackrel{\mathcal{D}}{=} \sum_{i=1}^{\mathscr{N}} \mathcal{C}_i \mathcal{R}_i + 1 - c,$$

where $P(\mathcal{C} \le x) = E[1(c/\mathscr{D} \le x)\mathscr{D}]/\mu$. To sample $\mathcal{R}$ we construct a WBP with generic branching vector $(\mathscr{N}, 1-c, \{\mathcal{C}_i\})$, with the $\{\mathcal{C}_i\}$ i.i.d. and independent of $\mathscr{N}$ and proceed as in the computation of $\hat{R}^{(n,k)}$. To simulate samples of $\mathcal{C}$ we use the acceptance-rejection method.

To show the convergence of $R_1^{(n,\infty)}$ to $\mathcal{R}^*$, we let $n = 10$, 100 and 10000. The values of the other parameters are $\alpha = 1.5$, $\beta = 2.5$, $\mathbb{E}[\mathscr{N}] = \mathbb{E}[\mathscr{D}] = 2$, $c = 0.3$. For the TBT, we simulate up to $k_n = \lfloor \log n \rfloor$ generations. For the WBP, we simulate 10 generations. For each n, we draw 1000 samples of $R_1^{(n,\infty)}$, $R_1^{(n,k_n)}$, $\hat{R}^{(n,k_n)}$ and $\mathcal{R}^*$, respectively, to approximate the distribution of these quantities.

Figure 3 shows the empirical CDFs of 1000 i.i.d. samples of the true PageRank, $R_1^{(n,\infty)}$; finitely many iterations of PageRank, $R_1^{(n,k_n)}$; and the TBT approximation $\hat{R}^{(n,k_n)}$; it also plots the distribution of the limit $\mathcal{R}^*$ using 1000 simulations. The approximations are so accurate that the CDFs are almost indistinguishable. Figure 4 illustrates the weak convergence of PageRank on the graph, $R_1^{(n,\infty)}$, to its limit $\mathcal{R}^*$ as the size of the graph grows.

To quantify the distance between the CDFs, we sort the samples in ascending order and compute the mean squared error (MSE) $\sum_{i=1}^{1000}(x_i^{(n)} - y_i)/1000$, where y_i is the sorted ith sample of $\mathcal{R}^*$ and $x_i^{(n)}$ is the sorted ith sample of $R_1^{(n,\infty)}$. For robustness, we discard the squared error of the maximal value. As a result, the MSEs are 0.2950, 0.1813 and 0.0406 respectively for $n = 10$, 100 and 10000. It is clear that the approximation improves as n increases.

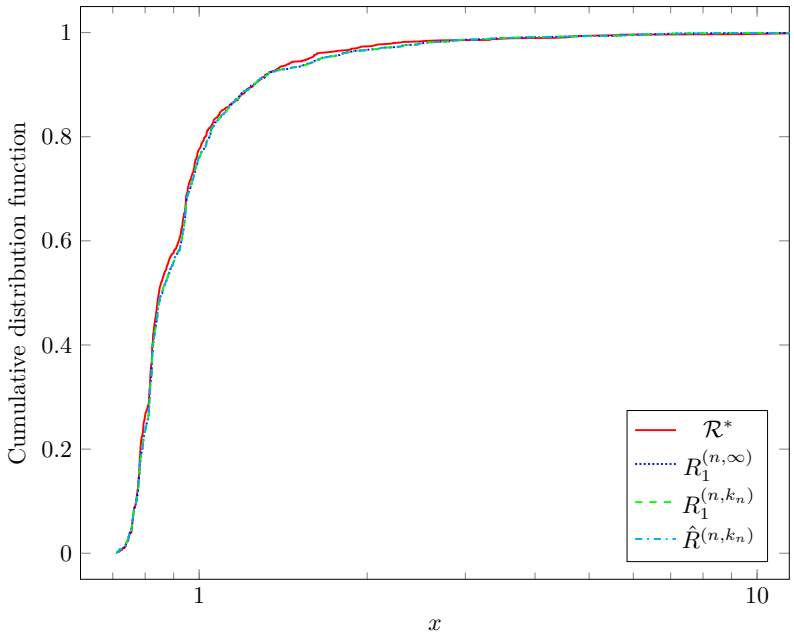


Figure 3: The empirical CDFs of 1000 samples of $\mathcal{R}^*$, $R_1^{(n,\infty)}$, $R_1^{(n,k_n)}$ and $\hat{R}^{(n,k_n)}$ for $n = 10000$ and $k_n = 9$.

8 Proofs

The last section of the paper contains most of the proofs. For the reader's convenience we have organized them in subsections according to the order in which their corresponding statements appear in the paper.

8.1 Proof of the coupling lemma

Recall from Section 5 that $\hat{N}_\emptyset$ denotes the number of offspring of the root node in the TBT (chosen from distribution (5.1)) and $\hat{N}_1$ denotes the number of offspring of a node chosen from distribution (5.2). Throughout this section we will also need to define

$$\mu_n^* = \mathbb{E}_n\left[\hat{N}_\emptyset\right] = \sum_{i,j,s,t} i f_n^*(i,j,s,t) = \frac{1}{n}\sum_{k=1}^{n} N_k = \frac{L_n}{n},$$

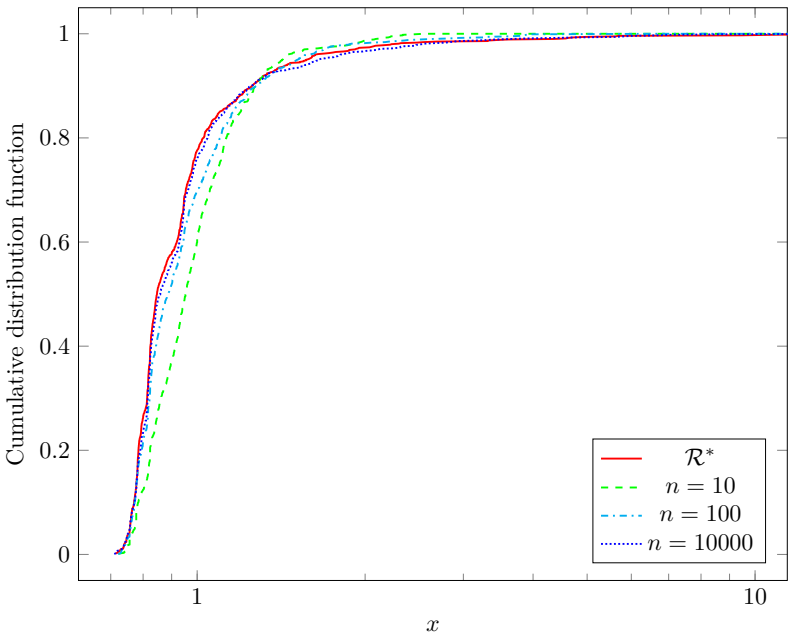


Figure 4: The empirical CDFs of 1000 samples of $\mathcal{R}^*$ and $R_1^{(n,\infty)}$ for $n = 10$, 100 and 10000.

and

$$\mu_n = \mathbb{E}_n\left[\hat{N}_1\right] = \sum_{i,j,s,t} i f_n(i,j,s,t) = \frac{1}{L_n}\sum_{k=1}^{n} N_k D_k.$$

Before we give the proof of the Coupling Lemma 5.4 we will need the following estimates for the growth of the process $\{\hat{Z}_k\}$.

Lemma 8.1 *Suppose* $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{C}_n, \mathbf{Q}_n)$ *satisfies Assumption 5.1 and recall that* $\mu = \nu_2/\nu_1$. *Then, for any constants* $K > 0$, *any nonnegative sequence* $\{x_n\}$ *with* $x_n \to \infty$ *and any* $k = O(n^\gamma)$,

$$P\left(\max_{0\le r\le k} \frac{\hat{Z}_r}{\mu^r} > K x_n \middle| \Omega_n\right) = O\left(x_n^{-1}\right), \qquad n \to \infty.$$

Proof. Start by noting that for any $r = 0, 1, 2, \ldots,$

$$\mathbb{E}_n[\hat{Z}_r] = \mu_n^* \mu_n^r. \tag{8.1}$$

Moreover, on the event Ω_n,

$$\mu_n = \frac{n\nu_2(1 + O(n^{-\gamma}))}{n\nu_1(1 + O(n^{-\gamma}))} = \mu(1 + O(n^{-\gamma})), \qquad \text{and}$$
$$\mu_n^* = \frac{n\nu_1(1 + O(n^{-\gamma}))}{n} = \nu_1(1 + O(n^{-\gamma})).$$

Next, note that conditionally on $\mathscr{F}_n$, the process

$$X_r = \frac{\hat{Z}_r}{\mu_n^* \mu_n^r} = \frac{1}{\mu_n^* \mu_n^r} \sum_{\mathbf{i} \in \hat{A}_{r-1}} \hat{N}_{\mathbf{i}}, \quad r \geq 1, \qquad X_0 = \frac{\hat{N}_\emptyset}{\mu_n^*}$$

is a nonnegative martingale with respect to the filtration $\sigma\left(\mathcal{F}_r \cup \mathscr{F}_n\right)$, where $\mathcal{F}_r = \sigma\left(\hat{N}_{\mathbf{i}} : \mathbf{i} \in \hat{A}_s,\, s \leq r\right)$. Therefore, we can apply Doob's inequality, conditionally on $\mathscr{F}_n$, to obtain

$$\begin{aligned}
P\left(\max_{0 \leq r \leq k} \frac{\hat{Z}_r}{\mu^r} > K x_n \middle| \Omega_n\right) &= P\left(\max_{0 \leq r \leq k} \frac{X_r \mu_n^* \mu_n^r}{\mu^r} > K x_n \middle| \Omega_n\right) \\
&= P\left(\max_{0 \leq r \leq k} X_r \nu_1 (1 + O(n^{-\gamma}))^{r+1} > K x_n \middle| \Omega_n\right) \\
&\leq \frac{1}{P(\Omega_n)} E\left[1(\Omega_n) \mathbb{E}_n\left[1\left(\max_{0 \leq r \leq k} X_r > \frac{K x_n}{\nu_1 (1 + O(n^{-\gamma}))^{k+1}}\right)\right]\right] \\
&\leq \frac{1}{P(\Omega_n)} E\left[1(\Omega_n) \frac{\mathbb{E}_n[X_k] \nu_1 (1 + O(n^{-\gamma}))^{k+1}}{K x_n}\right] \\
&= \frac{\nu_1 (1 + O(n^{-\gamma}))^{k+1}}{K x_n} \qquad (\text{since } \mathbb{E}_n[X_k] = 1).
\end{aligned}$$

Noting that $(1 + O(n^{-\gamma}))^k = e^{O(k n^{-\gamma})} = O(1)$ as $n \to \infty$ gives that this last term is $O(x_n^{-1})$. This completes the proof. ■

We now give the proof of the coupling lemma.

Proof of Lemma 5.4. Start by defining

$$x_n = \begin{cases} (n/\mu^{2k})^{1/2}, & \mu > 1, \\ (n/k^2)^{1/2}, & \mu = 1, \\ n^{1/2}, & \mu < 1, \end{cases} \qquad \text{and} \qquad F_k = \left\{\max_{0 \leq r \leq k} \frac{\hat{Z}_r}{\mu^r} \leq x_n\right\}.$$

Note that $x_n \to \infty$ as $n \to \infty$ for all $1 \leq k \leq h \log n$ when $\mu > 1$ and for all $1 \leq k \leq n^b$, $b < \min\{1/2, \gamma\}$, when $\mu \leq 1$. The constraint $b < \gamma$ will allow us to use Lemma 8.1.

Next, note that the jth inbound stub of node $i \in A_s$ (where the label i refers to the order in which the node was added to the graph during the exploration process) will be the first one to be paired with an outbound stub having label 2 or 3 with probability

$$\frac{1}{L_n}\left(\sum_{r=0}^{s-1} \hat{V}_r + \sum_{t=1}^{i-1} D_t + (j-1)\right) \leq \frac{1}{L_n} \sum_{r=0}^{s} \hat{V}_r =: P_s.$$

It follows that,

$$\begin{aligned}
P(\tau = s | \Omega_n) &\leq P(\tau = s, F_k | \Omega_n) + P(\tau = s, F_k^c | \Omega_n) \\
&\leq P(\mathrm{Bin}(\hat{Z}_s, P_s) \geq 1,\, F_k | \Omega_n) + P(\tau = s, F_k^c | \Omega_n),
\end{aligned}$$

where $\mathrm{Bin}(n,p)$ is a Binomial random variable with parameters (n,p). It follows that if we let $\mathcal{F}_k = \sigma(\hat{Z}_r, \hat{V}_r : 1 \le r \le k)$, then

$$\begin{aligned}
P(\tau \le k|\Omega_n) &= \sum_{s=0}^{k} P(\tau = s|\Omega_n) \\
&\le \sum_{s=0}^{k} \left\{ P\left(\mathrm{Bin}(\hat{Z}_s, P_s) \ge 1,\, F_k \middle|\, \Omega_n\right) + P\left(\tau = s,\, F_k^c |\, \Omega_n\right) \right\} \\
&\le \sum_{s=0}^{k} E\left[1(F_k) P(\mathrm{Bin}(\hat{Z}_s, P_s) \ge 1|\mathcal{F}_k) \middle|\, \Omega_n \right] + P\left(F_k^c |\, \Omega_n\right) \\
&\le \sum_{s=0}^{k} E\left[1(F_k) \hat{Z}_s P_s \middle|\, \Omega_n \right] + P\left(F_k^c |\, \Omega_n\right),
\end{aligned}$$

where in the last step we used Markov's inequality. Now, use the bound for $\hat{Z}_s$ implied by F_k and recall that $|\hat{A}_r| = \hat{Z}_{r-1}$ to obtain

$$\begin{aligned}
E\left[1(F_k) \hat{Z}_s P_s \middle|\, \Omega_n \right] &\le E\left[\mu^s x_n P_s |\, \Omega_n \right] \qquad (8.2) \\
&= \frac{\mu^s x_n}{\nu_1 n} \sum_{r=0}^{s} E\left[\hat{V}_r \middle|\, \Omega_n \right] (1 + O(n^{-\gamma})) \\
&= \frac{\mu^s x_n}{\nu_1 n} \left\{ E\left[\hat{V}_0 \middle|\, \Omega_n \right] + \sum_{r=1}^{s} E\left[\mathbb{E}_n\left[\hat{V}_r | \hat{Z}_{r-1} \right] \middle|\, \Omega_n \right] \right\} (1 + O(n^{-\gamma})) \\
&= \frac{\mu^s x_n}{\nu_1 n} \left\{ E\left[\mu_n^* |\, \Omega_n \right] + \sum_{r=1}^{s} E\left[\hat{Z}_{r-1} \lambda_n \middle|\, \Omega_n \right] \right\} (1 + O(n^{-\gamma})),
\end{aligned}$$

where in the first equality we used that on the set Ω_n we have $L_n = \nu_1 n(1 + O(n^{-\gamma}))$, and on the second equality we used the observation that

$$\mathbb{E}_n\left[\hat{V}_0\right] = \mathbb{E}_n\left[\hat{D}_\emptyset\right] = \mu_n^*, \qquad \mathbb{E}_n\left[\hat{V}_r \middle|\, \hat{Z}_{r-1}\right] = \hat{Z}_{r-1}\lambda_n, \quad r \ge 1,$$

where $\lambda_n = \mathbb{E}_n[\hat{D}_1]$. Moreover, on the set Ω_n we have that

$$\lambda_n = \frac{1}{L_n} \sum_{k=1}^{n} D_k^2 = \frac{n\nu_3(1 + O(n^{-\gamma}))}{n\nu_1(1 + O(n^{-\gamma}))} = \lambda(1 + O(n^{-\gamma})),$$

so we obtain

$$\begin{aligned}
E\left[1(F_k) \hat{Z}_s P_s \middle|\, \Omega_n \right] &\le \frac{\mu^s x_n}{\nu_1 n} \left\{ \nu_1 + \sum_{r=1}^{s} \lambda E\left[\hat{Z}_{r-1} \middle|\, \Omega_n \right] \right\} (1 + O(n^{-\gamma})) \\
&= \frac{\mu^s x_n}{\nu_1 n} \left\{ \nu_1 + \sum_{r=1}^{s} \lambda E\left[\mu_n^* \mu_n^{r-1} \middle|\, \Omega_n \right] \right\} (1 + O(n^{-\gamma})) \qquad \text{(by (8.1))}.
\end{aligned}$$

Using the observation that $E\left[\mu_n^* \mu_n^{r-1} |\, \Omega_n\right] = \nu_1 \mu^{r-1}(1 + O(n^{-\gamma}))^{r-1}$ (see the proof of Lemma 8.1), and the condition $r - 1 < s \le k = O(n^\gamma)$, gives

$$P(\tau \le k|\Omega_n) \le (1 + O(1)) \frac{(\lambda + 1) x_n}{n} \sum_{s=0}^{k} \sum_{r=0}^{s} \mu^{s+r} + P(F_k^c|\Omega_n).$$

Note that we did not compute $E\left[\hat{Z}_s P_s \middle| \Omega_n\right]$ in (8.2) directly, since that would have led to having to compute $\mathbb{E}_n\left[\hat{Z}_{s-1}^2\right]$ and neither $\hat{N}_0$ nor $\hat{N}_1$ are required to have finite second moments in the limit. Now, since by Lemma 8.1 we have that $P(F_k^c|\Omega_n) = O\left(x_n^{-1}\right)$, and

$$\sum_{s=0}^{k}\sum_{r=0}^{s}\mu^{s+r} \leq \begin{cases} \mu^{2(k+1)}/(\mu-1)^2, & \mu > 1, \\ (k+1)(k+2)/2, & \mu = 1, \\ 1/(1-\mu), & \mu < 1, \end{cases}$$

we conclude that

$$P(\tau \leq k|\Omega_n) = \begin{cases} O\left(x_n \mu^{2k} n^{-1} + x_n^{-1}\right) = O\left((n/\mu^{2k})^{-1/2}\right), & \mu > 1, \\ O\left(x_n k^2 n^{-1} + x_n^{-1}\right) = O\left((n/k^2)^{-1/2}\right), & \mu = 1, \\ O\left(x_n n^{-1} + x_n^{-1}\right) = O\left(n^{-1/2}\right), & \mu < 1, \end{cases}$$

as $n \to \infty$. This completes the proof. ■

8.2 Proof of the asymptotic behavior of $\mathcal{R}^*$

We give in this section the proof of Theorem 6.6 which describes the asymptotic behavior of the limit $\mathcal{R}^*$, which is essentially determined by the asymptotic behavior of the endogenous solution $\mathcal{R}$ given in (6.2). The tail behavior of $\mathcal{R}$ is the main focus of the work in [40, 22, 24, 23, 30].

Proof of Theorem 6.6. We consider the case when $\mathcal{N}$ is regularly varying first. By Theorem 3.4 in [30] and the remarks that follow it (see also Theorem 4.1 in [40]),

$$P(\mathcal{R} > x) \sim \frac{(E[\mathcal{Q}]E[\mathcal{C}_1])^\alpha}{(1-\rho)^\alpha(1-\rho_\alpha)} P(\mathcal{N} > x), \qquad x \to \infty,$$

and therefore, $P(\mathcal{R} > x) \in \mathscr{R}_{-\alpha}$. Next, since the $\{\mathcal{C}_i\}$ are i.i.d. and independent of $\mathcal{N}$, Minkowski's inequality gives for any $\beta \geq 1$,

$$E\left[\left(\sum_{i=1}^{\mathcal{N}}\mathcal{C}_i\right)^\beta\right] = E\left[E\left[\left(\sum_{i=1}^{\mathcal{N}}\mathcal{C}_i\right)^\beta \middle| \mathcal{N}\right]\right] \leq E\left[\mathcal{N}^\beta E[\mathcal{C}_1^\beta]\right]. \tag{8.3}$$

Applying Lemma 2.3 in [30] with $\beta = 1+\delta$ gives that $E[|\mathcal{R}|^{1+\delta}] < \infty$ for all $0 < \delta < \alpha - 1$. By conditioning on the filtration $\mathcal{F}_k = \sigma\left((\mathcal{N}_{\mathbf{i}}, \mathcal{C}_{(\mathbf{i},1)}, \mathcal{C}_{(\mathbf{i},2)}, \dots) : \mathbf{i} \in \mathcal{A}_s, s < k\right)$ it can be shown that $E\left[\sum_{\mathbf{i}\in\mathcal{A}_k}\Pi_{\mathbf{i}}\mathcal{Q}_{\mathbf{i}}\right] = \rho^k E[\mathcal{Q}]$, which implies that $E[\mathcal{R}] = (1-\rho)^{-1}E[\mathcal{Q}] > 0$. Also, by Lemma 3.7(2) in [25] we have

$$P\left(\sum_{i=1}^{\mathcal{N}_0}\mathcal{C}_i > x\right) \sim (E[\mathcal{C}_1])^\alpha P(\mathcal{N}_0 > x) \sim \kappa\frac{(1-\rho)^\alpha(1-\rho_\alpha)}{(E[\mathcal{Q}])^\alpha}P(\mathcal{R} > x).$$

Using Theorem A.1 in [30] we conclude that

$$P(\mathcal{R}^* > x) \sim \left(E[\mathcal{N}_0]E[\mathcal{C}_1^\alpha] + \kappa \frac{(1-\rho)^\alpha (1-\rho_\alpha)}{(E[\mathcal{Q}])^\alpha} \left(E[\mathcal{R}]\right)^\alpha \right) P(\mathcal{R} > x)$$
$$\sim \left(E[\mathcal{N}_0]E[\mathcal{C}_1^\alpha] + \kappa(1-\rho_\alpha)\right) \frac{(E[\mathcal{Q}]E[\mathcal{C}_1])^\alpha}{(1-\rho)^\alpha(1-\rho_\alpha)} P(\mathcal{N} > x)$$

as $x \to \infty$.

Now, for the case when $\mathcal{Q}$ is regularly varying, note that $E\left[\left(\sum_{i=1}^{\mathcal{N}} \mathcal{C}_i\right)^{\alpha+\epsilon}\right] < \infty$ by (8.3) and the theorem's assumptions. Then, by Theorem 4.4 in [30] (see also Theorem 4.1 in [40]) we have

$$P(\mathcal{R} > x) \sim (1-\rho_\alpha)^{-1} P(\mathcal{Q} > x), \qquad x \to \infty.$$

The same observations made for the previous case give $E[|\mathcal{R}|^{1+\delta}] < \infty$ for all $0 < \delta < \alpha - 1$. In addition, note that the same argument used above gives $E\left[\left(\sum_{i=1}^{\mathcal{N}_0} \mathcal{C}_i\right)^{\alpha+\epsilon}\right] < \infty$. Also,

$$P\left(\mathcal{Q}_0 > x\right) \sim \kappa P\left(\mathcal{Q} > x\right) \sim \kappa(1-\rho_\alpha) P(\mathcal{R} > x).$$

It follows, by Theorem A.2 in [30], that

$$P\left(\mathcal{R}^* > x\right) \sim \left(E[\mathcal{N}_0]E[\mathcal{C}_1^\alpha] + \kappa(1-\rho_\alpha)\right) P(\mathcal{R} > x)$$
$$\sim \left(E[\mathcal{N}_0]E[\mathcal{C}_1^\alpha] + \kappa(1-\rho_\alpha)\right) (1-\rho_\alpha)^{-1} P(\mathcal{Q} > x)$$

as $x \to \infty$. ■

8.3 Proofs of properties of the *IID Algorithm*

Before giving the proofs of Propositions 7.2 and 7.3 we will need some general results for sequences of i.i.d. random variables, which may be of independent interest. The first result establishes a bound for the sum of the largest order statistics in a sample. The second result is essentially an explicit version of the Weak Law of Large Numbers.

Lemma 8.2 *Let $X_1, X_2, \ldots, X_n$ be i.i.d. nonnegative random variables satisfying $E[X_1^{1+\kappa}] < \infty$ for some $\kappa > 0$, and let $X_{(i)}$ denote the ith smallest observation from the set $\{X_1, X_2, \ldots, X_n\}$. Let $\{\pi_1, \pi_2, \ldots, \pi_n\}$ be any permutation of the set $\{1, 2, \ldots, n\}$. Then, for any $k_n \in \{1, 2, 3, 4, \ldots, n\}$ we have*

$$P\left(\sum_{i=n-k_n+1}^{n} X_{(i)} > n^{1-\gamma}\right) = O\left(k_n^{\kappa/(1+\kappa)} n^{-(\kappa/(1+\kappa)-\gamma)}\right)$$

as $n \to \infty$.

Proof. Note that, by Markov's inequality,

$$P(X_1 > x) \leq E[X_1^{1+\kappa}] x^{-1-\kappa},$$

and therefore,

$$P(X_i > x) \le P(Y_i > x),$$

where $\{Y_1, Y_2, \ldots, Y_n\}$ are i.i.d. Pareto random variables having distribution $G(x) = 1 - (x/b)^{-1-\kappa}$ for $x > b := \left(E[X_1^{1+\kappa}]\right)^{-1/(1+\kappa)}$. We then have that

$$\begin{aligned} P\left(\sum_{i=n-k_n+1}^{n} X_{(i)} > n^{1-\gamma}\right) &\le P\left(\sum_{i=n-k_n+1}^{n} Y_{(i)} > n^{1-\gamma}\right) \\ &\le \frac{1}{n^{1-\gamma}} \sum_{i=n-k_n+1}^{n} E[Y_{(i)}], \end{aligned}$$

where $Y_{(i)}$ is the ith smallest from the set $\{Y_1, Y_2, \ldots, Y_n\}$. Moreover, it is known (see [37], for example) that

$$E[Y_{(i)}] = b \cdot \frac{n!}{(n-i)!} \cdot \frac{\Gamma(n-i+1-(1+\kappa)^{-1})}{\Gamma(n+1-(1+\kappa)^{-1})},$$

where $\Gamma(\cdot)$ is the Gamma function. By Wendel's inequality [44], for any $0 < s < 1$ and $x > 0$,

$$\left(\frac{x}{x+s}\right)^{1-s} \le \frac{\Gamma(x+s)}{x^s \Gamma(x)} \le 1,$$

and therefore, for $i < n$, and $\vartheta = (1+\kappa)^{-1}$,

$$E[Y_{(i)}] \le b \cdot \frac{n!}{\Gamma(n+1-\vartheta)} \cdot \frac{1}{(n-i)^{\vartheta}} \le b\left(\frac{n+1-\vartheta}{n-i}\right)^{\vartheta}.$$

We conclude that

$$\begin{aligned} \frac{1}{n^{1-\gamma}} \sum_{i=n-k_n+1}^{n} E[Y_{(i)}] &\le \frac{b}{n^{1-\gamma}} \left(\sum_{i=n-k_n+1}^{n-1} \left(\frac{n+1-\vartheta}{n-i}\right)^{\vartheta} + \frac{n!\Gamma(1-\vartheta)}{\Gamma(n+1-\vartheta)} \right) \\ &\le \frac{b(n+1-\vartheta)^{\vartheta}}{n^{1-\gamma}} \left(\sum_{i=n-k_n+1}^{n-1} \left(\frac{1}{n-i}\right)^{\vartheta} + \Gamma(1-\vartheta) \right) \\ &\le \frac{b(n+1)^{\vartheta}}{n^{1-\gamma}} \left(\sum_{j=1}^{k_n-1} \int_{j-1}^{j} \frac{1}{t^{\vartheta}}\, dt + \Gamma(1-\vartheta) \right) \\ &= \frac{b(n+1)^{\vartheta}}{n^{1-\gamma}} \left(\frac{(k_n-1)^{1-\vartheta}}{1-\vartheta} + \Gamma(1-\vartheta) \right) \\ &= O\left(\frac{k_n^{1-\vartheta}}{n^{1-\vartheta-\gamma}}\right), \end{aligned}$$

where in the second inequality we used Wendel's inequality. This completes the proof. ■

Lemma 8.3 *Let $\{X_1, X_2, \ldots, X_n\}$ be i.i.d. random variables satisfying $E[|X_1|^{1+\kappa}] < \infty$ for some $\kappa > 0$ and $\mu = E[X_1]$. Set $S_m = X_1 + \cdots + X_m$ and $\theta = \min\{1+\kappa, 2\}$. Then, for any $K > 0$,*

any nonnegative sequence $\{x_n\}$ *such that* $x_n \to \infty$ *as* $n \to \infty$*, and all* $m = o\left(x_n^{1+\kappa}\right)$*, there exists an* $n_0 \geq 1$ *such that for all* $n \geq n_0$ *,*

$$P\left(|S_m - m\mu| > Kx_n\right) \leq E[|X_1|^\theta]\left(\frac{2}{K^2} + 1\right)\frac{m}{x_n^\theta}.$$

Proof. If $\kappa \geq 1$, then Chebyshev's inequality gives, for all $m \geq 1$,

$$P\left(|S_m - m\mu| > Kx_n\right) \leq \frac{m\mathrm{Var}(X_1)}{K^2x_n^2} \leq \frac{mE[|X_1|^2]}{K^2x_n^2} = \frac{mE[|X_1|^\theta]}{K^2x_n^\theta}.$$

Suppose now that $0 < \kappa < 1$ and let $G(t) = P(|X_1| \leq t)$. Set $t = x_n$ and define $P(\tilde{X}_i \leq x) = P(X_i \leq x | X_i \leq t)$, and note that

$$\begin{aligned}
\left|E[\tilde{X}_1] - \mu\right| &= |E[X_1 1(|X_1| \leq t)]/G(t) - \mu| \\
&\leq \frac{1}{G(t)}\left|E[X_1 1(|X_1| \leq t)] - \mu\right| + \frac{|\mu|\overline{G}(t)}{G(t)} \\
&= \frac{1}{G(t)}\left(\left|E[X_1 1(|X_1| > t)]\right| + |\mu|\overline{G}(t)\right) \\
&\leq \frac{1}{G(t)}\left(t\overline{G}(t) + \int_t^\infty \overline{G}(x)dx + |\mu|\overline{G}(t)\right) \\
&\leq \frac{E[|X_1|^{1+\kappa}]}{G(t)}\left(t^{-\kappa} + \int_t^\infty x^{-1-\kappa}\,dx + |\mu|t^{-1-\kappa}\right) \qquad \text{(by Markov's inequality)} \\
&= \frac{E[|X_1|^{1+\kappa}]}{G(t)}\left(\frac{1+\kappa}{\kappa} + |\mu|t^{-1}\right)t^{-\kappa}.
\end{aligned}$$

Then, for sufficiently large n, we obtain that

$$\left|E[\tilde{X}_1] - \mu\right| \leq 2E[|X_1|^{1+\kappa}]\left(\frac{1+\kappa}{\kappa} + |\mu|\right)t^{-\kappa} \triangleq K't^{-\kappa} = K'x_n^{-\kappa}.$$

It follows that for sufficiently large n and $m = o(x_n^{1+\kappa})$,

$$\begin{aligned}
&P\left(|S_m - m\mu| > Kx_n\right) \\
&= P\left(\left|\sum_{i=1}^m(\tilde{X}_i - \mu)\right| > Kx_n\right)G(t)^m + P\left(\left|\sum_{i=1}^m(X_i - \mu)\right| > Kx_n,\ \max_{1\leq i\leq m}|X_i| > t\right) \\
&\leq P\left(\left|\sum_{i=1}^m(\tilde{X}_i - E[\tilde{X}_1])\right| + m\left|E[\tilde{X}_1] - \mu\right| > Kx_n\right)G(t)^m + P\left(\max_{1\leq i\leq m}|X_i| > t\right) \\
&\leq \frac{G(t)^m}{(Kx_n - K'mt^{-\kappa})^2}\cdot m\mathrm{Var}(\tilde{X}_1) + 1 - G(t)^m \qquad \text{(by Chebyshev's inequality)} \\
&\leq \frac{G(t)^m m\mathrm{Var}(\tilde{X}_1)}{K^2x_n^2(1 - mx_n^{-1-\kappa}K'/K)^2} + m\overline{G}(t).
\end{aligned}$$

To estimate $\mathrm{Var}(\tilde{X}_1)$ note that

$$\mathrm{Var}(\tilde{X}_1) \leq E[\tilde{X}_1^2] = \frac{E[X_1^2 1(|X_1| \leq t)]}{G(t)} \leq \frac{E[|X_1|^{1+\kappa}]t^{1-\kappa}}{G(t)},$$

so using Markov's inequality again to estimate $\overline{G}(t)$ gives us

$$\begin{aligned} P\left(|S_m - m\mu| > Kx_n\right) &\le \frac{E[|X_1|^{1+\kappa}]}{K^2(1 - mx_n^{-1-\kappa}K'/K)^2} \cdot \frac{mt^{1-\kappa}}{x_n^2} + \frac{E[|X_1|^{1+\kappa}]m}{t^{1+\kappa}} \\ &= E[|X_1|^{1+\kappa}] \left(\frac{1}{K^2(1 - mx_n^{-1-\kappa}K'/K)^2} + 1 \right) \frac{m}{x_n^{1+\kappa}} \\ &= E[|X_1|^{\theta}] \left(\frac{1}{K^2(1 - mx_n^{-1-\kappa}K'/K)^2} + 1 \right) \frac{m}{x_n^{\theta}}. \end{aligned}$$

This completes the proof. ■

By setting $m = n$ and $x_n = n^{1-\gamma}$ we immediately obtain the following corollary.

Corollary 8.4 *Let $\{X_1, X_2, \ldots, X_n\}$ be i.i.d. random variables satisfying $E[|X_1|^{1+\kappa}] < \infty$ for some $\kappa > 0$ and $\mu = E[X_1]$. Set $S_n = X_1 + \cdots + X_n$. Then, for any $0 \le \gamma < 1 - 1/\theta$, $\theta = \min\{1 + \kappa, 2\}$ and any constant $K > 0$, there exists an $n_0 \ge 1$ such that for all $n \ge n_0$*

$$P\left(|S_n - n\mu| > Kn^{1-\gamma}\right) \le E[|X_1|^{\theta}] \left(\frac{2}{K^2} + 1 \right) n^{-\theta(1-1/\theta-\gamma)}.$$

We now proceed to prove that the extended bi-degree sequence generated by the IID Algorithm satisfies Assumptions 5.1 and 6.2.

Proof of Proposition 7.2. It suffices to show that $P\left(\Omega_{n,i}^c\right) = O(n^{-\varepsilon})$ for some $\varepsilon > 0$ and $i = 1, \ldots, 6$. Throughout the proof let $E_n = \{|\Delta_n| \le n^{1-\kappa_0+\delta_0}\}$ and recall that by (7.1) $P(E_n^c) = O\left(n^{-\delta_0\eta}\right)$, where $\eta = (\kappa_0 - \delta_0)/(1 - \kappa_0)$.

We start with $\Omega_{n,2}$. Let $\nu_2 = (E[\mathscr{D}])^2$ and define $\chi_i = D_i - \mathscr{D}_i$, $\tau_i = N_i - \mathscr{N}_i$. Note that $\chi_i, \tau_i \in \{0, 1\}$ for all $i = 1, \ldots, n$; moreover, either all the $\{\chi_i\}$ or all the $\{\tau_i\}$ are zero, and therefore $\chi_i\tau_j = 0$ for all $1 \le i, j \le n$. We now have

$$\begin{aligned} \left| \sum_{i=1}^n D_i N_i - n\nu_2 \right| &= \left| \sum_{i=1}^n \mathscr{D}_i \mathscr{N}_i - n\nu_2 + \sum_{i=1}^n (\mathscr{D}_i \tau_i + \chi_i \mathscr{N}_i) \right| \\ &\le \left| \sum_{i=1}^n \mathscr{D}_i \mathscr{N}_i - n\nu_2 \right| + \max \left\{ \sum_{i=n-\Delta_n+1}^n \mathscr{D}_{(i)}, \sum_{i=n-\Delta_n+1}^n \mathscr{N}_{(i)} \right\}, \end{aligned}$$

where $\mathscr{D}_{(i)}$ (respectively, $\mathscr{N}_{(i)}$) is the ith smallest value from the set $\{\mathscr{D}_1, \ldots, \mathscr{D}_n\}$ (respectively, $\{\mathscr{N}_1, \ldots, \mathscr{N}_n\}$). Since $|\Delta_n| \le n^{1-\kappa_0+\delta_0}$ on E_n, we have

$$\begin{aligned} P(\Omega_{n,2}^c) &= P\left(\left| \sum_{i=1}^n D_i N_i - n\nu_2 \right| > n^{1-\gamma} \,\middle|\, E_n \right) \\ &\le \frac{1}{P(E_n)} \left\{ P\left(\left| \sum_{i=1}^n \mathscr{D}_i \mathscr{N}_i - n\nu_2 \right| > \frac{n^{1-\gamma}}{2} \right) \right. \\ &\quad \left. + P\left(\sum_{i=n-\lfloor n^{1-\eta(1-\kappa_0)} \rfloor+1}^n \mathscr{D}_{(i)} > \frac{n^{1-\gamma}}{2} \right) + P\left(\sum_{i=n-\lfloor n^{1-\eta(1-\kappa_0)} \rfloor+1}^n \mathscr{N}_{(i)} > \frac{n^{1-\gamma}}{2} \right) \right\}. \end{aligned}$$

Now apply Corollary 8.4 to $X_i = \mathscr{D}_i\mathscr{N}_i$, which satisfies $E[(\mathscr{D}_1\mathscr{N}_1)^{1+\eta}] = E[\mathscr{N}_1^{1+\eta}]E[\mathscr{D}_1^{1+\eta}] < \infty$, to obtain

$$P\left(\left|\sum_{i=1}^{n}\mathscr{D}_i\mathscr{N}_i - n\nu_2\right| > \frac{n^{1-\gamma}}{2}\right) = O\left(n^{-\eta+(1+\eta)\gamma}\right).$$

For the remaining two probabilities use Lemma 8.2 to see that

$$\begin{aligned}
&P\left(\sum_{i=n-\lfloor n^{1-\eta(1-\kappa_0)}\rfloor+1}^{n}\mathscr{D}_{(i)} > \frac{n^{1-\gamma}}{2}\right) + P\left(\sum_{i=n-\lfloor n^{1-\eta(1-\kappa_0)}\rfloor+1}^{n}\mathscr{N}_{(i)} > \frac{n^{1-\gamma}}{2}\right)\\
&= O\left(n^{(1-\eta(1-\kappa_0))\eta/(1+\eta)-(\eta/(1+\eta)-\gamma)}\right)\\
&= O\left(n^{-\eta(\kappa_0-\delta_0)/(1+\eta)+\gamma}\right).
\end{aligned}$$

It follows from these estimates that

$$P(\Omega_{n,2}^c) = O\left(n^{-\eta(\kappa_0-\delta_0)/(1+\eta)+\gamma}\right). \tag{8.4}$$

Next, we can analyze $\Omega_{n,1}, \Omega_{n,3}$ and $\Omega_{n,4}$ by considering the sequence $\{D_i^\vartheta\}$ where ϑ can be taken to be $1, 2$ or $2+\kappa$. Correspondingly, we have $\nu_1 = E[\mathscr{D}]$, $\nu_3 = E[\mathscr{D}^2]$ and $\nu_4 = E[\mathscr{D}^{2+\kappa}]$. Similarly as what was done for $\Omega_{n,2}$, note that

$$\begin{aligned}
\left|\sum_{i=1}^{n}D_i^\vartheta - nE[\mathscr{D}^\vartheta]\right| &\le \left|\sum_{i=1}^{n}\mathscr{D}_i^\vartheta - nE[\mathscr{D}^\vartheta]\right| + \sum_{i=1}^{n}\left((\mathscr{D}_i+\chi_i)^\vartheta - \mathscr{D}_i^\vartheta\right)\\
&\le \left|\sum_{i=1}^{n}\mathscr{D}_i^\vartheta - nE[\mathscr{D}^\vartheta]\right| + \sum_{i=1}^{n}\vartheta(\mathscr{D}_i+1)^{\vartheta-1}\chi_i,
\end{aligned}$$

where we used the inequality $(d+x)^\vartheta - d^\vartheta \le \vartheta(d+1)^{\vartheta-1}x$ for $d \ge 0$, $x \in [0,1]$ and $\vartheta \ge 1$. Now note that $E[(\mathscr{D}^\vartheta)^{1+\sigma}] < \infty$ for any $0 < \sigma < (\beta-2-\kappa)/(2+\kappa)$; in particular, since $\gamma < (\beta-2-\kappa)/\beta$, we can choose $\gamma/(1-\gamma) < \sigma < (\beta-2-\kappa)/(2+\kappa)$. For such σ, Corollary 8.4 gives

$$P\left(\left|\sum_{i=1}^{n}\mathscr{D}_i^\vartheta - nE[\mathscr{D}^\vartheta]\right| > \frac{n^{1-\gamma}}{2}\right) = O\left(n^{-\sigma+(1+\sigma)\gamma}\right).$$

For the term involving the $\{\chi_i\}$ we use again Lemma 8.2 to obtain

$$\begin{aligned}
P\left(\sum_{i=1}^{n}\vartheta(\mathscr{D}_i+1)^{\vartheta-1}\chi_i > \frac{n^{1-\gamma}}{2}\right) &\le P\left(\sum_{i=n-\lceil n^{1-\eta}\rceil+1}^{n}\vartheta(\mathscr{D}_{(i)}+1)^{\vartheta-1} > \frac{n^{1-\gamma}}{2}\right)\\
&= O\left(n^{(1-\eta)(1-1/2)-(1-\gamma-1/2)}\right)\\
&= O\left(n^{-\eta/2+\gamma}\right).
\end{aligned}$$

It follows that

$$P(\Omega_{n,i}^c) \le \frac{1}{P(E_n)}\cdot O\left(n^{-\sigma+(1+\sigma)\gamma} + n^{-\eta/2+\gamma}\right), \qquad i = 1,3,4. \tag{8.5}$$

Now note that since $|\zeta| \leq c < 1$ a.s., then $E[|\zeta|^2] < \infty$ and Corollary 8.4 gives

$$P(\Omega_{n,5}^c) = P\left(\left|\sum_{r=1}^{n} |\zeta_r| 1(D_r \geq 1) - n\nu_5\right| > n^{1-\gamma}\right)$$
$$= P\left(\left|\sum_{r=1}^{n} |\zeta_r| 1(\mathscr{D}_r \geq 1) - n\nu_5\right| + c|\Delta_n| > n^{1-\gamma}\right) = O\left(n^{-1+2\gamma}\right). \tag{8.6}$$

Finally, by Corollary 8.4 and (7.1),

$$P(\Omega_{n,6}^c) \leq P\left(\left|\sum_{r=1}^{n} |Q_r| - nE[|Q|]\right| > n \middle| E_n\right) = O\left(n^{-\epsilon_Q} + n^{-\delta_0\eta}\right). \tag{8.7}$$

Our choice of $0 < \gamma < \min\{\eta(\kappa_0 - \delta_0)(1+\eta),\, \sigma/(1+\sigma)\}$ guarantees that all the exponents of n in expressions (8.4) - (8.6) are strictly negative, which completes the proof. ■

Proof of Proposition 7.3. We will show that $d_1(F_n^*, F^*)$ and $d_1(F_n, F)$ converge to zero a.s. by using the duality formula for the Kantorovich-Rubinstein distance. To this end, let $S_n = \sum_{i=1}^{n} \mathscr{D}_i$, $\mathscr{C}_k = \zeta_k/\mathscr{D}_k 1(\mathscr{D}_k \geq 1) + c\,\mathrm{sgn}(\zeta_k) 1(\mathscr{D}_k = 0)$, and fix $\psi^* : \mathbb{R}^2 \to \mathbb{R}$ and $\psi : \mathbb{R}^3 \to \mathbb{R}$ to be Lipschitz continuous functions with Lipschitz constant one. Then,

$$\mathcal{E}_0 := \left|\frac{1}{n}\sum_{k=1}^{n} \psi^*(N_k, Q_k) - \frac{1}{n}\sum_{k=1}^{n} \psi^*(\mathscr{N}_k, Q_k)\right|$$
$$\leq \frac{1}{n}\sum_{k=1}^{n} |\psi^*(\mathscr{N}_k + 1, Q_k) - \psi^*(\mathscr{N}_k, Q_k)|\, 1(N_k = \mathscr{N}_k + 1)$$
$$\leq \frac{1}{n}\sum_{k=1}^{n} 1(N_k = \mathscr{N}_k + 1) \leq \frac{|\Delta_n|}{n},$$

and

$$\mathcal{E}_1 := \left|\sum_{k=1}^{n} \psi(N_k, Q_k, C_k)\frac{D_k}{L_n} - \sum_{k=1}^{n} \psi(\mathscr{N}_k, Q_k, \mathscr{C}_k)\frac{\mathscr{D}_k}{S_n}\right|$$
$$\leq \sum_{k=1}^{n} \frac{\mathscr{D}_k}{S_n} |\psi(N_k, Q_k, \mathscr{C}_k) - \psi(\mathscr{N}_k, Q_k, \mathscr{C}_k)|\, 1(\Delta_n \leq 0)$$
$$+ \sum_{k=1}^{n} \frac{D_k}{L_n} |\psi(\mathscr{N}_k, Q_k, C_k) - \psi(\mathscr{N}_k, Q_k, \mathscr{C}_k)|\, 1(\Delta_n > 0)$$
$$+ \sum_{k=1}^{n} \left|\psi(\mathscr{N}_k, Q_k, \zeta_k/\mathscr{D}_k)\left(\frac{D_k}{L_n} - \frac{\mathscr{D}_k}{S_n}\right)\right| 1(\Delta_n > 0)$$
$$\leq \sum_{k=1}^{n} \frac{\mathscr{D}_k}{S_n} 1(N_k = \mathscr{N}_k + 1) + \sum_{k=1}^{n} \frac{D_k}{L_n} |\zeta_k/(\mathscr{D}_k + 1) - \mathscr{C}_k|\, 1(D_k = \mathscr{D}_k + 1)$$
$$+ \sum_{k=1}^{n} |\psi(\mathscr{N}_k, Q_k, \mathscr{C}_k)| \left|\frac{(D_k - \mathscr{D}_k)S_n - \mathscr{D}_k\Delta_n}{L_n S_n}\right| 1(\Delta_n > 0),$$

where we used the fact that ψ^* and ψ have Lipschitz constant one. To bound further $\mathcal{E}_1$ use the Cauchy-Schwarz inequality to obtain

$$\sum_{k=1}^{n} \frac{\mathscr{D}_k}{S_n} 1(N_k = \mathscr{N}_k + 1) \leq \frac{n}{S_n} \left(\frac{1}{n} \sum_{k=1}^{n} \mathscr{D}_k^2 \right)^{1/2} \left(\frac{|\Delta_n|}{n} \right)^{1/2}.$$

Now, use the observation that $|\zeta_k| \leq c$ to obtain

$$\begin{aligned}
&\sum_{k=1}^{n} \frac{D_k}{L_n} \left| \zeta_k/(\mathscr{D}_k + 1) - \mathscr{C}_k \right| 1(D_k = \mathscr{D}_k + 1) \\
&\leq c \sum_{k=1}^{n} \frac{1}{L_n \mathscr{D}_k} 1(D_k = \mathscr{D}_k + 1, \mathscr{D}_k \geq 1) + \sum_{k=1}^{n} \frac{1}{L_n} \left| \zeta_k - c \operatorname{sgn}(\zeta_k) \right| 1(D_k = \mathscr{D}_k + 1, \mathscr{D}_k = 0) \\
&\leq \frac{c}{L_n} \sum_{k=1}^{n} 1(D_k = \mathscr{D}_k + 1) \leq \frac{c|\Delta_n|}{S_n}.
\end{aligned}$$

Next, use the bound $|\psi(m, q, x)| \leq ||(m, q, x)||_1 + |\psi(0, 0, 0)|$ and Hölder's inequality to obtain

$$\begin{aligned}
&\sum_{k=1}^{n} |\psi(\mathscr{N}_k, Q_k, \mathscr{C}_k)| \left| \frac{(D_k - \mathscr{D}_k) S_n - \mathscr{D}_k \Delta_n}{L_n S_n} \right| 1(\Delta_n > 0) \\
&\leq \sum_{k=1}^{n} |\psi(\mathscr{N}_k, Q_k, \mathscr{C}_k)| \frac{1(D_k = \mathscr{D}_k + 1)}{S_n} + \sum_{k=1}^{n} |\psi(\mathscr{N}_k, Q_k, \mathscr{C}_k)| \frac{\mathscr{D}_k |\Delta_n|}{S_n^2} \\
&\leq \frac{1}{S_n} \sum_{k=1}^{n} ||(\mathscr{N}_k, Q_k, c)||_1 \, 1(D_k = \mathscr{D}_k + 1) + \frac{|\Delta_n|}{S_n^2} \sum_{k=1}^{n} (\mathscr{N}_k \mathscr{D}_k + |Q_k| \mathscr{D}_k + c) + \frac{2|\psi(0, 0, 0) \Delta_n|}{S_n} \\
&\leq \frac{n}{S_n} \left\{ \left(\frac{1}{n} \sum_{k=1}^{n} \mathscr{N}_k^{1+\delta} \right)^{1/(1+\delta)} + \left(\frac{1}{n} \sum_{k=1}^{n} |Q_k|^{1+\delta} \right)^{1/(1+\delta)} \right\} \left(\frac{|\Delta_n|}{n} \right)^{\delta/(1+\delta)} \\
&\quad + \frac{|\Delta_n|}{S_n^2} \sum_{k=1}^{n} (\mathscr{N}_k \mathscr{D}_k + |Q_k| \mathscr{D}_k) + \frac{H|\Delta_n|}{S_n},
\end{aligned}$$

where $0 < \delta < \min\{\alpha - 1, \epsilon_Q\}$ and $H = 2|\psi(0, 0, 0)| + 2c$. Now note that since the bi-degree sequence is constructed on the event $|\Delta_n| \leq n^{1-\kappa_0+\delta_0}$, we have that $\mathcal{E}_0 \leq n^{-\kappa_0+\delta_0}$ a.s. To show that $\mathcal{E}_1$ converges to zero a.s. use the Strong Law of Large Numbers (SLLN) (recall that $E[\mathscr{D}^2] < \infty$ and that $\mathscr{N}, \mathscr{D}, Q$ are mutually independent) and the bounds derived above.

Finally, by the SLLN again and the fact that $E[||(\mathscr{N}, Q, \mathscr{C})||_1] < \infty$, we have

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \psi^*(N_k, Q_k) = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \psi^*(\mathscr{N}_k, Q_k) = E[\psi^*(\mathscr{N}, Q)] \quad \text{a.s.}$$

and

$$\lim_{n\to\infty} \sum_{i=1}^{n} \psi(N_k, Q_k, C_k) \frac{\mathscr{D}_i}{S_n} = \lim_{n\to\infty} \sum_{k=1}^{n} \psi(\mathscr{N}_k, Q_k, \mathscr{C}_k) \frac{\mathscr{D}_k}{S_n} = \frac{1}{\mu} E[\psi(\mathscr{N}, Q, \mathscr{C}) \mathscr{D}] \quad \text{a.s.}$$

The first limit combined with the duality formula gives that $d_1(F_n^*, F^*) \to 0$ a.s. For the second limit we still need to identify the limiting distribution, for which we note that

$$\frac{1}{\mu}E[\psi(\mathscr{N}, Q, \mathscr{C})\mathscr{D}] = \frac{1}{\mu}E\left[E[\psi(\mathscr{N}, Q, \mathscr{C})\mathscr{D}|\mathscr{N}, Q]\right] = \frac{1}{\mu}E\left[\sum_{i=1}^{\infty}\int_{-\infty}^{\infty}\psi(\mathscr{N}, Q, z/i)i\, dF^{\zeta}(z)P(\mathscr{D}=i)\right]$$
$$= \frac{1}{\mu}E\left[\sum_{i=1}^{\infty}\int_{-\infty}^{\infty}\psi(\mathscr{N}, Q, y)i\, dF^{\zeta}(yi)P(\mathscr{D}=i)\right] =: E\left[\psi(\mathscr{N}, Q, Y)\right],$$

where Y has distribution function

$$P(Y \le x) = \frac{1}{\mu}E\left[\sum_{i=1}^{\infty}\int_{-\infty}^{\infty}1(y \le x)i\, dF^{\zeta}(yi)P(\mathscr{D}=i)\right] = \frac{1}{\mu}E\left[\sum_{i=1}^{\infty}iF^{\zeta}(ix)P(\mathscr{D}=i)\right]$$
$$= \frac{1}{\mu}E[\mathscr{D}F^{\zeta}(\mathscr{D}x)] = \frac{1}{\mu}E[\mathscr{D}1(\zeta/\mathscr{D} \le x)] = P(\mathcal{C} \le x).$$

It follows that $E[\psi(\mathscr{N}, Q, \mathscr{C})\mathscr{D}]/\mu = E[\psi(\mathscr{N}, Q, \mathcal{C})]$, which combined with the duality formula gives that $d_1(F_n, F) \to 0$ a.s. ■

References

[1] G. Alsmeyer, J.D. Biggins, and M. Meiners. The functional equation of the smoothing transform. *Ann. Probab.*, 40(5):2069–2105, 2012.

[2] G. Alsmeyer, E. Damek, and S. Mentemeier. Tails of fixed points of the two-sided smoothing transform. In *Springer Proceedings in Mathematics & Statistics: Random Matrices and Iterated Random Functions*, 2012.

[3] G. Alsmeyer and M. Meiners. Fixed points of inhomogeneous smoothing transforms. *Journal of Difference Equations and Applications*, 18(8):1287–1304, 2012.

[4] G. Alsmeyer and M. Meiners. Fixed points of the smoothing transform: Two-sided solutions. *Probab. Theory Rel.*, 155(1-2):165–199, 2013.

[5] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using PageRank vectors. In *Proceedings of FOCS2006*, pages 475–486, 2006.

[6] S. Bhamidi, J.M. Steele, and T. Zaman. Twitter event networks and the superstar model. *Preprint*, 2012. arXiv: 0902.0885.

[7] P. Boldi and S. Vigna. Axioms for centrality. *To Appear in Internet Mathematics*, 2014.

[8] B. Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316, 1980.

[9] B. Bollobás, O. Riordan, J. Spencer, G. Tusnády, et al. The degree sequence of a scale-free random graph process. *Random Structures & Algorithms*, 18(3):279–290, 2001.

[10] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Comput. Networks ISDN Systems*, 30(1-7):107–117, 1998.

[11] N. Chen and M. Olvera-Cravioto. Directed random graphs with given degree distributions. *Stochastic Systems*, 3(1):147–186, 2013.

[12] N. Chen and M. Olvera-Cravioto. Coupling on weighted branching trees. `http://www.columbia.edu/~mo2291/Coupling_Chen_Olv.pdf`, 2014. Preprint.

[13] P. Chen, H. Xie, S. Maslov, and S. Redner. Finding scientific gems with Google's PageRank algorithm. *Journal of Informetrics*, 1(1):8–15, 2007.

[14] F. Chung and L. Lu. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics*, 6(2):125–145, 2002.

[15] P.G. Constantine and D.F. Gleich. Random alpha pagerank. *Internet Mathematics*, 6(2):189–236, 2009.

[16] R. Durret and T. Liggett. Fixed points of the smoothing transformation. *Z. Wahrsch. verw. Gebeite*, 64:275–301, 1983.

[17] R. Durrett. *Random graph dynamics*. Cambridge Series in Statistics and Probabilistic Mathematics. Cambridge university press Cambridge, 2007.

[18] S. Fortunato, M. Boguñá, A. Flammini, and F. Menczer. Approximating PageRank from in-degree. In *Algorithms and Models for the Web-Graph*, pages 59–71. Springer, 2008.

[19] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating Web spam with TrustRank. In *Proceeding of VLDB2004*, pages 576–587, 2004.

[20] T. H. Haveliwala. Topic-sensitive PageRank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM, 2002.

[21] R. Holley and T. Liggett. Generalized potlatch and smoothing processes. *Z. Wahrsch. verw. Gebeite*, 55:165–195, 1981.

[22] P.R. Jelenković and M. Olvera-Cravioto. Information ranking and power laws on trees. *Adv. Appl. Prob.*, 42(4):1057–1093, 2010.

[23] P.R. Jelenković and M. Olvera-Cravioto. Implicit renewal theorem for trees with general weights. *Stoch. Proc. Appl.*, 122(9):3209–3238, 2012.

[24] P.R. Jelenković and M. Olvera-Cravioto. Implicit renewal theory and power tails on trees. *Adv. Appl. Prob.*, 44(2):528–561, 2012.

[25] A.H. Jessen and T. Mikosch. Regularly varying functions. *Publications de L'Institut Mathematique, Nouvelle Serie*, 80(94):171–192, 2006.

[26] and D. Lebedev K. Avrachenkov. PageRank of scale-free growing networks. *Internet Mathematics*, 3(2):207–231, 2006.

[27] A.N. Langville and C.D. Meyer. *Google's PageRank and beyond: The science of search engine rankings*. Princeton University Press, 2011.

[28] N. Litvak, W.R.W. Scheinhardt, and Y. Volkovich. In-degree and PageRank: Why do they follow similar power laws? *Internet Math.*, 4(2-3):175–198, 2007.

[29] Mark Newman. *Networks: an introduction*. Oxford University Press, 2010.

[30] M. Olvera-Cravioto. Tail behavior of solutions of linear recursions on trees. *Stochastic Processes and their Applications*, 122(4):1777–1807, 2012.

[31] G. Pandurangan, P. Raghavan, and E. Upfal. Using pagerank to characterize web structure. In *Computing and Combinatorics*, pages 330–339. Springer, 2002.

[32] U. Rösler. The weighted branching process. *Dynamics of complex and irregular systems (Bielefeld, 1991)*, pages 154–165, 1993. Bielefeld Encounters in Mathematics and Physics VIII, World Science Publishing, River Edge, NJ.

[33] U. Rösler, V.A. Topchii, and V.A. Vatutin. Convergence conditions for the weighted branching process. *Discrete Math. Appl.*, 10(1):5–21, 2000.

[34] U. Rösler, V.A. Topchii, and V.A. Vatutin. The rate of convergence for weighted branching processes. *Siberian Adv. Math.*, 12(4):57–82, 2002.

[35] R. van der Hofstad. Random graphs and complex networks. `http://www.win.tue.nl/rhofstad/NotesRGCN.pdf`, 2014.

[36] R. van der Hofstad, G. Hooghiemstra, and P. Van Mieghem. Distances in random graphs with finite variance degrees. *Random Structures and Algorithms*, 27(1):76–123, 2005.

[37] K. Vännman. Estimators based on order statistics from a Pareto distribution. *Journal of the American Statistical Association*, 71(355):704–708, 1976.

[38] S. Vigna. A weighted correlation index for rankings with ties. *Preprint*, 2014. arXiv: 1404.3325.

[39] C. Villani. *Optimal transport, old and new*. Springer, New York, 2009.

[40] Y. Volkovich and N. Litvak. Asymptotic analysis for personalized web search. *Adv. Appl. Prob.*, 42(2):577–604, 2010.

[41] Y. Volkovich, N. Litvak, and D. Donato. Determining factors behind the PageRank log-log plot. In *Proceedings of the 5th International Conference on Algorithms and Models for the Web-graph*, pages 108–123, 2007.

[42] Y. Volkovich, N. Litvak, and B. Zwart. Extremal dependencies and rank correlations in power law networks. In *Complex Sciences*, pages 1642–1653. Springer, 2009.

[43] L. Waltman and N.J. van Eck. The relation between eigenfactor, audience factor, and influence weight. *J. Am. Soc. Inf. Sci.*, 61(7):1476–1486, 2010.

[44] J.G. Wendel. Note on the gamma function. *Amer. Math. Monthly*, 55(9):563–564, 1948.

Distributed Frameworks for Alternating Least Squares

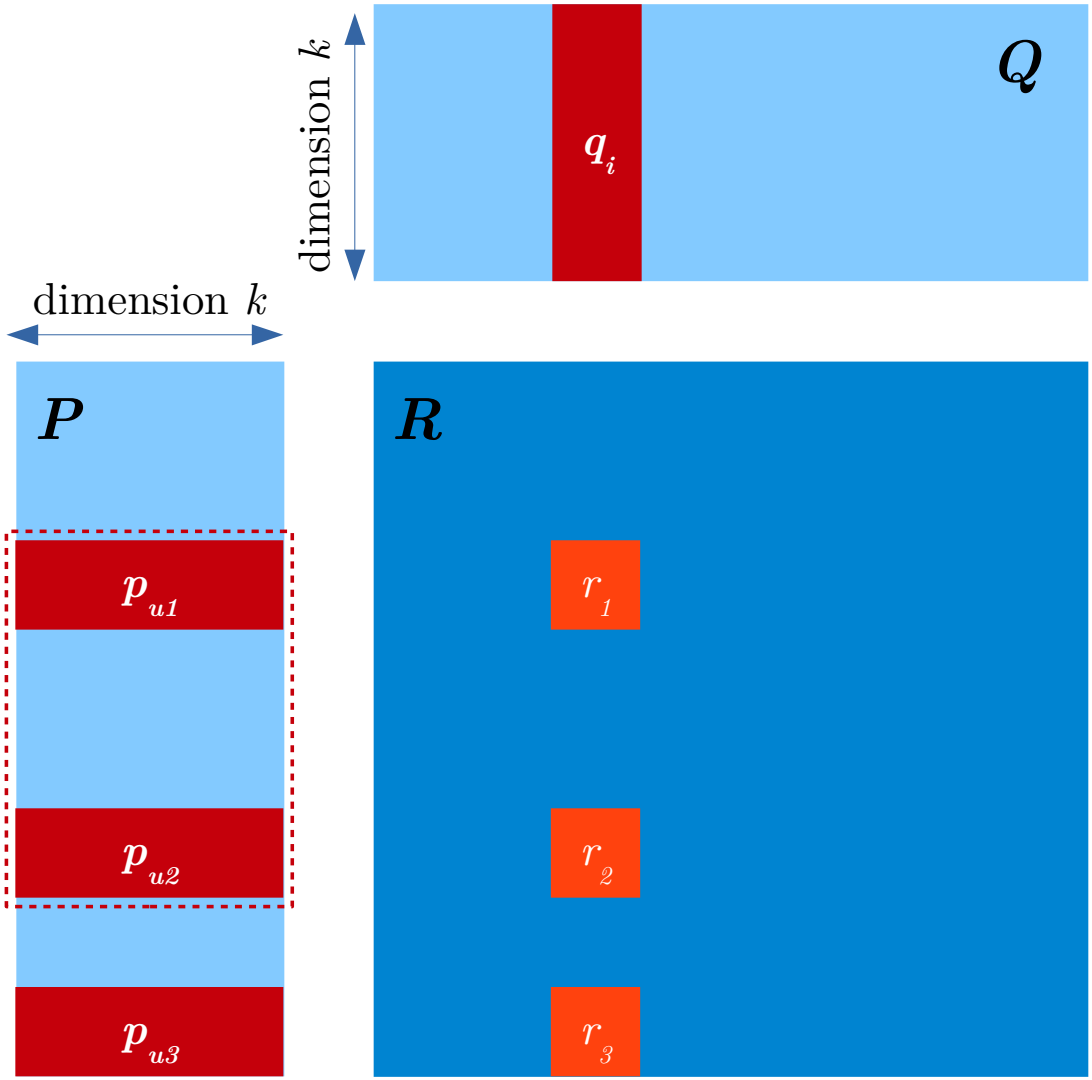
Márton Balassi **Róbert Pálovics** András A. Benczúr
{mbalassi, rpalovics, benczur}@ilab.sztaki.hu

Informatics Laboratory, Department of Computer and Automation Research Institute, Hungarian Academy of Sciences

The publication was supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956)

8th ACM Conference on Recommender Systems
Foster City, Silicon Valley, USA, 6th-10th October 2014

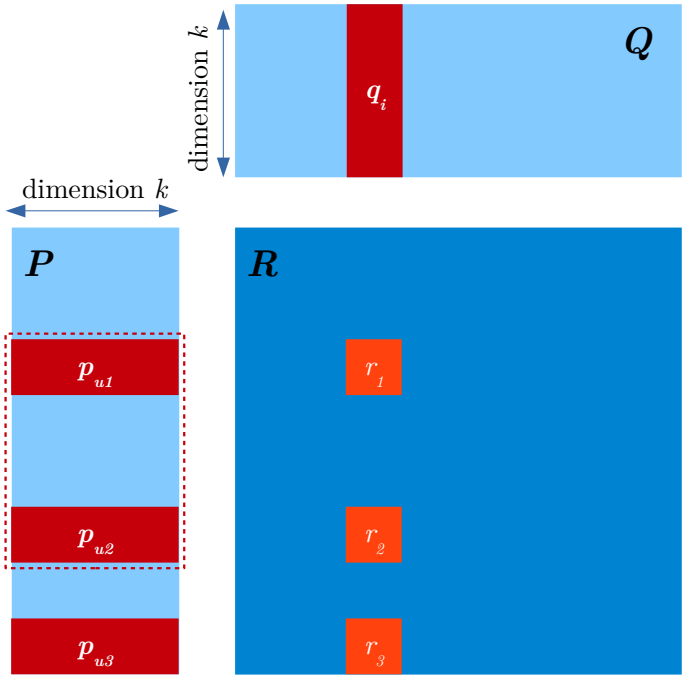
Alternating Least Squares



Alternating Least Squares

$$f_{RMSE}(P,Q) = \sum_{(u,i)\in \text{Training}} (R_{ui} - p_u \cdot q_i^T)^2 + \lambda \cdot (\|P\|_F^2 + \|Q\|_F^2)$$

- Update step for Q: $Q_i \leftarrow (P^T P)^{-1} P^T R_i$
- For each nonzero rating we communicate $(P^T P)^{-1}$ of dim k^2

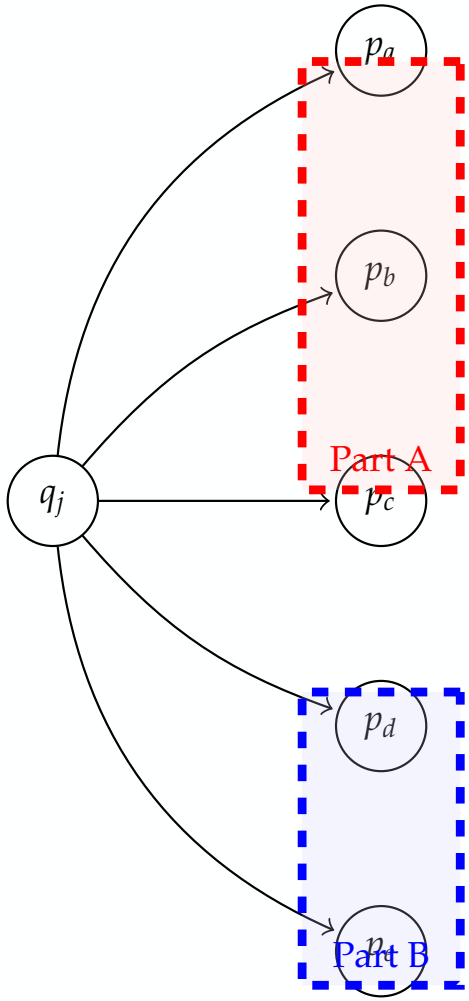


ALS MULTI-MACHINE NO SHARED MEMORY

- Goal: efficient ALS *and* models for other algorithms
- Problem: Large amount of communication alternating between rows and columns
  - ALS message size is quadratic in number of latent factors
- Drawback of "think as a node" philosophy
  - Repeat the same message for all graph nodes
  - Even if they reside on the same server

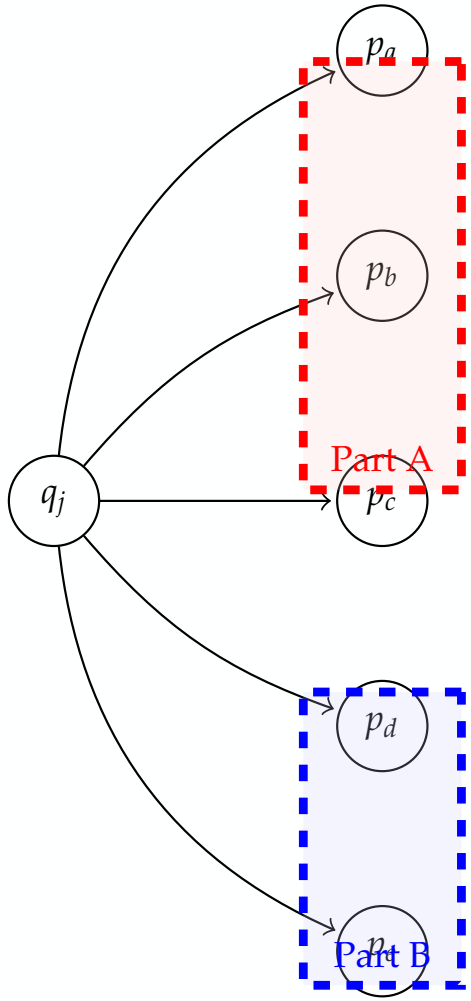
Distribution overhead

- Partitioned graph or ratings matrix
- Naive approach: q_j communicates to each p_i individually
- In ALS, PageRank, ..., messages from q_j are identical
- **Network communication becomes the bottleneck.**



Proposed solution

- Efficient communication between partitions
- Translated to graph processing this is just a *multicast*.



Big Data Frameworks

- Big Data frameworks lack an operator for this job.

Hadoop	(Mahout) Map, Reduce
Spark	"Functional" operators on (memory) Resilient Distributed Datasets
Flink	"Functional" operators and iteration
	Our experimental platform

- **Notion of the partition hidden from user when implementing ALS by vector-to-vector communication.**

Big Data Frameworks - Solution

- Mahout implementation: “CustomALS”.
- Algorithm provides an artificial partition ID
- Map-Reduce grouped by partition ID, expected one partition per reducer
- Partitioning to minimize the communication between partitions **not ensured** but left for the framework

Graph processing engines

- Bulk Synchronous Parallel (BSP)
  - Sends along ALL nonzero ratings
  - Even if the message is identical
  - This issue holds even for PageRank
- Example: Giraph
  - “Think like a vertex”, no partition notion
  - No multicast support in framework

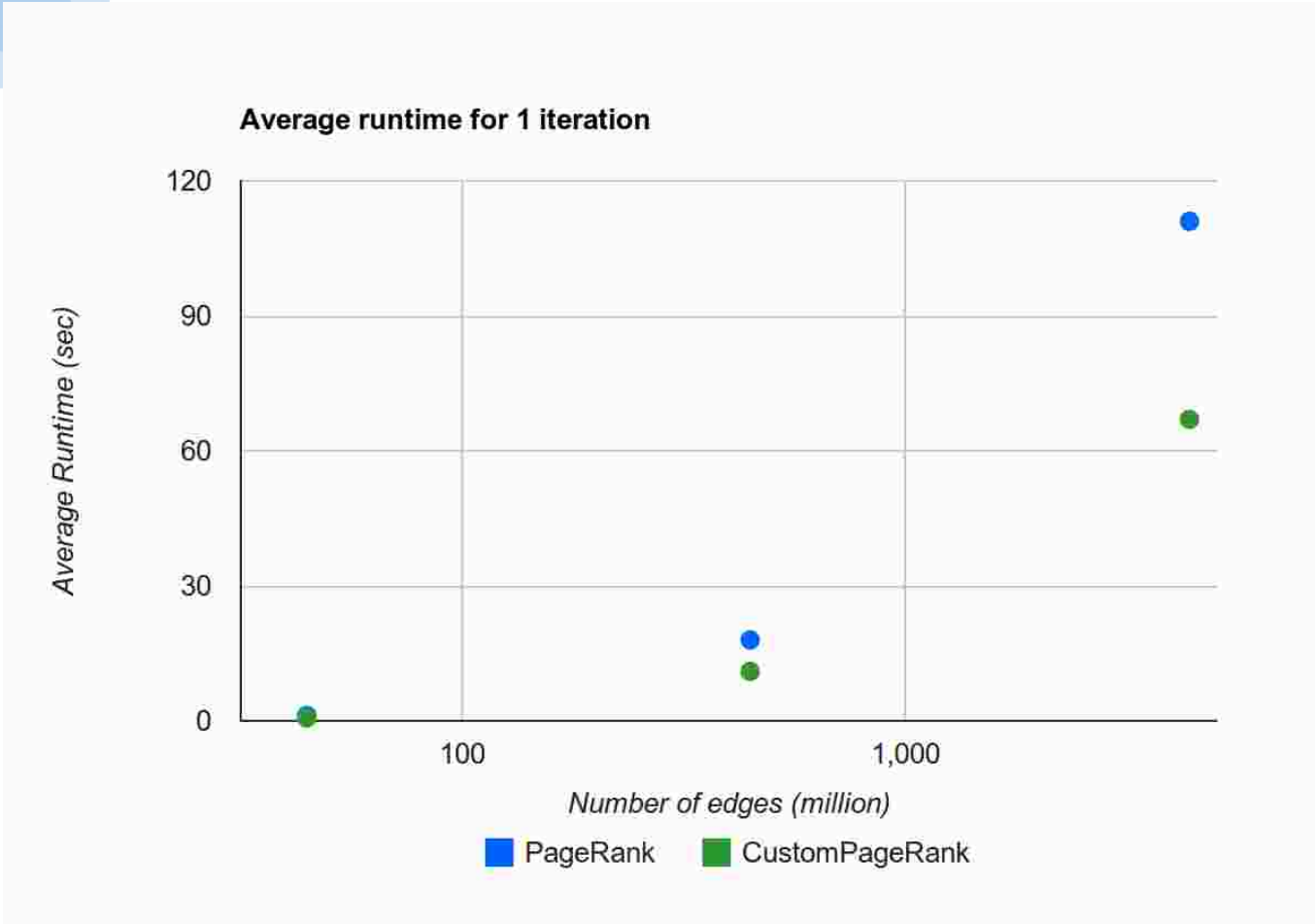
Distributed GraphLab

- Several optimization over plain BSP:
  - Framework support to distribute very high degree nodes: PowerGraph partitions scatters and gathers
  - Optimization: emit unchanged information by caching on gather side
  - Optimization: graph partitioning to reduce number of edges cut (hard to partition a real implicit ratings matrix)
- **But no handling for multiple identical messages**

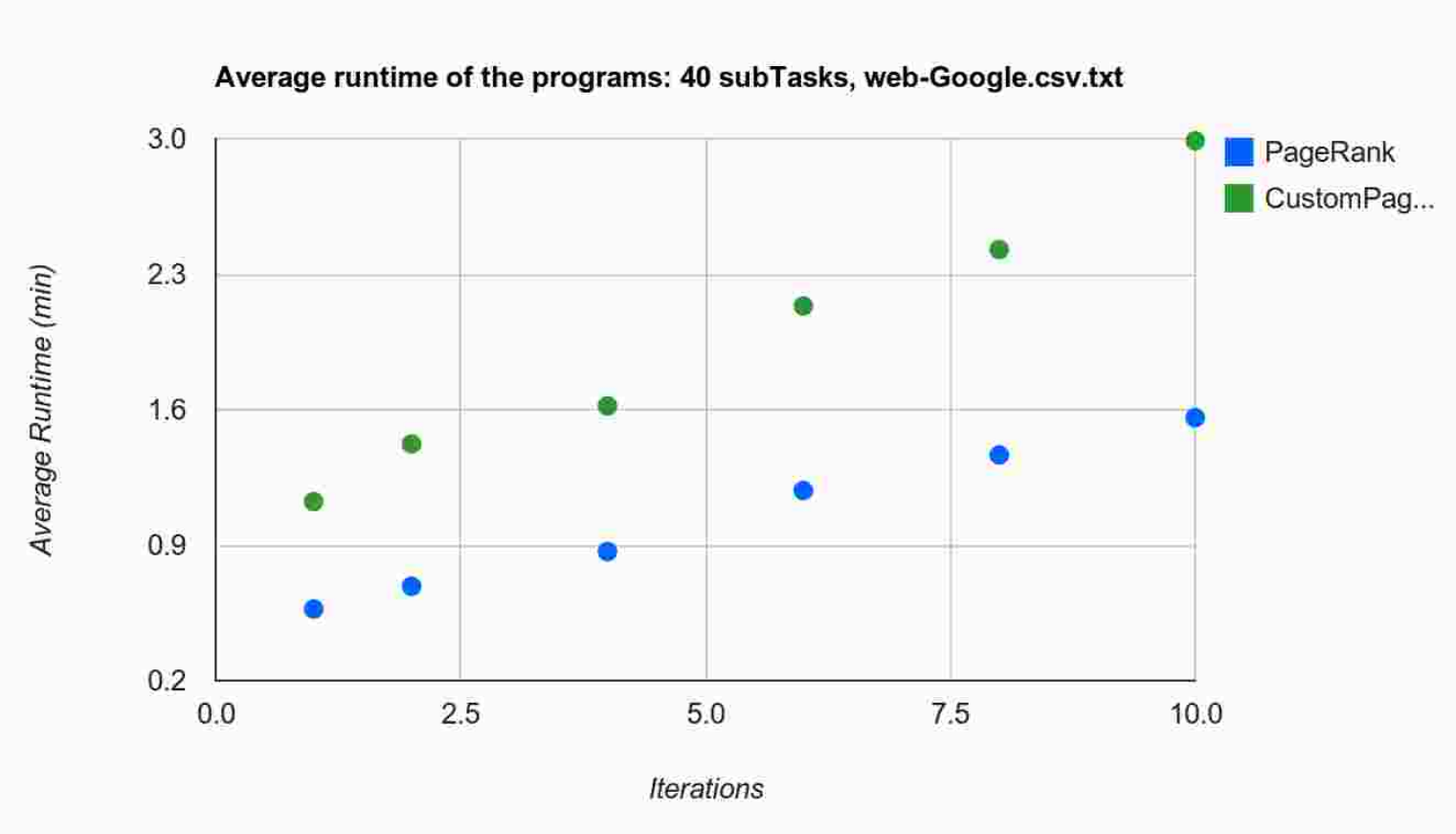
Experiments – distributed message passing in C++

- Proof of concept for a low communication task: PageRank
- We rely on direct control over partitions
- Each vertex sends the message to relevant partitions once
- Test on large Web crawl (`.pt`): 300M nodes, 1B edges
- Significant improvement

CUSTOM PAGERANK IN C++

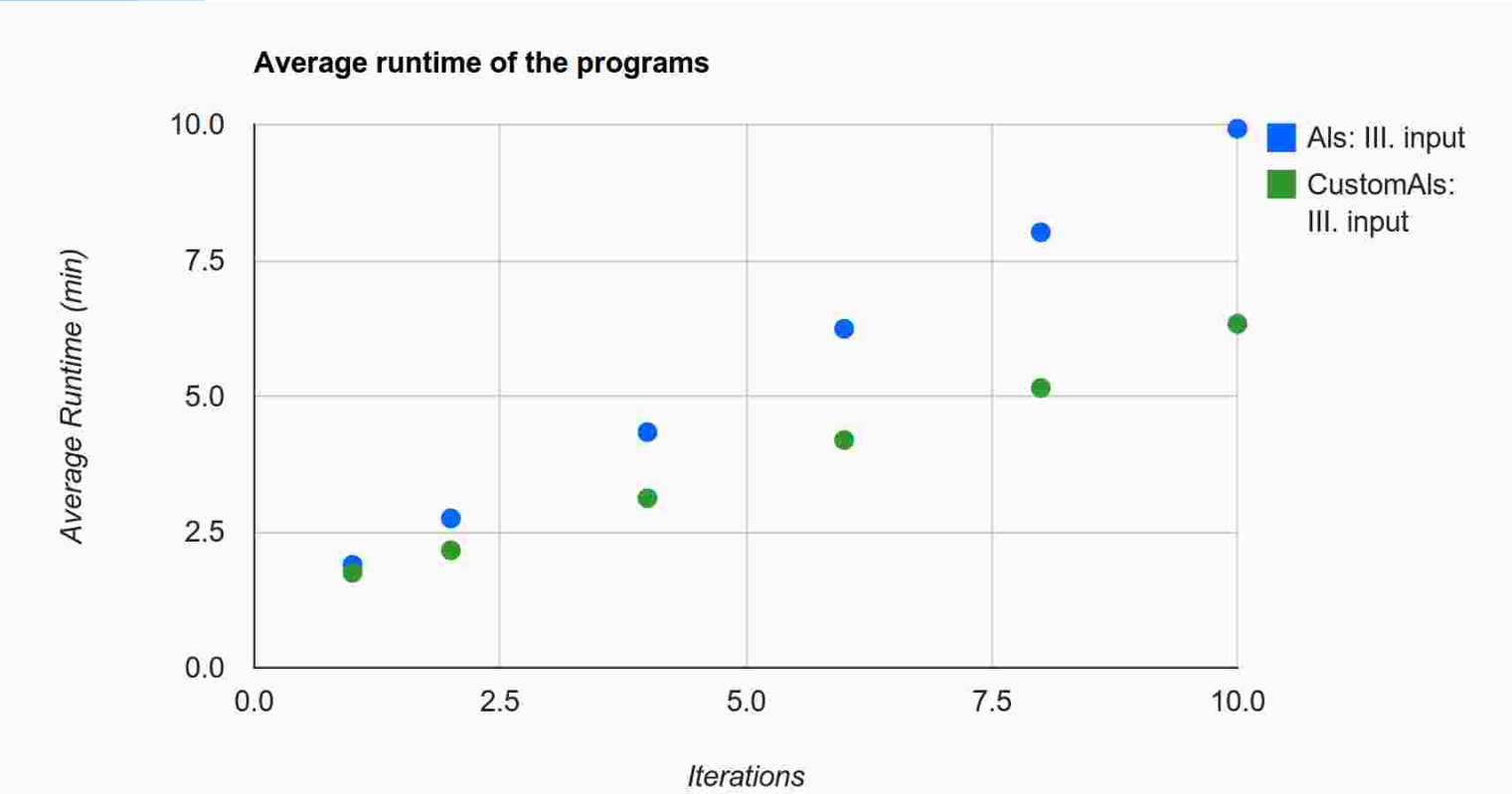


CustomPageRank in Apache Flink



- We define hypernodes – Mahout CustomALS style
- Insufficient for low communication tasks
- Web-Google graph from Stanford Large Network Dataset Collection, $9 \cdot 10^5$ nodes,

CustomALS in Apache Flink



- Generated test data 15 million ratings (courtesy: Gravity)
- Framework support already sufficient for ALS

Conclusions

- ALS multi-machine no shared memory
  - Heavy communication alternating between rows and columns
  - ALS message size is quadratic in number of latent factors
  - Affects MapReduce with no permanent storage (Mahout "CustomALS")
  - Graph parallel frameworks with nonzero ratings mapped to edges
- Ongoing experiments with Message Passing, Giraph, Apache Flink, and its Pregel implementation Spargel.
  - Communication primitives to bind identical messages - use multicast
  - Promising even for seemingly low communication intense algorithms such as PageRank.

Similarity Kernel Learning

Bálint Daróczy[1] Krisztian Buza[2] András A. Benczúr[1]
[1]Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI)
[2]BioIntelligence Lab, Institute of Genomic Medicine and Rare Disorders, Semmelweis University
{daroczyb,benczur}@ilab.sztaki.hu, buza@biointelligence.hu

ABSTRACT

Kernel methods are popular in machine learning tasks. For Support Vector Machine classification or Support Vector Regression, the central question is the selection of the appropriate kernel. The task is difficult in particular if the data points have complex or multimodal attributes such as time series or visual content enhanced with geographic, numeric or text metadata. Unlike earlier approaches of the so-called Multiple Kernel Learning problem, where a large number of kernels are fused by wrapper methods as part of the optimization process, in this paper we mathematically derive an optimal kernel for the data set in question. We begin with selecting appropriate distances for the appropriate modalities, for example dynamic time warping distance for time series and Jensen-Shannon distance for the bag of words text representation. Our kernel is defined, without needs of wrapper methods, by considering the distances as attributes generated by a Markov Random Field. For the Markov Random Field, the natural kernel is based on the Fisher information matrix and its exact form can be computed from the data. We experiment with the above similarity kernel over a wide variety of data sets, including

- 64-channel EEG data;
- General time series data sets;
- Images with text annotations;
- Web documents;
- Gene expression levels.

Over the complex, multimodal or multiple time series classification tasks, our method outperforms the state of the art while reaching identical performance even over the simple unimodal problems as well, hence our method seems applicable under very general settings.

General Terms

Kernel methods, Classification, Mining rich data types, Similarity-based methods, Bioinformatics, Web mining

Keywords

Fisher information matrix, Support Vector Machine

1. INTRODUCTION

Kernel methods [47] are popular in various fields of data mining and knowledge discovery such as classification, regression, clustering or dimensionality reduction. Its numerous applications range from relation extraction [57] to the prediction of protein-protein interactions [4] and other problems in computational biology [45].

While kernel methods are well-founded from the theoretical point of view, the selection of the appropriate kernel is essential in many real-world tasks. In order to allow wide range of applications, various kernels have been introduced in the last decades such as the general-purpose polynomial and RBF-kernels as well as application-specific kernels, see e.g. string-kernels in text mining [6] or computational biology [31]. Learning optimal hyperparameters of these kernels may be computationally prohibitive in case of large datasets. Furthermore, even if the best hyperparameters have been found, the resulting kernel may not completely reflect the true structure of the data, which is likely to manifest in suboptimal results regardless of the particular analysis task.

The selection of feature set dependent distance or similarity metrics is crucial for learning. Although selecting and in some cases computing the potential metrics may constitute a challenging task, once metrics are defined, they can often be used to transform the original complex optimization problem to a less challenging one. Most notably, the Support Vector Machine (SVM) optimization phase is independent of the underlying metric based on precomputed kernel values.

An additional and interesting opportunity arise from the freedom of selecting similarity or distance metrics to define SVM kernels. In a number of practical applications such as image or document classification, we have to learn over multiple representations, often with different kernel functions. Images are often enriched by text description or other non-visual metadata such as geo-location or date, yielding a multimodal classification task with each mode (visual, text, geospatial) having its own natural metric [19]. Another example is Web content, where text, hyperlinks and style give us different kernels when categorizing Web pages or filtering Web spam [10].

In order to address the kernel selection problem, in this paper, we propose a principled meta-kernel learning approach based on Fisher information theory. Our new approach is computationally inexpensive and needs no wrapper methods

for learning a kernel over multiple modalities. In experiments on publicly-available real-world datasets from various domains such as classification of images, texts, time series and gene expression data, we show that our approach outperforms the state-of-the-art.

2. RELATED WORK

In many cases, one single kernel may perform suboptimally. In the last decade, this issue has primarily been addressed in the framework of multiple kernel learning (MKL) [3, 30, 49, 23]. With proposing a method to learn a kernel over multiple modalities, in this paper, we address a problem that is related to MKL, but is substantially different from MKL in several respects. First, we assume that all the modalities are used in the kernel, not only a fraction of them. Second, in order to devise a computationally efficient approach, we only calculate the distance between each instance and a small set of reference instances. This is in contrast to MKL techniques that require full kernel matrices. Last, but not least, our approach runs only one SVM optimization procedure while most MKL approaches are wrapper approaches and therefore they execute large amount of SVM optimizations.

Selecting the appropriate kernel under multiple modalities can be seen as a special case of the Multiple Kernel Learning problems where the kernels are computed on different feature sets. Bach et al. [39] suggested to solve the MKL problem with an iterative, wrapper like, sparse algorithm where in each iteration they solve a standard SVM dual problem and update the weights of the basic kernels. Instead of optimizing multiple times over the training set with a combination of kernel functions, we will define a novel kernel function combining all the representations into a single feature space. Our method is wrapper-free and is hence scalable for large data sets as well.

Late fusion approaches, see e.g. [56] and the references therein, combine the outputs of various kernel methods. Usually, they take an estimated certainty of each kernel method into account. In contrast to late fusion, our approach learns a kernel over various modalities instead of combining the outputs of different kernel methods.

3. THE SIMILARITY KERNEL

A natural idea to handle distances of pairs of observation is to use kernel methods. A kernel acts as an inner product between two observations in certain large dimensional space where Support Vector Machine, a form of a high dimensional linear classifier, can be used to separate the data points [44]. Under certain mathematical conditions, we have a freedom to define the kernel function by giving the formula for each pair of observations.

In this section, we show how the Fisher information matrix defines a natural distance over a possibly multimodal representation of complex instances. Our goal is to define a unified kernel function with the following properties:

1. A single kernel should include all modalities to avoid the computational complexity of the multiple kernel learning problem and in particular the need for wrapper methods.
2. The kernel should be based on an underlying model that captures the connection and dependencies between the modalities or the multiple representations.
3. Data points should posses a generative model so that the Fisher information matrix can be used to define a mathematically justified optimal kernel.

3.1 Random Field representation

As the first step, we represent our data as a Random Field by assuming that the data instances are generated by defining their distances from certain selected instances S. Practically, we will select the training instances or, in case of too many of them, a subset of the training set but we may in fact use an arbitrary sample S.

We will consider our data points as random variables forming a Markov Random Field described by an undirected graph. For a target instance x, we define a generative model by a simple graph that has edges between x and each elements of the sample S.

We define a generative model of x based on its similarity or distance $\mathrm{dist}(x, s)$ to elements of sample S. In this random field, the factor graph is a star that consists of the pairs of x connected to the elements $s \in S$. By the Hammersley–Clifford theorem [40], the joint distribution of the generative model for X is a Gibbs distribution. Next we derive this distribution via an appropriate potential function.

3.2 The potential function

Given a Markov Random Field defined by a graph, a wide variety of proper potential functions can be used to define a Gibbs distribution. The weak but necessary restrictions are that the potential function has to be positive real valued, additive over the maximal cliques of the graph, and more probable configurations (specific sets of parameters) have to have lower potential.

Our first and least complex graph is a bipartite graph connecting only the actual observations and the finite set of previously known observations. For simplicity first we will discuss the single modality case. In this graph the maximal cliques are the pairs of the actual observation and the elements of the sample set, therefore our potential function can have a really simple form,

$$U(X \mid S, \theta = \{\alpha_i\}) = \sum_{i=1}^{|S|} \alpha_i \mathrm{dist}(x, s_i), \tag{1}$$

where θ is the hyperparameter and $s_i \in S$ is the ith sample.

For K modalities with different distance functions between the instances, the potential function has the form

$$U(x \mid S, \theta = \{\alpha_{ik}\}) = \sum_{i=1}^{|S|} \sum_{k=1}^{K} \alpha_{ik} \mathrm{dist}_k(x, s_i), \tag{2}$$

where K is the number of different distance functions and $\theta = \{\alpha_{ik}\}$ is the set of the parameters. For simplicity, from now on we omit S and use θ to denote the hyperparameters.

Given the potential function over the maximal cliques, by the Hammersley–Clifford theorem [40], the joint distribution of the generative model for X is a Gibbs distribution

$$p(X \mid \theta) = \mathrm{e}^{-U(X|\theta)}/Z(\theta) \tag{3}$$

where

$$Z(\theta) = \int_{X \in \mathcal{X}} \mathrm{e}^{-U(X|\theta)} \mathrm{d}X \tag{4}$$

is the expected value of the energy function over our generative model, a normalization term called the partition func-

tion. If the model parameters are previously determined, Z is a constant.

3.3 The Fisher Kernel

According to Jaakkola and Haussler [27], generative models have a natural kernel function based on the Fisher information matrix F.

The main innovation of Jaakkola and Haussler [28] is to obtain the kernel function *directly from a generative probability model* and therefore obtain a kernel quite closely related to the underlying model. They consider a parametric class of probability models $P(X|\theta)$ where $\theta \in \Theta \subseteq \mathbb{R}^l$ for some positive integer l.

Provided that the dependence on θ is sufficiently smooth, the collection of models with parameters from Θ can then be viewed as a (statistical) manifold M_Θ. M_Θ can be turned into a Riemannian manifold[1] [29] by giving a scalar product at the tangent space of each point. $P(X|\theta) \in M_\Theta$ via a positive semidefinite matrix $F(\theta)$, which varies smoothly with the base point θ. Such positive semidefinite matrices are provided by the Fisher information matrix

$$F(\theta) := \mathbf{E}(\nabla_\theta \log P(X|\theta) \nabla_\theta \log P(X|\theta)^T),$$

where the gradient vector $\nabla_\theta \log P(X|\theta)$ is

$$\nabla_\theta \log P(X|\theta) = \left(\frac{\partial}{\partial \theta_1} \log P(X|\theta), \ldots, .. \frac{\partial}{\partial \theta_l} \log P(X|\theta) \right),$$

and the expectation is taken over $P(X|\theta)$. In particular, if $P(X|\theta)$ is a probability density function, then the ij-th entry of $F(\theta)$ is

$$f_{ij} = \int_X P(X|\theta)(\frac{\partial}{\partial \theta_i} \log P(X|\theta))(\frac{\partial}{\partial \theta_j} \log P(X|\theta))dX.$$

In many cases the kernel can actually be viewed as an inner product:

$$K(X,Y) = \phi_X^T \phi_Y,$$

where the feature vectors $\phi_X, \phi_Y \in \mathbb{R}^k$ are obtained via a fixed, problem specific map $X \mapsto \phi_X$ which describes the examples X in terms of a real vector of length k.

The vector $G_X = \nabla_\theta \log P(X|\theta)$ is called the *Fisher score* of the example X. Now the mapping $X \mapsto \phi_X$ of examples to feature vectors can be $X \mapsto F^{-\frac{1}{2}} G_X$ (we suppressed here the dependence on θ), the *Fisher vector*. Thus, to capture the generative process, the gradient space of the model space M_Θ is used to derive a meaningful feature vector. The corresponding kernel function

$$K(X,Y) := G_X^T F^{-1} G_Y$$

is called the *Fisher kernel*.

An intuitive interpretation is that G_X gives the direction where the parameter vector θ should be changed to fit best the data X [36].

[1] A Riemannian manifold M is a smooth real manifold, where for each point $p \in M$ there is an inner product defined on the tangent space of p. This inner product varies smoothly with p. One can define the length of a tangent vector via this inner product on the tangent space. This makes possible to define the length of a curve $\gamma(t)$ on M by integrating the length of the tangent vector $\dot{\gamma}(t)$. This in turn allows to define a metric on M. The distance between two points Q and Q' is just the length of the shortest curve on M from. Q to Q'.

3.4 Fisher Kernel over Markov Random Fields

In this section we prove that the Fisher Information matrix assuming Gibbs distribution with potential function (1) is the variance matrix of the distances $\text{dist}_k(x, s_i)$ for $s \in S$, and therefore the Fisher kernel is the linear kernel over the normalized distances.

First, let us calculate the Fisher score based on our general generative model,

$$\begin{aligned} G_X^i &= \nabla_{\theta_i} \log p(X|\theta) \\ &= -\frac{\partial(U(X|\theta)}{\partial \theta_i} + \frac{1}{Z(\theta)} \int_{X \in \mathcal{X}} \mathrm{e}^{U(X|\theta)} \frac{\partial(U(X|\theta)}{\partial \theta_i} \mathrm{d}X. \end{aligned} \quad (5)$$

As we set our model θ fixed, $Z(\theta)$ is a constant and our formula can be simplified as

$$G_X^i = \mathbf{E}_\theta[\frac{\partial(U(X|\theta)}{\partial \theta_i}] - \frac{\partial(U(X|\theta)}{\partial \theta_i}. \quad (6)$$

The first part of the formula can be calculated from the observation X while the expected value (the mean of the gradient of the potential function) is hard to compute. Worth to mention, if there exists a probability density function $f(X \mid \theta)$ such that

$$U(X \mid \theta) = -\log f(X \mid \theta) \quad (7)$$

then the expected term of (6) is zero trivially. For a potential function as in equation (1), the Fisher score of X has a simple form,

$$G_X^i = \mathbf{E}_\theta[\text{dist}(\mathbf{x}, \mathbf{s_i})] - \text{dist}(x, s_i). \quad (8)$$

Before we move on to the analysis of the dimensionality, let us examine the computational properties of the Fisher information matrix.

3.5 Approximation of the Fisher Kernel over Gibbs distribution

The computational complexity of the Fisher information matrix is $\mathcal{O}(N|\theta|^2)$ where N is the size of the training set. The linearization of the Fisher kernel through Cholesky decomposition is also an expensive procedure depending only on the size of the parameter set.

To reduce the complexity to $\mathcal{O}(N|\theta|)$ we can approximate the Fisher information matrix with the diagonal as suggested in [27, 36].

Focusing on the diagonal of the Fisher information matrix, we get

$$\begin{aligned} f_{i,i} &= \mathbf{E}_\theta[\nabla_{\theta_i} \log p(X|\theta)^T \nabla_{\theta_i} \log p(X|\theta)] \\ &= \mathbf{E}_\theta[(\mathbf{E}_\theta[\frac{\partial(U(X|\theta)}{\partial \theta_i}] - \frac{\partial(U(X|\theta)}{\partial \theta_i})^2] \\ &= \int_{X \in \mathcal{X}} p(X \mid \theta)(\mathbf{E}_\theta[\frac{\partial(U(X|\theta)}{\partial \theta_i}] - \frac{\partial(U(X|\theta)}{\partial \theta_i})^2 \mathrm{d}X. \end{aligned} \quad (9)$$

For the potential function of equation (1), the diagonal of the Fisher kernel is the standard deviation of the distances from the samples and therefore the Fisher vector of X has the following form

$$\mathcal{G}_X^i = F_{ii}^{-\frac{1}{2}} G_X^i = \frac{\mathbf{E}_\theta[\text{dist}(x, s_i)] - \text{dist}(x, s_i)}{\mathbf{E}_\theta^{\frac{1}{2}}[(\mathbf{E}_\theta[\text{dist}(x, s_i)] - \text{dist}(x, s_i))^2]} \quad (10)$$

The above formula can be directly computed from the distance matrix of the sample S and the training and testing instances X. The dimensionality of the Fisher vector (the normalized Fisher score) is equal to the size of the parameter set of our joint distribution. In our case it depends only on the size of the sample S and the number of modalities (K), $dim_{Fisher} = K \cdot |S|$.

4. EXPERIMENTS

We performed experiments on publicly available real-world datasets from various domains. Next, we briefly describe the datasets, the underlying domains followed by the experimental protocol, results and discussion.

In all our experiments, we approximate the mean and variance of $\text{dist}_k(x, s_i)$ from the training data to compute the kernel as defined by equation (10). Since kernel methods are feasible for regression [38, 44], we also use the methods for predicting numerical values.

We used LibSVM [12] for classification problems and the Weka implementation of SMOReg [54][38] for regression.

Table 1: EEG prediction

Method	AUC	Gain(%)
DTW k-NN k=1	0.7534	+0.0
DTW k-NN k=100	0.7847	+0.0%
SimKer: 64xDTW |S|=100	0.8275	+5.4%
SimKer: MultiDTW |S|=|T|	0.8506	+8.4%

Table 2: Visual concept detection over the Yahoo! MIR Flickr dataset

Method	Mod.	MiAP	Gain(%)
ColHOG (CH)	Vis.	0.3670	
SimKer: Flickr tags (Sim.JS)	Text.	0.3015	
SimKer: CH + JS (Sim.JSCH)	Multi	0.4257	+2.0%
L.Comb: CH + Sim.JS	Multi	0.4170	+0.0%
L.Comb: Sim.JSCH + CH + Sim.JS	Multi	0.4467	+7.1%
SLWF by Liu (2014) [33]	Multi	0.4367	

Table 3: Quality prediction over the C3 dataset

Method	Mod.	MAE	RMSE	Gain(%)
BM25 SVM (BM)	Text.	0.6144	0.7915	+0.0%
C3 features GBT (GBT)	Netw.	1.3528	1.4961	
Lin. Comb.: BM + GBT	Multi	0.7459	0.8839	
SimKer: BM25	Text.	0.6196	0.8095	
SimKer: C3	Netw.	0.6900	0.8278	
SimKer: BM25 + C3	Multi	0.5891	0.7753	+4.2%

4.1 Time series classification

We performed experiments on the publicly available EEG dataset [58] from UCI machine learning repository[2] and the time series datasets from the UCR time series archive.[3]

For the classification of time-series, the k nearest-neighbor (k-NN) method using dynamic time warping (DTW) as distance measure was reported to be competitive, if not superior, to many state-of-the-art time-series classifiers, such as neural networks, hidden Markov models or support vector machines, see e.g. [15, 24, 55] and the references therein. Furthermore, Chen et al. [14] gave theoretical guarantees for the performance of nearest neighbor-like classifiers for time series. Therefore, we use k-NN with DTW as baseline.

EEG (electroencephalogram) is usually recorded on multiple channels, therefore, multimodality naturally arises with

[2]http://archive.ics.uci.edu/ml/datasets/EEG+Database
[3]www.cs.ucr.edu/~eamonn/time_series_data

Table 4: Quality prediction over the C3 dataset

Method	Mod	AUC	Gain(%)
tf SVM linear	Text.	0.6531	
tf SVM poly. d=2	Text.	0.6498	
tf SVM poly. d=3	Text.	0.6530	
tf.idf SVM linear	Text.	0.6496	
tf.idf SVM poly. d=2	Text.	0.6428	
tf.idf SVM poly. d=3	Text.	0.6464	
BM25 SVM linear (Lin)	Text.	0.6923	
BM25 SVM poly. d=2	Text.	0.6826	
BM25 SVM poly. d=3	Text.	0.6714	
C3 features LibFM	Netw.	0.6695	
C3 features GBT	Netw.	0.6688	
L.Comb.: Lin + LibFM	Multi	0.7100	
L.Comb.: Lin + GBT	Multi	0.7133	+0.0%
SimKer: tf JS (Sim.JS)	Text.	0.6978	
SimKer: BM25 L2 (Sim.BM)	Text.	0.7141	
SimKer: C3	Netw.	0.6571	
SimKer: BM+JS+C3 (Sim.All)	Multi	0.7363	+3.2%

Table 5: Web Spam detection over ClueWeb dataset

Method	Mod.	AUC	Gain(%)
BM25 SVM	Text.	0.8450	
Content features	Cont.	0.7882	
L.Comb.: BM + Cont.	Multi	0.8517	+0.0%
SimKer: BM25	Text.	0.8546	
SimKer: BM25 + Cont.	Multi	0.8622	+1.2%

Table 6: Classification of gene expression data

Method	AUC	Gain(%)
Linear SVM	0.9338	
Cosine SVM	0.9496	+0.0%
SimKer: cosine distance	0.9588	+0.9%

such data. Classification of EEG signals is one of the most prominent application domains in the light of ongoing American and European large scale research projects dedicated to study the brain and its disorders, such as the BRAIN Initiative [4] and the European Human Brain Project[5]. EEG is one of the most well-established techniques to capture the activity of the brain, it is widely used in research and clinical practice, see e.g. [2, 20, 42]. Paralyzed patients may benefit from EEG-controlled devices, such as spelling tools [8] or web browsers [5]. Furthermore, there were attempts to predict upcoming emergency braking based on EEG signals [26] which could result in reducing the braking distance of vehicles. A common feature of the aforementioned applications is that they involve classification of EEG signals.

The UCI EEG collection [58] contains in total 11028 EEG signals recorded from 122 persons. The total (decompressed) size of the data is several gigabytes which is roughly three orders of magnitude larger than the datasets from the UCR repository. Out of the 122 persons, there are 77 alcoholic patients and 45 healthy individuals. While capturing EEG, both alcoholic patients and healthy individuals were exposed to three different stimuli: subjects were shown either one picture or two different pictures or the same picture twice.

[4]http://en.wikipedia.org/wiki/BRAIN_Initiative
[5]https://www.humanbrainproject.eu

The dataset contains recordings for all the tree types of stimuli for all the subjects. Each signal was recorded using 64 electrodes at 256 Hz for 1 second. Therefore, each EEG signal is a 64-dimensional time series of length 256 in this collection. Multimodality, a core aspect of the proposed technique, naturally arises with multidimensional time series: each channel may correspond to a modality.

As a noise filter, a simple preprocessing step, we reduced the length of the signals from 256 to 64 by binning with a window size of four, i.e., we averaged four consecutive values of the signal.

In order to simulate the clinically relevant scenario in which the classifier is applied to the EEG of new patients, we randomly assign each person to either training or test split of the data and *all* the signals of the same person were either assigned to the training set or to the test set. In total, randomly selected 50 % of the all persons were assigned to the training set, while the remaining persons were assigned to the test set.

We performed two experiments on EEG data. In the first experiment, we randomly selected 100 signals as sample set S and calculated the DTW distances between these reference signals and other train and test signals for each channel *separately*. This experiment simulates application scenarios in which classification time is essential: in order to classify a new time series, we only need to calculate its distance to relatively few reference signals and use these distances as features in our approach. This allows quick and accurate classification of new signals. As the third row of Table 1 shows, our approach outperforms the baseline in terms of AUC.

In the second experiment, we used multivariate DTW as distance of two EEG signals. For a detailed description of multivariate DTW we refer to [9]. In this experiment, the distances from all the training signals were used as features in our approach, SimKer. While the DTW-calculations in this scenario require non-negligible computational effort, as Table 1 shows, this results in further improvements in terms of classification accuracy as measured by AUC.

Additionally, we performed experiments on the datasets of UCR time series archive which is one of the most frequently used benchmark in the time series literature. Note that the datasets in this collection are rather small, a few megabytes each, therefore, training advanced models on the datasets from the UCR collection is inherently difficult. Consequently, the advantage of complex models to simpler ones may not be pronounced on the UCR time series datasets, and we do not expect to observe substantial differences between different models on the UCR time series. In our approach, SimKer, we used DTW as distance measure and considered the distances from each training time series.

The results on the datasets of the UCR archive show that our approach clearly outperformed the baseline on some of the datasets of the archive, while the overall difference between the performance of our approach and the baseline was not found to be statistically significant using paired t-test at significance level of 0.05. We note that while we performed experiments on the data from the UCR time series archive, we considered only *one* modality (the DTW-distance of a time series x from the train time series), because no other modality was available for this data. Therefore, we could not exploit one of the major advantages of the proposed method, i.e., its ability to fuse several modalities.

4.2 Gene Expression

Proteins play essential role in almost all biological processes at the cellular level. Genes are particular subsequences of DNA that code for proteins. While each cell of the organism has the same DNA, the activation levels of genes may vary in different tissues: informally speaking, the expression level of a gene means how frequently the corresponding DNA fragment is transcribed to RNA and translated to proteins. Various tissues are characterized by different gene expression patterns, furthermore, diseases such as cancer may be associated with characteristic gene expression patterns. Therefore, classification of gene expression data may contribute to diagnosis of various types of cancer such as colon cancer, lymphoma, lung cancer or subtypes of breast cancer [32]. In this paper, we used publicly available gene expression data of breast cancer tissues, colon cancer tissues, and lung cancer tissues, see [32] and the references thereoin for details. In these datasets, the expression levels of 7650, 6500 and 12,600 genes have been measured for 95, 62 and 203 patients in the breast cancer, colon cancer and lung cancer datasets respectively.

Similarly to [32], we performed experiments according to the 5-fold crossvalidation protocol. As baselines, we used SVMs, because SVMs were reported to perform excellently on these datasets.

Table 6 summarizes our results: we report AUC averaged over all the three datasets for SimKer and SVMs with linear and cosine kernel. The results show that SimKer outperforms both types of SVMs.

4.3 Web Spam detection over ClueWeb09

The first results on automatic Web quality classification focus on Web spam [11]. In this section, we show experiments over the Waterloo Spam Rankings [16] of the ClueWeb09 corpus.

Our baseline classification procedures are collected in [48] by analyzing the results of the Web Spam Challenges and the ECML/PKDD Discovery Challenge 2010. As our main conclusion, Web spam can be classified purely based on the terms used. Over different Web spam and quality corpora [22], the bag-of-words classifiers based on the top few 10,000 terms performed best and significantly improved the traditional Web spam features [11]. SVM based content classification was first used in [1]. In our earlier result, we use libSVM [12] with several kernels and apply late fusion as described in [48]. We improve over this later result by using the Fisher kernel next.

Our most important feature set is the bag of words representation of the text over the Web host. Let there be H hosts consisting of an average $\overline{\ell}$ terms. Given a term t of frequency f over a given host that contains ℓ terms, we used the BM25 term weightning scheme, where the weight of t in the host becomes

$$\log \frac{H-h+0.5}{h+0.5} \cdot \frac{f(k+1)}{f+k(1-b+b\cdot\frac{\ell}{\overline{\ell}})}. \qquad (11)$$

Low k means very quick saturation of the term frequency function while large b downweights content from very large Web hosts.

In addition, we use the public feature set [10] that includes the following values computed for the home page, page with the maximum pagerank and average over the entire host:

1. Number of words in the page, title;
2. Average word length, average word trigram likelihood;
3. Compression rate, entropy;
4. Fraction of anchor text, visible text;
5. Corpus and query precision and recall.

Here feature classes 1–4 can be normalized by using the average and standard deviation values over the two collections while class 4 is likely domain and language independent.

Corpus precision and recall are defined over the k most frequent words in the dataset, excluding stopwords. Corpus precision is the fraction of words in a page that appear in the set of popular terms while corpus recall is the fraction of popular terms that appear in the page. This class of features is language independent but rely on different lists of most frequent terms for the two data sets.

Results for spam detection in Table 5 show 1.2% improvement for the multimodal Similarity kernel over the linear combination of the predictions of the BM25 based SVM and the content feature based SVM.

4.4 Web credibility classification

Mining opinion from the Web and assessing its quality and credibility became a well-studied area [21]. Classifying various aspects of quality was introduced as part of the ECML/PKDD Discovery Challenge 2010 tasks [48] and among others, Microsoft created a reference data set [46].

Recent results on Web credibility assessment [34] use content quality and appearance features combined with social and general popularity and linkage. After feature selection, they use 10 features of content and 12 of popularity by standard machine learning methods of the scikit-learn toolkit.

In this section we show the performance of the Fisher kernel for the WebQuality 2015 Data Challenge by comparing prediction methods for the C3 data set. The data set was created in the Reconcile[6] project and contains 22325 Web page evaluations in five dimensions (credibility, presentation, knowledge, intentions, completeness) of 5704 pages given by 2499 people. The mTurk platform were used for collecting evaluations. Ratings are similar to the dataset built by Microsoft for assessing Web credibility [46], on a scale of four values 0-4, with 5 indicating no rating. Since multiple values may be assigned to the same aspect of a page, we simply average the human evaluations per page. We may also consider binary classification problems by assigning 1 for above 2.5 and 0 for below 2.5.

While we are aware of no other results over the C3 data set, we collect reference methods from Web credibility research results. Existing results fall in four categories: Bag of Words; language statistical, syntactic, semantic features; numeric indicators of quality such as social media activity; and assessor-page based collaborative filtering. User and page-based collaborative filtering is suggested in [35] in combination with search engine rankings. Social media and network based features appear already for Web spam [25, 11]. Content statistics as a concise summary that may replace the actual terms in the document were introduced first in the Web spam research [11]. The C3 data set includes content quality and appearance features described among others in [34].

[6] `http://reconcile.pjwstk.edu.pl/`

In order to perform text classification, we crawled the pages listed in the C3 data set. By using the bag of words representation of the Web page content, our goal is to combine all above methods with known and new kernel based text classifiers.

Our classifier ensemble consists of the following components:

- Gradient Boosted Trees and recommenders
- Standard text classifiers
- Similarity kernel based SVM using not only the text but also the C3 attributes.

In our experiments the Bag of words models contain the top $30k$ term frequencies after stemming. Besides BM25 (see Section 4.3), we experimented with two additional term frequency normalization schemes:

- Term frequency (tf): simply f, for all terms in the documents of H.
- Term frequency times inverse document frequency (tf.idf):

$$\log \frac{H - h + 0.5}{h + 0.5} \cdot f. \tag{12}$$

One of the main questions is how to select proper distance measures over the bag of words and C3 features. In addition to the linear metric over the C3 attributes and the L2 normalized bag of words representations (tf, tf.idf and BM25), we apply Jensen-Shannon divergence (JS) over the L1 normalized term distributions according to our previous results [48, 18].

Our most complex Fisher kernel (Sim.All) is based on three representations: Jensen-Shannon divergence over raw term distribution, Euclidean distance over L2 normalized BM25, Euclidean distance over scaled site features.

According the results in in Table 4 the use of Fisher kernel over the term frequency based Jensen-Shannon divergence (Sim.JS) already reaches accuracy of the best non Fisher method with a single modality (Lin, linear SVM over the BM25 features). Out of the non Fisher methods using only the C3 attributes the LibFM and the Gradient Boosted Tree (GBT) perform very similar. The ensemble of GBT and the linear SVM over BM25 performs 0.713 in AUC, achieving the accuracy of the best Fisher kernel with only one distance (Sim.BM).

The best method (Sim.All) outperforms the best non Fisher method (Linear combination of Lin and GBT) by 3.2% on average in AUC. The largest difference is 7.2% by classifying "knowledge". Similarity kernel performs similarly for regression (Table 3). We measured 4.2% improvement in MAE (Mean Absolute Error) and 2.1% in RMSE (Root Mean Squared Error) over the baseline method.

4.5 Visual concept detection: Yahoo! MIR Flickr dataset

Images are rarely being present alone, usually we can extract some content related textual or other non-visual information such as geo-location or date from their context. Besides non visual meta features we can think of any visual representation as an individual modality. Altogether we can easily define a set of very diverse distance functions over images.

Vast amount of tagged images is available over photo sharing services or even the public Web. In our experiments we

used the Yahoo! MIR Flickr dataset containing $15k$ images as the training set and $10k$ images as a test set [51]. The dataset was used for various challenges such as ImageCLEF 2012 Photo Annotation task [51] and in recent articles [33][7][50]. The aim is to detect the presence of 94 categories (a wide variety of concepts not limited to objects, e.g. daylight, indoor, underwater or citylife) in terms of their visual and textual features.

Among large number of Bag of Visual Words models (super vector [59], kernel codebook [52], locality-constrained [53] to name a few), Gaussian Mixture based Fisher encoding [37] appears best out of BoVW models by the evaluation work of [13][43], hence we choose the same method. The Fisher metric over Gaussian mixtures is a well-known method to measure the distance between two images based on their visual content [36, 13, 43]. The model extracts a large amount of local descriptors over various parts of the image. The Gaussian Mixture model describes the set of descriptors of the image assuming naive independence between the descriptors. In our experiments we calculated grayscale HOG (Histogram of Oriented Gradients [17]) and RGB color moments over a dense grid and multiple scales using four different macroblock sizes (24x24, 32x32, 48x48 and 64x64 pixels per block). Both descriptors were L2 normalized. To reduce the dimension of the descriptors we transformed the vectors by Principal Component Analysis (PCA). The procedure resulted approximately $140k$ descriptors per image. The final visual Fisher vectors with 512 Gaussians were calculated over the descriptors per image. Moreover we splitted the images into three parts according to Lazebnik et al. [41] increasing the number poolings per image.

Additionally, we computed Jensen-Shannon divergence of the images based on their Flickr tags. As a baseline, we combined linearly the predictions of the linear SVM over the Gaussian Mixture based visual Fisher kernel and the Similarity kernel of the Jensen-Shannon divergence over the Flickr tags. The multimodal Similarity kernel (JSCH) outperforms the baseline by 2% (see Table 2) in MiAP (Mean interpolated Average Precision, the metric at the task [51]). Our best method, surprassing the baseline by 7.1%, is a linear combination of the predictions using the visual Fisher kernel and the Similarity kernels, both textual and multimodal.

In comparison to recent results, our method outperforms the Selective Weighted Late Fusion (Liu et al. [33]) by 2.28%, the best result published to our knowledge over the MIR Flickr dataset.

5. CONCLUSIONS

From a generative model based on instance similarities, we derived a "similarity" kernel applicable for SVM classification and regression. The method is capable of defining a single unified kernel even in the case of rich data types, including multimodal or multiple time series data. The parameters of the kernel are directly computable from the data and hence we may avoid the high computational costs of multiple kernel learning and in particular the need for wrapper methods.

We evaluated our methods on a variety of publicly available real data sets, including multi-channel EEG, univariate time series, gene expression data, Web spam and credibility as well as image content with text annotation. Besides the presence of multiple modalities, complexity of classification and regression tasks in the aforementioned domains arise from various additional sources, such as high dimensionality (compared to the number of available instances), interdependence between attributes, presence of noise and uncertainty. Our experiments show that the proposed approach is able to successfully solve the underlying machine learning tasks, even under the presence of such additional domain and data complexity.

In particular, on all the aforementioned data sets, our method reaches and in many cases improves over the state-of-the-art. Hence we conclude generative models based on instance similarities with multiple modes is a generally applicable model for classification and regression tasks ranging over various domains, including but not limited to the ones presented in this paper.

6. ACKNOWLEDGEMENTS

This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. Research partially performed within the framework of the grant of the Hungarian Scientific Research Fund (grant No. OTKA 111710).

The publication was supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956), by the Momentum Grant of the Hungarian Academy of Sciences, by OTKA NK 105645, the KTIA_AIK_12-1-2013-0037 and the PIAC_13-1-2013-0197 projects. The projects are supported by Hungarian Government, managed by the National Development Agency, and financed by the Research and Technology Innovation Fund.

7. REFERENCES

[1] Jacob Abernethy, Olivier Chapelle, and Carlos Castillo. WITCH: A New Approach to Web Spam Detection. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2008.

[2] Jessica Askamp and Michel J.A.M. van Putten. Diagnostic decision-making after a first and recurrent seizure in adults. *Seizure*, 22(7):507 – 511, 2013.

[3] Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004.

[4] Asa Ben-Hur and William Stafford Noble. Kernel methods for predicting protein–protein interactions. *Bioinformatics*, 21(suppl 1):i38–i46, 2005.

[5] Michael Bensch, Ahmed A Karim, Jürgen Mellinger, Thilo Hinterberger, Michael Tangermann, Martin Bogdan, Wolfgang Rosenstiel, and Niels Birbaumer. Nessi: an EEG-controlled web browser for severely paralyzed patients. *Computational intelligence and neuroscience*, 2007.

[6] Mikhail Bilenko and Raymond J Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39–48. ACM, 2003.

[7] Alexander Binder, Wojciech Samek, Klaus-Robert Müller, and Motoaki Kawanabe. Enhanced

representation and multi-task learning for image annotation. *Computer Vision and Image Understanding*, 117(5):466–478, 2013.

[8] Niels Birbaumer, Nimr Ghanayim, Thilo Hinterberger, Iver Iversen, Boris Kotchoubey, Andrea Kübler, Juri Perelmouter, Edward Taub, and Herta Flor. A spelling device for the paralysed. *Nature*, 398(6725):297–298, 1999.

[9] Krisztian Antal Buza. *Fusion methods for time-series classification*. Peter Lang Verlag, 2011.

[10] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: web spam detection using the web topology. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 423–430, 2007.

[11] Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini, and Sebastiano Vigna. A reference collection for web spam. *SIGIR Forum*, 40(2):11–24, December 2006.

[12] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[13] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*, 2011.

[14] George H. Chen, Stanislav Nikolov, and Devavrat Shah. A latent source model for nonparametric time series classification. In *Advances in Neural Information Processing Systems 26*, pages 1088–1096. 2013.

[15] Yanping Chen, Bing Hu, Eamonn Keogh, and Gustavo EAPA Batista. Dtw-d: time series semi-supervised learning from a single example. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 383–391. ACM, 2013.

[16] G.V. Cormack, M.D. Smucker, and C.L.A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval*, 14(5):441–465, 2011.

[17] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, 2005.

[18] B. Daróczy, A. Benczúr, and R. Pethes. Sztaki at imageclef 2011. *Working Notes of CLEF 2011*, 2011.

[19] Bálint Daróczy, Dávid Siklósi, and András A Benczúr. Dms-sztaki@ imageclef 2012 photo annotation. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[20] J. Dauwels, F. Vialatte, T. Musha, and A. Cichocki. A comparative study of synchrony measures for the early diagnosis of alzheimer's disease based on eeg. *NeuroImage*, 49(1):668 – 693, 2010.

[21] K. Dave, S. Lawrence, and D.M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.

[22] Miklós Erdélyi, András Garzó, and András A. Benczúr. Web spam classification: a few features worth more. In *Joint WICOW/AIRWeb Workshop on Web Quality (WebQuality 2011) In conjunction with the 20th International World Wide Web Conference in Hyderabad, India.* ACM Press, 2011.

[23] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268, 2011.

[24] Steinn Gudmundsson, Thomas Philip Runarsson, and Sven Sigurdsson. Support vector machines and dynamic time warping for time series. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2772–2776. IEEE, 2008.

[25] Zoltán Gyöngyi and Hector Garcia-Molina. Spam: It's not just for inboxes anymore. *IEEE Computer Magazine*, 38(10):28–34, October 2005.

[26] Stefan Haufe, Matthias S Treder, Manfred F Gugler, Max Sagebaum, Gabriel Curio, and Benjamin Blankertz. Eeg potentials predict upcoming emergency brakings during simulated driving. *Journal of neural engineering*, 8(5):056001, 2011.

[27] Tommi S Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493, 1999.

[28] Tommi S Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493, 1999.

[29] Jürgen Jost. *Riemannian geometry and geometric analysis*. Springer, 2011.

[30] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

[31] Christina S Leslie, Eleazar Eskin, and William Stafford Noble. The spectrum kernel: A string kernel for svm protein classification. In *Pacific symposium on biocomputing*, volume 7, pages 566–575, 2002.

[32] Wei-Jiun Lin and James J Chen. Class-imbalanced classifiers for high-dimensional data. *Briefings in bioinformatics*, 14(1):13–26, 2013.

[33] Ningning Liu, Emmanuel Dellandrea, Bruno Tellez, and Liming Chen. A selective weighted late fusion for visual concept recognition. In *Fusion in Computer Vision*, pages 1–28. Springer, 2014.

[34] Alexandra Olteanu, Stanislav Peshterliev, Xin Liu, and Karl Aberer. Web credibility: Features exploration and credibility prediction. In *Advances in Information Retrieval*, pages 557–568. Springer, 2013.

[35] Thanasis G Papaioannou, Jean-Eudes Ranvier, Alexandra Olteanu, and Karl Aberer. A decentralized recommender system for effective web credibility assessment. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 704–713. ACM, 2012.

[36] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–8, 2007.

[37] Florent Perronnin, Jorge Sánchez, and Thomas

Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010.

[38] John C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, 1998.

[39] Alain Rakotomamonjy, Francis Bach, Stephane Canu, and Yves Grandvalet. simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[40] Brian D Ripley and Frank P Kelly. Markov point processes. *Journal of the London Mathematical Society*, 2(1):188–192, 1977.

[41] C. Schmid S. Lazebnik and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York, June 2006*, 2006.

[42] Malihe Sabeti, Serajeddin Katebi, and Reza Boostani. Entropy and complexity measures for eeg signal classification of schizophrenic and control participants. *Artificial Intelligence in Medicine*, 47(3):263 – 274, 2009.

[43] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[44] Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors. *Advances in kernel methods: support vector learning*. MIT Press, Cambridge, MA, USA, 1999.

[45] Bernhard Schölkopf, Koji Tsuda, and Jean-Philippe Vert. *Kernel methods in computational biology*. MIT press, 2004.

[46] Julia Schwarz and Meredith Morris. Augmenting web pages and search results to support credibility assessment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1245–1254. ACM, 2011.

[47] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[48] D. Siklósi, B. Daróczy, and A.A. Benczúr. Content-based trust and bias classification via biclustering. In *Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality*, pages 41–47. ACM, 2012.

[49] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006.

[50] B Thomee, M Huiskes, and M S. Lew. Special issue on visual concept detection in the mirflickr/imageclef benchmark. *Computer Vision and Image Understanding*, 117:451–452, 2013.

[51] B. Thomee and A. Popescu. Overview of the imageclef 2012 flickr photo annotation and retrieval task. *Working Notes of CLEF 2012, Rome, Italy*, 2012, 2012.

[52] Jan C van Gemert, Jan-Mark Geusebroek, Cor J Veenman, and Arnold WM Smeulders. Kernel codebooks for scene categorization. In *Computer Vision–ECCV 2008*, pages 696–709. Springer, 2008.

[53] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.

[54] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005.

[55] Xiaopeng Xi, Eamonn Keogh, Christian Shelton, Li Wei, and Chotirat Ann Ratanamahatana. Fast time series classification using numerosity reduction. In *Proceedings of the 23rd international conference on Machine learning*, pages 1033–1040. ACM, 2006.

[56] Guangnan Ye, Dong Liu, I-Hong Jhuo, and Shih-Fu Chang. Robust late fusion with rank minimization. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3021–3028. IEEE, 2012.

[57] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106, 2003.

[58] Xiao Lei Zhang, Henri Begleiter, Bernice Porjesz, Wenyu Wang, and Ann Litke. Event related potentials during object recognition tasks. *Brain Research Bulletin*, 38(6):531–538, 1995.

[59] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S. Huang. Image classification using super-vector coding of local image descriptors. In *Proceedings of the 11th European conference on Computer vision: Part V*, ECCV'10, pages 141–154, Berlin, Heidelberg, 2010. Springer-Verlag.

Exploiting Temporal Influence in Online Recommendation

Róbert Pálovics[1,2] András A. Benczúr[1,3] Levente Kocsis[1,4]
Tamás Kiss[1,3] Erzsébet Frigó[1,2]
[1]Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI)
[2]Technical University Budapest [3]Eötvös University Budapest [4]University of Szeged
{rpalovics, benczur, kocsis, kisstom, fbobee}@ilab.sztaki.hu

ABSTRACT

In this paper we give methods for time-aware music recommendation in a social media service with the potential of exploiting immediate temporal influences between users. We consider events when a user listens to an artist the first time and this event follows some friend listening to the same artist short time before. We train a blend of matrix factorization methods that model the relation of the influencer, the influenced and the artist, both the individual factor decompositions and their weight learned by variants of stochastic gradient descent (SGD). Special care is taken since events of influence form a subset of the positive implicit feedback data and hence we have to cope with two different definitions of the positive and negative implicit training data. In addition, in the time-aware setting we have to use online learning *and* evaluation methods. While SGD can easily be trained online, evaluation is cumbersome by traditional measures since we will have potentially different top recommendations at different times. Our experiments are carried over the two-year "scrobble" history of 70,000 Last.fm users and show a 5% increase in recommendation quality by predicting temporal influences.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information filtering; I.2.6 [**Artificial Intelligence**]: Learning

Keywords

temporal recommendation and evaluation; social influence; online matrix factorization; Last.fm; music recommendation

1. INTRODUCTION

Part of the appeal of Web 2.0 is to find other people who share similar interests. Last.fm organizes its social network around music recommendation: users may automatically share their listening habits and at the same time grow their friendship. Based on the profiles shared, users may see what artists friends really listen to the most.

In a recent paper [22], we proved the existence of the influence of friends on musical taste by carefully decoupling trends and homophily, the fact that friends are a priori more likely to have similar taste. In this paper we exploit the timely information gathered by the Last.fm service on users with public profile to exploit the potential influence between friends for recommendation. Last.fm's service is unique in that we may obtain a detailed timeline and catch immediate effects by comparing the history of friends in time and comparing to pairs of random users instead of friends.

As our main contribution, we give a matrix factorization mixture model for influence between friends that yield improved collaborative filtering methods. In the simplest setting, we may recommend a new artist a to a user u closely after a friend v listened to the same artist. When we turn to modeling the tensor data $< u, v, a >$ that may even involve the time elapsed since v listening to a, we face a very sparse problem. Hence instead of modeling the tensor, we flatten out along the variables and define three matrices in addition to single-variable effects similar to the ones defined by the centralization procedures of [4].

Since influence from friends has a very strong time dependence in that only the events of the last few hours or days may have an effect on the user behavior, in this paper we consider online learning with very strong time sensitivity. Compared to standard collaborative filtering methods, we process events only once and in the order they have appeared. As baseline we use online stochastic gradient descent (SGD) with high learning rate so that recent events have high contribution to the factor weights. The online factor model already incorporates not just popularity by using a high learning rate and involving an online updated item bias, but also part of friends' influence. Immediately after a user listens to an artist, the corresponding factor weights are relative strongly adjusted due to the high learning rate. If a friend has similar factor weights e.g. by homophily, the same artist will have high recommendation score after the learning step. The online factor model hence involves an implicit variant of an influence recommender by itself that we will further improve by a direct modeling of the influences.

To obtain the weighted combination of the baseline and the influence recommenders, we propose a new method for online learning user-dependent blending weights. If the derivatives of the individual models are available, a single SGD could optimize both the internal parameters and the blending weights. However as it turns out, the influence recom-

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.

http://dx.doi.org/10.1145/2645710.2645723.

mender requires a different set of implicit positive items and a procedure for generating a negative sample than the traditional online matrix factor model.

In our blend, we obtain a 5% of increase in quality, a strong result in view of the three-year Netflix Prize competition [6] to improve recommender quality by 10%. The fact that influences blend well with collaborative filtering and temporal effects prove that close events in the network bring in new information: friends' close events in the past can be exploited in a recommender system.

Finally, as part of our results, we introduce quality measures for time-aware recommender evaluation. As influence from friends has only a short, typically few hours effect, we retrain part of our models after each event and hence potentially give completely new top list of items for each event in the testing period. We highlight that discounted cumulative gain (DCG) computed individually for each event and averaged over time is an appropriate measure for real time recommender evaluation.

The rest of this paper is organized as follows. After describing our Last.fm data in Section 2, we explore for measurable signs of influence by friends in Section 3. Our main influence recommender is defined in Section 4, our online evaluation metric in Section 5, the online blending method in Section 6 and the baseline algorithms in Section 7. Finally we show our measurements for improved recommendation quality in Section 8.

1.1 Related results

The Netflix Prize competition [6] has recently generated increased interest in recommender algorithms in the research community and put recommender algorithms under a systematic thorough evaluation on standard data [5]. The final best results blended a very large number of methods whose reproduction is out of the scope of this paper.

Bonchi [7] summarizes the data mining aspects of research on social influence. He concludes that "another extremely important factor is the temporal dimension: nevertheless the role of time in viral marketing is still largely (and surprisingly) unexplored", an aspect that is key in our result. Notion of influence similar to ours is derived in [3, 8] for Flickr and Twitter cascades, respectively.

Closest to our results are the applications of network influence in collaborative filtering under the term of "social regularization" [18, 21, 25, 26]. These results add smoothing to make friends' model similar. We use social regularization as one baseline model in our experiments. In other results, only ratings and no social contacts are given [11], or in [13], both friendship and view information was present over Flickr, but the main goal was to measure the strength of the influence and no measurements were designed to separate influence from other effects.

Since our goal is to recommend different artists at different times, our evaluation must be based on the quality of the top list produced by the recommender. This so-called top-K recommender task is known to be hard [10]. A recent result on evaluating top-K recommenders is found in [9].

Music recommendation is considered in several results orthogonal to our methods that will likely combine well. Mood data set is created in [14]. Similarity search based on audio is given in [16]. Tag based music recommenders [12, 23] and many more, a few of them based on Last.fm tags, use annotation and fall into the class of content based methods as opposed to collaborative filtering considered in our paper [15, 19, 20].

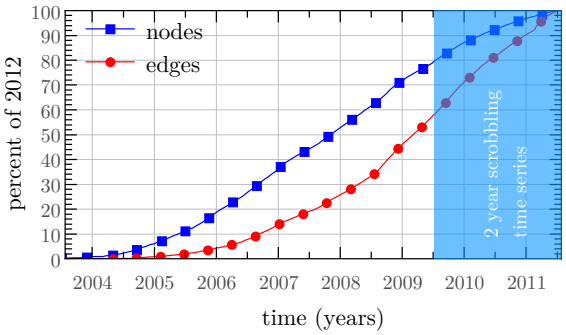


Figure 1: The number of the users and friendship edges in time as the fraction of the values at the time of the data set creation (2012).

2. THE LAST.FM DATA SET

Last.fm became a relevant online service in music based social networking. For registered users, it collects, "scrobbles"[1] what they have listened. Each user has its own statistics on listened music that is shown in her profile. Most user profiles are public, and each user of Last.fm may have friends inside the Last.fm social network. Therefore one relevant information for the users is that they see their own and their friends' listening statistics.

We investigate a data set that consists of the contacts and the implicit feedback timeline, the "scrobble history" of the users. Our goal is to exploit the influence of social contacts for recommendation. For privacy considerations, throughout our research, we selected an anonymous sample of users. Anonymity is provided by selecting random users while maintaining a connected friendship network. We set the following constraints for random selection:

- User location is stated in UK;
- Age between 14 and 50, inclusive;
- Profile displays scrobbles publicly (privacy constraint);
- Daily average activity between 5 and 500.
- At least 10 friends that meet the first four conditions.

The above selection criteria were set to select a representative part of Last.fm users and as much as possible avoid users who artificially generate inflated scrobble figures. In this anonymized data set of two years of artist scrobble timeline, edges of the social network are undirected and timestamped by creation date (Fig. 1). The number of users both in the time series and in the network is 71,000 with 285,241 edges; no edges are ever deleted from the network.

The time series contain 979,391,001 scrobbles from 2,073,395 artists and were collected between 01 January 2010 and 31 December 2011. The same user can scrobble an artist several times. The number of unique user-artist scrobbles is 57,274,158.

[1]The name "scrobbling" is a word by Last.fm, meaning the collection of information about user listening.

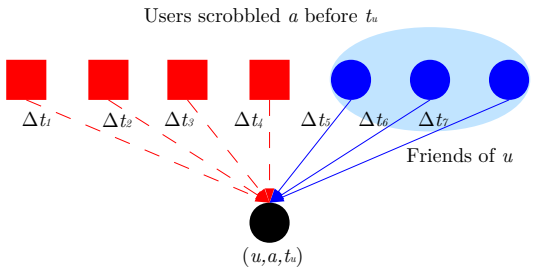


Figure 2: Potential influence on u by other users to scrobble (u, a, t_u).

3. NETWORK INFLUENCE

The key concept in this paper is a user v influencing another u to scrobble a. The sign of an influence is if u scrobbles artist a the first time at time t_u, after v last scrobbling the same artist at some time $t_v < t_u$ before. The time difference $\Delta t = t_u - t_v$ is the *delay*, as seen in Fig. 2. Our key assumption is that we observe such a subsequent first time scrobbling between non-friends only by coincidence while some of these events between friends are the result of certain interaction. Our goal is to prove that friends indeed influence each other and this effect can be exploited for recommendations.

Similar influence definitions are given in [3, 8, 13]. As detailed in [3], one main difference between these definitions is that in some papers t_v is defined as the first and not the last time when user v scrobbles a. The smaller the delay Δt between the scrobbles of v and u, the more certain we are that u is affected by the previous scrobble of v. The distribution of delay with respect to friends and non-friends will help us in determining the frequency and strength of influence over the Last.fm social network.

Out of the 57,274,158 first-time scrobbles of a certain artist a by some user, we find a friend who scrobbled a before 10,993,042 times (19%) in the whole time series and 4,203,109 times in the second year. Note that one user can be influenced by more friends, therefore the total number of influences is 24,204,977. If we only consider influences with delay less than one week, this number reduces to 4,625,141. Note that there is no influencing user for the very first scrobbler of a in the data set. For other scrobbles there is always an earlier scrobble by some other user, however, that user may not be a friend of u. Some of the observed subsequent scrobbles may result by pure coincidence, especially when a new album is released or the popularity of the artist increases for some other reason.

In order to quantify real influence within the set of subsequent first time scrobbles, our goal is to determine the probability that the subsequent scrobbles are result of influence. If we condition this probability for friends and by a limit t on the delay, we should obtain a monotonically decreasing function $\text{Infl}(t)$.

To formalize, let us consider the probability space of subsequent first time scrobbles among all users. Let I denote the event that an subsequent first time scrobble is the result of an influence. I^c is the opposite, no influence occurs. Coincidence or other, external reason such as the overall increase in popularity causes the subsequent first time scrobble in

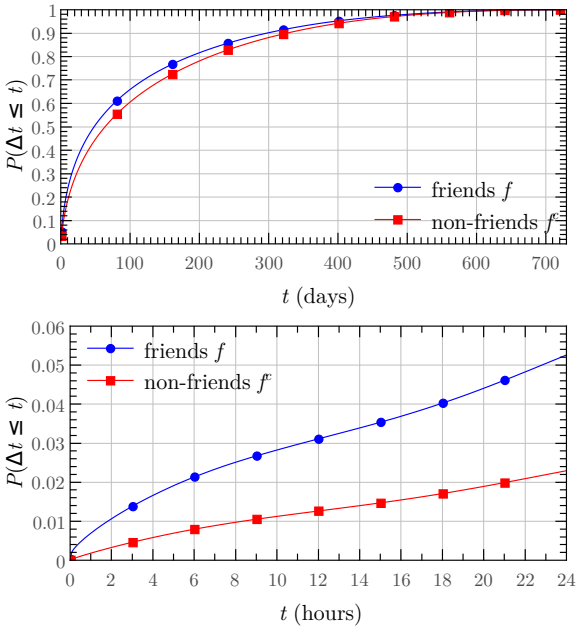


Figure 3: Fraction of subsequent first time scrobbles with delay $\Delta t \leq t$ as the function of t, in case of friends ($P(\Delta t \leq t \mid f)$) and non-friends ($P(\Delta t \leq t \mid f^c)$) over the entire timeline (**top**) and the first 24 hours (**bottom**).

the time series. Let f denote events between friends and f^c between non-friends. Finally let $\Delta t \leq t$ denote the set of events with delay at most t. With these notations,

$$\text{Infl}(t) = P(I \mid \Delta t \leq t, f) = \quad (1)$$

$$= \frac{P(I, \Delta t \leq t, f)}{P(\Delta t \leq t, f)} = \frac{P(\Delta t \leq t, I \mid f)P(f)}{P(\Delta t \leq t \mid f)P(f)} \quad (2)$$

$$= \frac{P(\Delta t \leq t, I \mid f)}{P(\Delta t \leq t \mid f)} = \frac{P(\Delta t \leq t \mid f) - P(\Delta t \leq t, I^c \mid f)}{P(\Delta t \leq t \mid f)}. \quad (3)$$

As non-friends f^c should not have any real influence on each other, we assume that

$$P(\Delta t \leq t, I^c \mid f) \approx P(\Delta t \leq t, I^c \mid f^c) = P(\Delta t \leq t \mid f^c). \quad (4)$$

Using this approximation, we can compute the probability of influences between friends as in (1) by expanding (3),

$$\text{Infl}(t) = P(I \mid \Delta t \leq t, f) \approx \frac{P(\Delta t \leq t \mid f) - P(\Delta t \leq t \mid f^c)}{P(\Delta t \leq t \mid f)}. \quad (5)$$

By the above equation, the influence probability can be approximated by observing the cumulative density curves in Fig. 3. The estimate of this function as in (5) is shown in Fig. 4. As expected, $\text{Infl}(t)$ is a monotonically decreasing function of t. However, the decrease is slow unlike in some recent influence models that propose exponential decay in time [13]. Therefore, we approximate the influence probability with a slowly decreasing logarithmic function instead of an exponential decay,

$$\text{Infl}(t) = 1 - c \log t, \quad (6)$$

where c is a constant.

4. INFLUENCE BASED RECOMMENDATION

Based on the measurements in the previous section, we model the observed influences and give a method to apply for

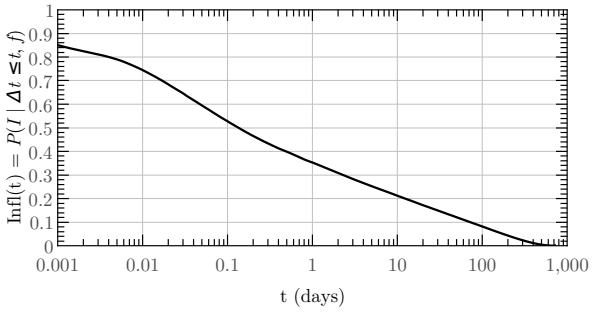


Figure 4: The influence probability approximated by equation (5), the ratio of increase among friends compared to non-friends very closely follows a logarithmic function of delay $\Delta t \leq t$.

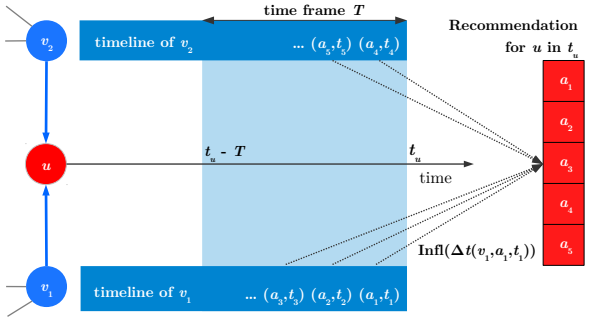


Figure 5: Scheme of the influence based recommender algorithm.

recommendation. Influence depends on time and no matter how relative slow, the influential power of a friend scrobbling an artist decays as time passes by. For this reason, the influence based recommender must learn online.

To formalize, let $v \xrightarrow{a;\Delta t \in \mathcal{T}} u$ denote the event that user u scrobbles artist a the first time in her time series, and Δt time after her friend v also scrobbled a. The time difference Δt is restricted to be in a time interval $\mathcal{T}$. As illustrated in Fig. 5, we would like to decompose the probability that $v \xrightarrow{a;\Delta t \in \mathcal{T}} u$ happens *and* the reason for this event is influence (I) between the users into a factor that only depends on Δt and another one that is independent of Δt. First we decompose the full event into a conditional probability as

$$P(I, v \xrightarrow{a;\Delta t \in \mathcal{T}} u) = P(I \mid v \xrightarrow{a;\Delta t \in \mathcal{T}} u) \cdot P(v \xrightarrow{a;\Delta t \in \mathcal{T}} u). \quad (7)$$

When a scrobble event happens at time exactly t after the scrobble of v, the interval becomes a point and hence we are looking for the derivative

$$\lim_{\tau \to 0} \left(P(I, v \xrightarrow{a;\Delta t \leq t+\tau} u) - P(I, v \xrightarrow{a;\Delta t \leq t} u) \right) / \tau. \quad (8)$$

We model the right hand side of (7), the "strength" of the influence between users u and v, independent of time as

$$f(v \xrightarrow{a} u) := P(v \xrightarrow{a;\Delta t \in \mathcal{T}} u)/|\mathcal{T}|, \quad (9)$$

hence by equation (7) we may divide the derivative (8) by $f(v \xrightarrow{a} u)$ to get

$$\lim_{\tau \to 0} \left(P(I \mid v \xrightarrow{a;\Delta t \leq t+\tau} u)(t+\tau) - P(I \mid v \xrightarrow{a;\Delta t \leq t} u)t \right) / \tau. \quad (10)$$

We model influence conditional probabilities by a global function for all users that depend only on the time Δt elapsed,

$$P(I \mid v \xrightarrow{a;\Delta t \leq t} u) \approx P(I \mid \Delta t \leq t, f). \quad (11)$$

By using our function in (5) and (6), equation (10) becomes

$$\lim_{\tau \to 0} \left((1 - c \log(t+\tau)) \cdot (t+\tau) - (1 - c \log t) \cdot t \right) / \tau \quad (12)$$

$$= 1 - c(1 + \log t). \quad (13)$$

Now we give our matrix factorization model for $f(v \xrightarrow{a} u)$. We decompose the model into seven terms that give a global model for one or more of the three variables u, v, a in $(v \xrightarrow{a} u)$. By replacing variables considered globally by $\bullet$ and noting that the last term with all three variables global is a constant, we get

$$\begin{aligned} f(v \xrightarrow{a} u) \sim\ & w_0 + w(\bullet \xrightarrow{\bullet} u) + w(v \xrightarrow{\bullet} \bullet) + w(\bullet \xrightarrow{a} \bullet) \\ & + w(\bullet \xrightarrow{a} u) + w(v \xrightarrow{a} \bullet) + w(v \xrightarrow{\bullet} u). \end{aligned} \quad (14)$$

We have four global effects, a constant, an influencer, an influenced, and an artist, and three bivariate terms that can be modeled by matrix factorization as

$$\begin{aligned} f(v \xrightarrow{a} u) \sim\ & \alpha_0 + \alpha_1 b_v + \alpha_2 b_u + \alpha_3 b_a \\ & + \vec{U}\vec{A} + \vec{A'}\vec{V} + \vec{U'}\vec{V'}. \end{aligned} \quad (15)$$

The three bias terms b_v, b_u and b_a correspond to the frequencies of user v influencing, user u being influenced and influences occurred with artist a. $\alpha_1, \ldots, \alpha_3$ are learned weights of the biases, and α_0 is the global learned bias. The six vectors correspond to six different latent vectors.

The final prediction score $\hat{r}$ is based on (8), by using (15) and (13) we can write it in a form

$$\begin{aligned} \hat{r}(u, a, t_u) = \sum_{v \in n(u)} & (\alpha_0 + \alpha_1 b_v + \alpha_2 b_u + \alpha_3 b_a \\ & + \vec{U}\vec{A} + \vec{A'}\vec{V} + \vec{U'}\vec{V'})(1 - c(1 + \log(t_u - t_v))), \end{aligned} \quad (16)$$

where we sum up for all neighbors of u and t_v is when v last scrobbled a before t_u. For training, we only update $f(v \xrightarrow{a} u)$ by the actual positive events and a generated sample of negative events. In our algorithm we use SGD with respect to MSE to train the latent factors and the weights $\alpha_0, \ldots, \alpha_3$. Notice that the weight of the factor models is included within the factors, since the entire formula (15) is trained by a single SGD procedure. As we learn online, the weight of the effects are also trained by SGD and not by the least squares optimization procedure proposed in [4].

In an efficient implementation, since the expression (13) quickly decays with t, we only need to retrieve the past scrobbles of all friends of u. This step is computationally inexpensive unless u has too many friends, when the recommendation is noisy anyway. To speed up computations, we only consider influence with delay T not more than a predefined time frame and hence we set $c = 1/(1 + \log T)$. With a sufficiently small parameter of the time frame in the range of a few days, our algorithm can hence be implemented even to provide recommendations based on real time updated models.

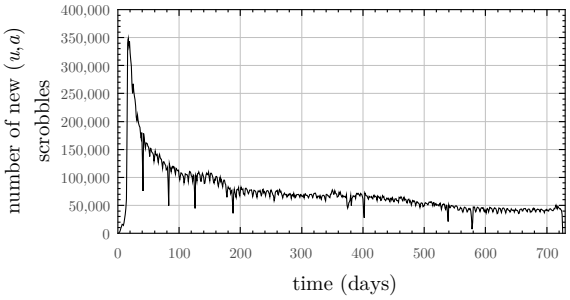


Figure 6: Number of new (u, a) scrobbles as the function of time.

5. ONLINE EVALUATION

Recommender systems in practice need to rank the best K items for the user. In this top-K recommendation task [10, 9] the goal is not to rate some of the individual items but to provide the best candidates. Despite the fact that only prediction for the top list matters in top-K evaluation, several authors propose models trained for RMSE with good top-K performance [17, 24] and hence we follow their approach.

In a time sensitive or online recommender that potentially retrains its model after each and every scrobble, we have to generate new top-K recommendation list for every single scrobble in the test period. The online top-K task is hence different from the standard recommender evaluation settings, since there is always a single item only in the ground truth and the goal is to aggregate the rank of these single items over the entire testing period. For our task, we need carefully selected quality metrics that we describe next.

Out of the two year scrobbling data, we use the full first year as training period. The second year becomes the testing period where we consider scrobbles one by one. We allow a recommender algorithm to use part or full of the data before the scrobble in question for training and require a ranked top list of artists as output. We evaluate the given single actual scrobble a in question against the recommended top list of length K. As seen in Fig. 6, by the second year, the number of first-time scrobbles stabilize around 50,000 a day after the artificial peak in the beginning caused by the lack of earlier data. For the reason of stability, we measure our recommender methods in Year 2 of the timeline.

One possible measure for the quality of a recommended top list of length K could be precision and recall [25, 26]. Note that we evaluate against a single scrobble. Both the number of relevant (1) and the number of retrieved (K) items are fixed. Precision is $1/K$ if we retrieve the single item scrobbled and 0 otherwise. Recall is 0 if we do not retrieve the single relevant item and 1 otherwise. The value of K that maximizes precision is the rank of the item scrobbled and hence "maximal precision" follows the function of 1/rank.

Recently, measures other than precision and recall are preferred for measuring the quality of top-K recommendation [2]. The most common measure is NDCG that is a normalized version of the discounted cumulative gain (DCG) with threshold K

$$\text{DCG@K}(a) = \begin{cases} 0 & \text{if } \text{rank}(a) > K; \\ \dfrac{1}{\log_2(\text{rank}(a)+1)} & \text{otherwise.} \end{cases} \quad (17)$$

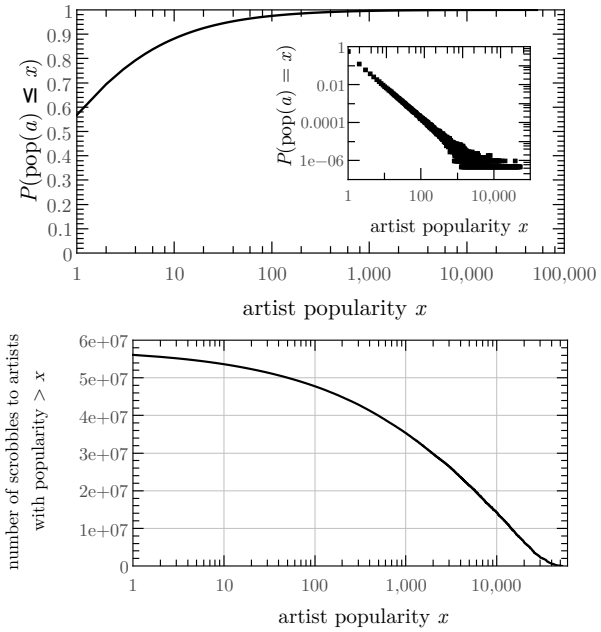


Figure 7: **Top:** Distribution of scrobble count to a given artist and the cumulative distribution. **Bottom:** Fraction of scrobbles for artists with popularity at least a given value x, as the function of x.

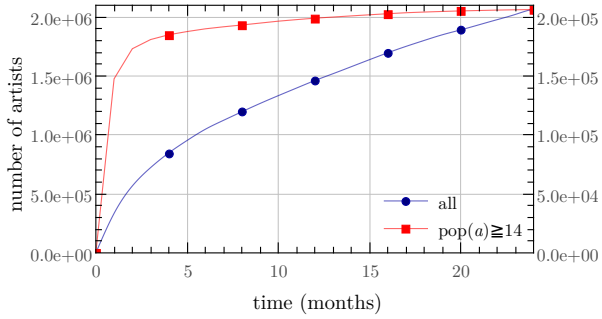


Figure 8: The number of different artists scrobbled before a given time in the two year period of the data set.

Since DCG is a slower decreasing function of the rank than what we observed for maximal precision, DCG is more advantageous since we have a large number of artists of potential interest to each user. Our choice is in accordance with the observations in [2] as well.

Note that in our unusual setting of DCG evaluation, there is a single relevant item and hence for example no normalization is needed as in case of the DCG measure. Also note that the DCG values will be small since the NDCG of a relative short sequence of actual scrobbles will roughly be equal to the sum of the individual DCG values. The DCG measured over 100 subsequent scrobbles of different artists cannot be more than the ideal DCG, which is $\sum_{i=1}^{100} 1/\log_2(i+1) = 20.64$ in this case (the ideal value is 6.58 for $K = 20$). Hence the DCG of an individual scrobble will on average be less than 0.21 for $K = 100$ and 0.33 for $K = 20$.

In our evaluation we discard infrequent artists from the data set both for efficiency considerations and due to the fact that our item based recommenders will have too little information on them. As seen in Fig. 7, top, the number of

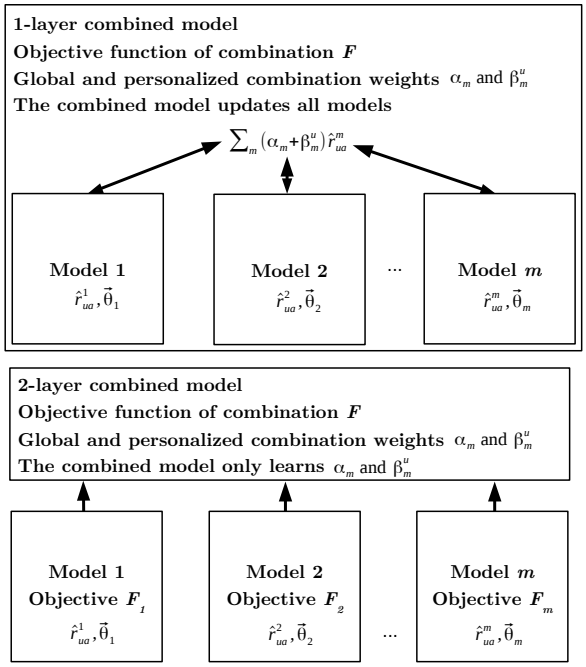


Figure 9: Scheme of the 1-layer(**top**) and 2-layer(**bottom**) online combination models.

artists with a given scrobble count follow a power-law distribution with near 60% of the artists appearing only once. While 90% of the artists gathered less than 20 scrobbles in two years, as seen in Fig. 7, bottom, they attribute to only less than 10% of the data set. In other words, by discarding a large number of artists, we only lose a small fraction of the scrobbles. For efficiency we only consider artists of frequency more than 14.

As time elapses, we observe near linear increase in the number of artists that appear in the data set in Fig. 8. This figure shows artists with at least 14 scrobbles separately. Their count grows slower but still we observe a large number of new artist that appear in time and exceed the minimum count of 14. Very fast growth for infrequent artists may be a result of noise and unidentified artists from e.g. YouTube videos and similar Web sources.

6. ONLINE BLENDING

We give two methods based on SGD that learn the online blending weight of recommender algorithms. Note that the algorithms may or may not themselves be based on SGD, i.e. the derivative of the individual models may or may not be available for the blending optimization procedure. Furthermore, we may blend methods with different definitions of the implicit feedback data sequence: the positive instances for the influence based recommender form a small subset of all the events and hence the influence recommender also needs different methods for generating negative training samples.

If the derivatives of the individual models are available for the top level optimizer, we may optimize in a single layer (top of Fig. 9) by minimizing

$$F(\hat{r}_{ua}) = F\left(\sum_m (\alpha_m + \beta_m^u)\hat{r}_{ua}^m\right), \tag{18}$$

where we sum over all models m, and F is the error measure, MSE in our case. Notice that we learn a user dependent blending weight vector β_m^u, hence for example the blending of a k and a k' factorization will in theory have at most as high F as a single $k+k'$ one, and in our experience performed only slightly better.

We may take the derivatives for both the constants α and β and the individual model parameters $\vec{\theta}_m$:

$$\frac{\partial F}{\partial \vec{\theta}_m} = \frac{\partial F}{\partial \hat{r}_{ua}} \frac{\partial \hat{r}_{ua}}{\partial \hat{r}_{ua}^m} \frac{\partial \hat{r}_{ua}^m}{\partial \vec{\theta}_m} = \frac{\partial F}{\partial \hat{r}_{ua}} (\alpha_m + \beta_m^u) \frac{\partial \hat{r}_{ua}^m}{\partial \vec{\theta}_m}; \tag{19}$$

$$\frac{\partial F}{\partial \alpha_m} = \frac{\partial F}{\partial \hat{r}_{ua}} \frac{\partial \hat{r}_{ua}}{\partial \alpha_m} = \frac{\partial F}{\partial \hat{r}_{ua}} \hat{r}_{ua}^m; \tag{20}$$

$$\frac{\partial F}{\partial \beta_m^u} = \frac{\partial F}{\partial \hat{r}_{ua}} \frac{\partial \hat{r}_{ua}}{\partial \beta_m^u} = \frac{\partial F}{\partial \hat{r}_{ua}} \hat{r}_{ua}^m. \tag{21}$$

If the derivatives are not available, the individual models are considered as black box for blending and we have to train in two layers (bottom of Fig. 9) and we may only use the last two derivatives (20) and (21).

If different models need different training samples, we cannot use the derivative (19) either. This is the case if we combine the baseline matrix factorization with the algorithm of Section 4. If the current positive event is not the result of an influence (i.e. not a first time scrobble or no friend scrobbling the same artist before), then we only update the baseline models. And if there is at least one possible influencer $v \xrightarrow{a} u$ for the current event (u, a), then we generate separate negative training instances for the baseline and the influence models. Notice that even a negative influence training data $v' \xrightarrow{a} u$ must satisfy that v' is a friend of u who scrobbled a and hence we usually have to choose from a restricted small set. Blending is meaningful only over this restricted set too, since for other events, the influence recommender has no t value in equation (16) to compute its prediction. Hence for blending, we have to use the same negative samples as for training the influence model.

7. MUSIC RECOMMENDATION BASELINE METHODS

We describe three baseline methods. The first one is based on dynamic popularity in Section 7.1. The second one in Section 7.2 is an online matrix factorization and the third one in Section 7.3 adds regularization over friendship as in [18].

All the methods discussed here are online algorithms, as opposed to the batch methods used in challenges such as Netflix. In some preliminary experiments the batch algorithms performed significantly worse in the online task compared to their online versions. We plan to compare the performance of batch and online versions of the algorithms in an online task more extensively in the future.

7.1 Dynamic popularity based recommendation

Given a predefined time frame T as in Section 4, at time t_u we recommend an artist based on the popularity in time not earlier than $t_u - T$ but before t_u. In our algorithm we update the counts and store artists sorted by the current popularity. In one time step, we may either add a new scrobble event or remove the earliest one, corresponding to a count increment

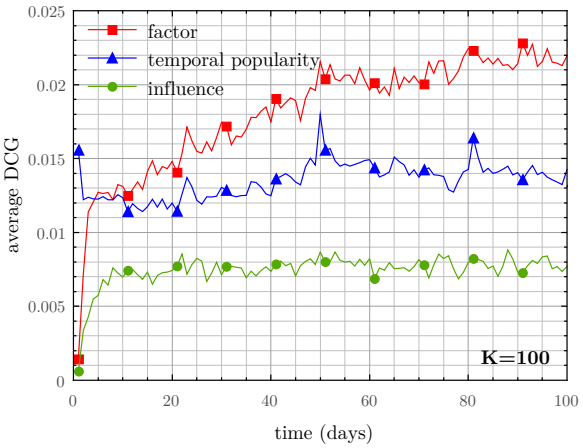


Figure 10: Online performance of the three different recommenders.

or decrement. For globally popular items, the sorted order can be maintained by a few changes in the order only.

7.2 Online matrix factorization

Stochastic gradient descent methods in batch setting may iterate several times over the training set until convergence. In an online setting [1], the model needs to be retrained after each new event and hence reiterations over the earlier parts of the data is ruled out. We may implement an online recommender algorithm by allowing a single iteration over the training data only, and this single iteration processes the events in the order of time. We used the first time scrobbles as positive training instances and generated negative training instances by selecting three random artists uniformly at the time when a user first scrobbled an artist.

Online recommenders seem more restricted than those that may iterate over the data set several times and one would expect inferior quality by the online methods. Online methods however have the advantage of giving much more emphasis on recent events. In some sense, the online methods may incorporate the notion of influence from Section 3: if friends have similar taste and hence similar factor weights, a friend scrobbling some artist a will in the near future strengthen the weight for this artist for all users who have similar taste.

7.3 Social regularization

Ma et al. [18] propose a method to implement constraints in a factor model based recommender algorithm for keeping the profile of friends similar. We implemented both the average-based and the individual-based regularization of [18] and found the latter superior, hence we use individual-based regularization in our experiments. Note that these algorithms have no knowledge of time and hence cannot incorporate our notion of subsequent first time scrobbles as in Section 3, even though they may work very well for other, non-first-time scrobbles that we do not consider in this paper.

8. EXPERIMENTS

In this section we describe the quality of our results for the second year testing period. Under various settings, we give daily average DCG@K defined by equation (17).

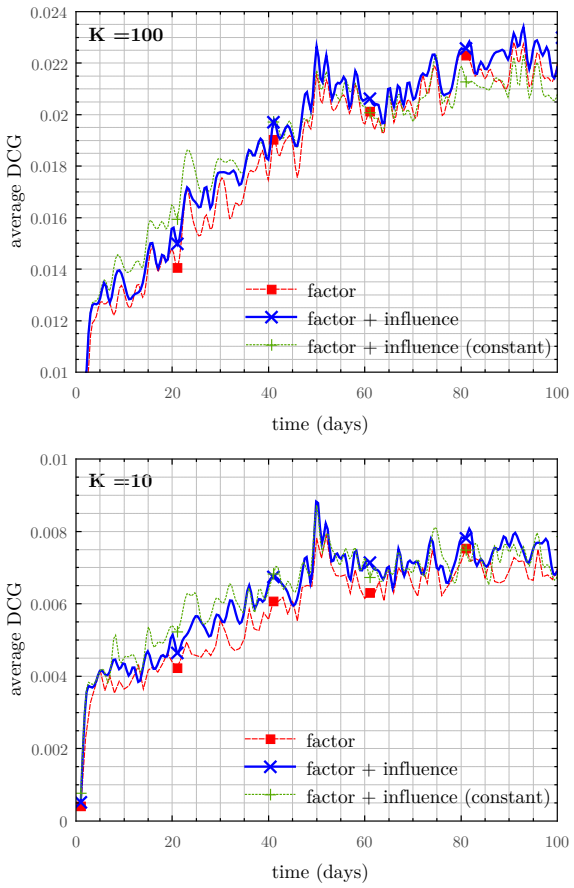


Figure 11: Combination of the influence and factor models.

Our experiments were carried out over the single core of an AMD based virtual server with 128GB RAM. On average, it took 28 minutes to process one day of scrobble history, update the online models and provide top-K recommendation corresponding to each user event.

Parameter K in equation (17) controls the length of the top list considered for evaluation. In other words, K can be interpreted as the size of the list presented to the user. Practically K must be small in order not to flood the user with information. We show results for $K = 10$ and 100. In Fig. 10, DCG@100 is shown for two baseline methods, matrix factorization and temporal popularity, as well as our influence model.

When combining variants of baseline and influence recommendation predictions, we observed that that social regularization did not improve matrix factorization and temporal popularity did not blend with online factorization. Indeed in Fig. 10 we may observe that peaks in temporal popularity performance immediately appear as peaks in matrix factorization performance, since online factorization learns temporal trends very well.

In Fig. 11, one can see that the online combination with influence recommendation improves over online matrix factorization both for DCG@10 and DCG@100. The average improvement is roughly 7% for DCG@10, and 3% for DCG@100. Over the same figure, we plot the performance of the constant term alone in equation (15). This simple recommender corresponds to adding up all the $(1 - c(1 + \log t))$ values for possible influencers without model building be-

yond learning the blending weight involved. At first this simple model blends best with the baseline, however, as the factor models get more training data, they become superior and the importance of the constant term α_0 in the model diminishes.

Conclusions

Based on a 70,000-entry sample of Last.fm users, we were able to exploit the effect of users influencing the taste of friends for improving the quality of music recommendation. Over static baseline recommenders, we achieved a 5% improvement in recommendation accuracy when presenting artists from friends' past scrobbles that the given user had never seen before.

Our system has very strong time-awareness: when we recommend, we look back in the near past and combine friends' scrobbles with the baseline methods. The influence from a friend at a given time is certain function of the observed influence in the past and the time elapsed since the friend scrobbled the given artist.

All of our methods learn online and provide top-K recommendation lists recomputed for every user query. Because of the inherent time dependence, we defined average DCG as our evaluation metric and gave a new online blending procedure that learns online user-dependent weights.

Acknowledgments

To the Last.fm team for preparing us this volume of the anonymized data set that cannot be efficiently fetched through the public Last.fm API.

The publication was supported by the KTIA_AIK_12-1-2013-0037 and the PIAC_13-1-2013-0205 projects. The projects are supported by Hungarian Government, managed by the National Development Agency, and financed by the Research and Technology Innovation Fund. Work conducted at the Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI) was supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956), by the Momentum Grant of the Hungarian Academy of Sciences, and by OTKA NK 105645. Work conducted at the Technical University Budapest has been developed in the framework of the project "Talent care and cultivation in the scientific workshops of BME" project. This project is supported by the grant TÁMOP - 4.2.2.B-10/1–2010-0009. Work conducted at University of Szeged was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013).

9. REFERENCES

[1] J. Abernethy, K. Canini, J. Langford, and A. Simma. Online collaborative filtering. *University of California at Berkeley, Tech. Rep*, 2007.

[2] A. Al-Maskari, M. Sanderson, and P. Clough. The relationship between ir effectiveness measures and user satisfaction. In *Proc. 30th SIGIR*, pages 773–774. ACM, 2007.

[3] E. Bakshy, J. M. H., W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proc. 4th WSDM*, pages 65–74. ACM, 2011.

[4] R. Bell and Y. Koren. Improved Neighborhood-based Collaborative Filtering. *KDD-Cup and Workshop*, 2007.

[5] R. Bell and Y. Koren. Lessons from the Netflix prize challenge. 2007.

[6] J. Bennett and S. Lanning. The netflix prize. In *KDD Cup and Workshop in conjunction with KDD*, 2007.

[7] F. Bonchi. Influence propagation in social networks: A data mining perspective. *IEEE Intelligent Informatics Bulletin*, 12(1):8–16, 2011.

[8] M. Cha, A. Mislove, B. Adams, and K. P. Gummadi. Characterizing social cascades in flickr. In *Proc. first workshop on Online social networks*, pages 13–18. ACM, 2008.

[9] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proc. 4th RecSys*, pages 39–46. ACM, 2010.

[10] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM TOIS*, 22(1):143–177, 2004.

[11] P. Domingos and M. Richardson. Mining the network value of customers. In *Proc. 7th SIGKDD*, pages 57–66. ACM, 2001.

[12] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. *Advances in neural information processing systems*, 20:385–392, 2007.

[13] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *Proc. 3rd WSDM*, pages 241–250. ACM, 2010.

[14] X. Hu, M. Bay, and J. Downie. Creating a simplified music mood classification ground-truth set. In *Proc. 8th ISMIR*, 2007.

[15] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. *PKDD*, pages 506–514, 2007.

[16] P. Knees, T. Pohle, M. Schedl, and G. Widmer. A music search engine built upon audio-based and web-based similarity measures. In *Proc. 30th SIGIR*, pages 447–454. ACM, 2007.

[17] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. 14th SIGKDD*, pages 426–434. ACM, 2008.

[18] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proc. 4th WSDM*, pages 287–296. ACM, 2011.

[19] B. Markines, C. Cattuto, F. Menczer, D. Benz, A. Hotho, and G. Stumme. Evaluating similarity measures for emergent semantics of social tagging. In *18th WWW*, pages 641–641. ACM, 2009.

[20] C. Marlow, M. Naaman, D. Boyd, and M. Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *Proc. 17th Hypertext and Hypermedia*, pages 31–40. ACM, 2006.

[21] J. Noel, S. Sanner, K.-N. Tran, P. Christen, L. Xie, E. V. Bonilla, E. Abbasnejad, and N. Della Penna. New objective functions for social collaborative filtering. In *Proc. 21st WWW*, pages 859–868. ACM, 2012.

[22] R. Pálovics and A. A. Benczúr. Temporal influence over the last. fm social network. In *Proc. ASONAM*, pages 486–493. ACM, 2013.

[23] K. Tso-Sutter, L. Marinho, and L. Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proc. Symp. on Applied Computing*, pages 1995–1999. ACM, 2008.

[24] M. Weimer, A. Karatzoglou, and A. Smola. Adaptive collaborative filtering. In *Proc. 2nd RecSys*, pages 275–282. ACM, 2008.

[25] X. Yang, H. Steck, Y. Guo, and Y. Liu. On top-k recommendation using social networks. In *Proc. 6th RecSys*, pages 67–74. ACM, 2012.

[26] Q. Yuan, L. Chen, and S. Zhao. Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In *Proc. 5th RecSys*, pages 245–252. ACM, 2011.

Text Classification Kernels for Quality Prediction over the C3 Data Set*

Bálint Daróczy Dávid Siklósi Róbert Pálovics András A. Benczúr
Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI)
{daroczyb, sdavid, rpalovics, benczur}@ilab.sztaki.hu

ABSTRACT

We compare machine learning methods to predict quality aspects of the C3 dataset collected as a part of the Reconcile project. We give methods for automatically assessing the credibility, presentation, knowledge, intention and completeness by extending the attributes in the C3 dataset by the page textual content. We use Gradient Boosted Trees and recommender methods over the evaluator, site, evaluation triplets and their metadata and combine with text classifiers. In our experiments best results can be reached by the theoretically justified normalized SVM kernel. The normalization can be derived by using the Fisher information matrix of the text content. As the main contribution, we describe the theory of the Fisher matrix and show that SVM may be particularly suitable for difficult text classification tasks.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information Storage and Retrieval; I.2 [**Computing Methodologies**]: Artificial Intelligence; I.7.5 [**Computing Methodologies**]: Document Capture—*Document analysis*

General Terms

Kernel Methods, Document Classification, Information Retrieval

Keywords

Web Quality, Credibility, Machine Learning, Fisher Information Matrix

*The publication was supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956), by the Momentum Grant of the Hungarian Academy of Sciences, by OTKA NK 105645, the KTIA_AIK_12-1-2013-0037 and the PIAC_13-1-2013-0197 projects. The projects are supported by Hungarian Government, managed by the National Development Agency, and financed by the Research and Technology Innovation Fund. Work conducted at the Technical University Budapest has been developed in the framework of the project "Talent care and cultivation in the scientific workshops of BME" project. This project is supported by the grant TÁMOP - 4.2.2.B-10/1–2010-0009.

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742126.

1. INTRODUCTION

Mining opinion from the Web and assessing its quality and credibility became a well-studied area [9]. Known results typically mine Web data on the micro level, analyzing individual comments and reviews. Recently, several attempts were made to manually label and automatically assess the credibility of Web content [19, 21]; among others, Microsoft created a reference data set [27]. Classifying various aspects of quality on the Web host level were, to our best knowledge, first introduced as part of the ECML/PKDD Discovery Challenge 2010 tasks [28].

Classification for quality aspects of Web pages or hosts turned out to be very hard. For example, the ECML/PKDD Discovery Challenge 2010 participants stayed with AUC values near 0.5 for classifying trust, bias and neutrality. Later we were able to slightly improve their results and our best performance has only slightly extended the AUC of 0.6 [28]. Since these attributes constitute key aspects of Web quality, our goal is to improve the classification techniques for these tasks.

In this paper we address the WebQuality 2015 Data Challenge by comparing prediction methods for the C3 data set. The data set was created in the Reconcile[1] project and contains 22325 evaluations (five dimensions, among them credibility) of 5704 pages given by 2499 people. The mTurk platform were used for collecting evaluations.

In our earlier findings on different Web spam and quality corpora [12], the bag-of-words classifiers based on the top few 10,000 terms performed best. We were able to significantly improve the traditional Web spam features [5] similar to the C3 attributes. In this paper our main goal is to evaluate known methods and combine them with new means of text classification particularly suited to the quality related tasks in question.

While we are aware of no other results over the C3 data set, we collect reference methods from Web credibility research results. Existing results fall in four categories: Bag of Words; language statistical, syntactic, semantic features; numeric indicators of quality such as social media activity; and assessor-page based collaborative filtering.

User and page-based collaborative filtering is suggested in [21] in combination with search engine rankings. We reuse our RecSys Challenge 2014 second place winner solution [20] to build a strong baseline method over the evaluator, site, evaluation triplets including the evaluator and site side information.

[1] `http://reconcile.pjwstk.edu.pl/`

Social media and network based features appear already for Web spam [5, 15]. In a collection designed similar to C3 [19], social and general popularity and linkage were introduced and used for credibility assessment. Some of these features, in particular social media popularity, are used by the RecSys Challenge 2014 [20] as well and hence we deploy the methods we used there.

Content statistics as a concise summary that may replace the actual terms in the document were introduced first in the Web spam research [5]. The C3 data set includes content quality and appearance features described among others in [19].

In order to perform text classification, we crawled the pages listed in the C3 data set. By using the bag of words representation of the Web page content, our goal is to combine all above methods with known and new kernel based text classifiers. Our classifier ensemble consists of the following components:

- Gradient Boosted Trees and recommender methods that reached us second place at the RecSys Challenge 2014 [20].
- Standard text classifiers, including our biclustering based method that performed best over the DC2010 data set [28].
- A new similarity kernel based SVM on the Fisher Information Matrix that may work over arbitrarily defined similarity measures over pairs of pages, using not only the text but also the C3 attributes.

Our best results reach the AUC of 0.74 for credibility, 0.81 for Presentation, 0.70 for Knowledge, 0.71 for Intentions and 0.70 for Completeness. We may hence say that all results reach the level of practical usability. Text classification is the main component: alone it reaches 0.73, 0.77, 0.69, 0.71 and 0.70, respectively, for the five quality dimensions.

The rest of this paper is organized as follows. First we begin with an extended motivation of our new text classification technique. After listing related results, in Section 2 we describe the data set used in this paper. In Section 3 we describe our classification framework. The results of the classification experiments over the C3 data set can be found in Section 4.

1.1 Motivation

In our new similarity kernel method, our goal is to move from terms as features to content similarity as features. On one hand, content similarity is more general and it can be defined by using the attributes other than term frequencies as well. Similarity based description is also scalable since we may select the number of reference documents as large as it remains computationally feasible.

In the paper our main goal is to define a theoretically justified kernel function over Web page similarities defined in a general way. Similarity may be based on the distribution of terms, the distance in the numeric C3 data attributes, or distances from clusters as we defined in [28].

By considering general notions of similarity as object descriptors for classification, we may combine different modalities in a theoretically justified way too. For example, kernel selection methods [23] performed well for image classification tasks [8] but kernel fusion methods from [23] have a very large number of parameters that are difficult to learn.

In our new method, we consider the similarity of a Web page in question to a set of selected reference pages as a generative model. By assuming independence of the reference pages, the generative model can be computed as we will describe in Section 3.2. Hence we may obtain theoretically justified coefficients to weight the importance of the different similarity functions and reference Web pages.

1.2 Related Results

Web users usually lack evidence about author expertise, trustworthiness and credibility [5]. The first results on automatic Web quality classification focus on Web spam. In the area of the so-called Adversarial Information Retrieval workshop series ran for five years [13] and evaluation campaigns, the Web Spam Challenges [4] were organized. The ECML/PKDD Discovery Challenge 2010 extended the scope by introducing labels for genre and in particular for three quality aspects [28].

Our baseline classification procedures are collected by analyzing the results of the Web Spam Challenges and the ECML/PKDD Discovery Challenge 2010. In our previous work [11, 28], we improved over the best results of the participants by using new text classification methods.

Recent results on Web credibility assessment [19] use content quality and appearance features combined with social and general popularity and linkage. After feature selection, they use 10 features of content and 12 of popularity by standard machine learning methods of the scikit-learn toolkit.

If sufficiently many evaluators assess the same Web page, one may consider evaluator and page-based collaborative filtering [21] for credibility assessment. In this setting, we face a dyadic prediction task where rich metadata is associated with both the evaluator and especially with the page. The Netflix Prize competition [3] put recommender algorithms through a systematic evaluation on standard data [2]. The final best results blended a very large number of methods whose reproduction is out of the scope of this experiment. Among the basic recommender methods, we use matrix factorization [17, 29]. In our experiments we use the factorization machine [24] as a very general toolkit for expressing relations within side information. Recently, the RecSys Challenge 2014 run a similar dyadic prediction task where Gradient Boosted Trees [30] performed very well [20].

2. THE DATA SET

The C3 data set consists of 22325 Web page evaluations in five dimensions (credibility, presentation, knowledge, intentions, completeness) of 5704 pages given by 2499 people. Ratings are similar to the dataset built by Microsoft for assessing Web credibility [27], on a scale of four values 0-4, with 5 indicating no rating. The distribution of the scores for the five evaluation dimensions can be seen in Fig. 1. Since multiple values may be assigned to the same aspect of a page, we simply average the human evaluations per page. We may also consider binary classification problems by assigning 1 for above 2.5 and 0 for below 2.5.

Since earlier results [21] suggest the use of collaborative filtering along the page and evaluator dimensions, we measure the distribution of the number of evaluations given by the same evaluator and for the same site in Fig. 2.

Distribution of the variance of the ratings is shown by heatmap of all pairs of ratings given for the same page and same dimension by pairs of different evaluators in Fig. 3.

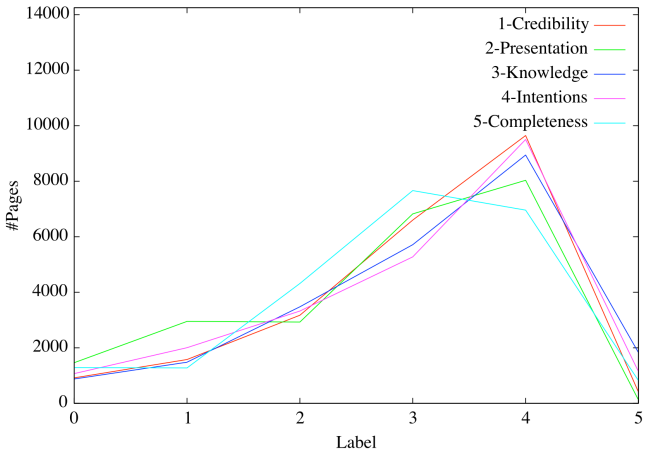


Figure 1: The distribution of the scores for the five evaluation dimensions.

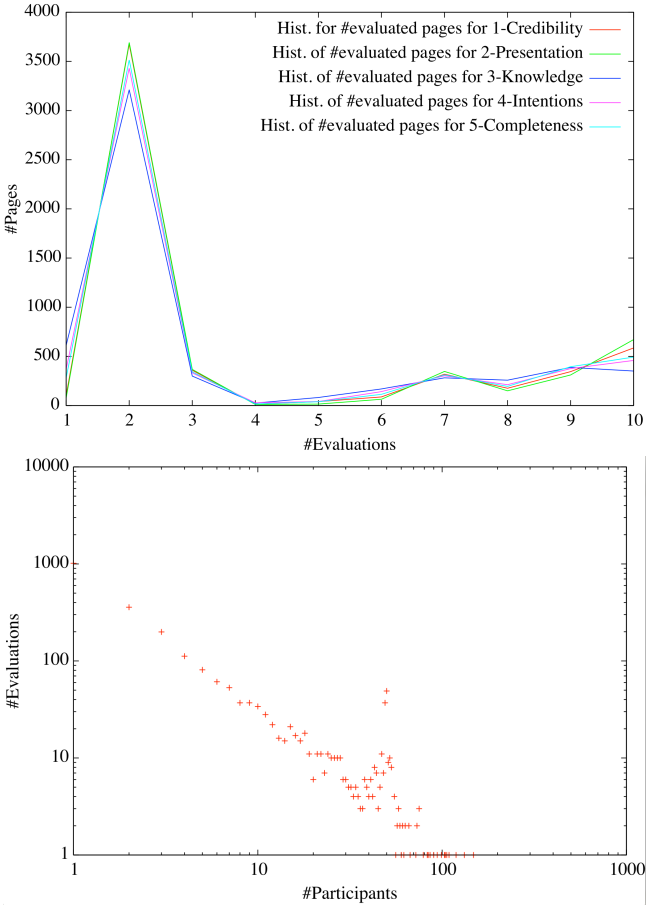


Figure 2: The distribution of the number of evaluations given by the same evaluator (top) and for the same site (bottom).

Note that 65% of the C3 URLs returned OK HTTP status but 7% of them could no longer be crawled. Redirects reached over 20% that we followed and substituted for the original page.

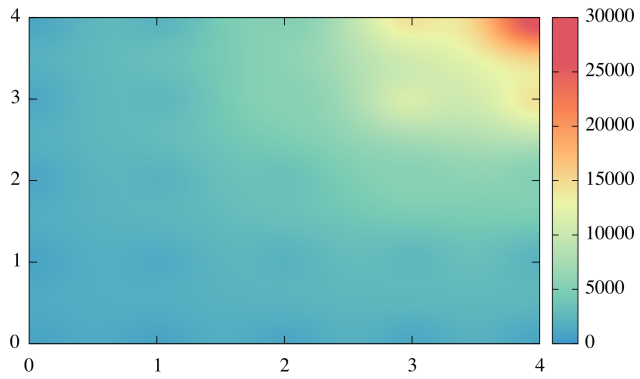


Figure 3: The number of pairs of ratings given by different assessors for the same aspect of the same page.

3. CLASSIFICATION FRAMEWORK

In this section we enumerate the methods we combine for assessing the five quality aspects. The C3 data set contains numeric attributes for the evaluator, the page, and the evaluation itself, which can be considered as triplets in a recommender system. The majority of the evaluators however rated only one Web page and hence we expect low performance of the recommender methods over this data set. Most important elements of our classifier ensemble will hence use the bag of words representation of the page content.

3.1 SVM over bag of words

The classification power of Support Vector Machine [7] over bag of words representations has been shown in [1, 5]. The models rely on term and inverse document frequency values (TF and IDF): aggregated as TF.IDF and BM25. The BM25 scheme turned out to perform best in our earlier results [11, 28], where we applied SVM with various linear and polynomial kernel functions and their combinations.

3.2 New method: Fisher Kernel over similarities

A natural idea to handle distances of pairs of observation is to use kernel methods. A kernel acts as an inner product between two observations in certain large dimensional space where Support Vector Machine, a form of a high dimensional linear classifier, can be used to separate the data points [26]. Under certain mathematical conditions, we have a freedom to define the kernel function by giving the formula for each pair of observations.

In order to combine the textual and C3 data attributes for kernel based classification and regression, we use a linear kernel support vector machine over distances from a selected set of reference pages as described in [8].

Given a sample R of the Web pages, we define a generative model where testing pages are characterized based on their similarity to samples in R. By Jaakkola and Haussler [16], generative models have a natural kernel function based on the Fisher information matrix F:

$$K_{Fisher}(X, Y) = G_X^T F^{-1} G_Y, \quad (1)$$

where G_X and G_Y are the gradient vectors (Fisher score) derived from the underlying generative model. The Fisher kernel can be translated into a linear kernel function using

Cholesky decomposition of the Fisher information matrix. We will refer the normalized Fisher score as Fisher vector: $F_X = G_x F^{-\frac{1}{2}}$. In our experiments we approximate the Fisher information matrix with the diagonal as suggested in [16].

Next we sketch the steps of deriving that the Fisher matrix based distance is simply the Euclidean distance over the $K \cdot |R|$ dimensional vector of the similarity to pages in R with K representations.

In the generative model of pages based on the similarity to pages in the sample R, our factor graph is a star that consists of the pairs of x connected to the elements $r \in R$. We think of our graph as a Markov Random Field over the samples. By the Hammersley–Clifford theorem [25] our joint distribution has a form of

$$p(x \mid \Omega) = \frac{\exp(-U(x \mid \Omega))}{Z}, \tag{2}$$

where Z is a normalizing constant and Ω is the set of parameters of our joint distribution. We define our energy function as

$$U(x \mid \Omega = \{\alpha\}) = \sum_{r \in R} \sum_{k=1}^{K} \alpha_{rk} \mathrm{dist}_k(x, x_r), \tag{3}$$

where K is the number of different distance functions and $\Omega = \{\alpha_{rk}\}$ is the set of the parameters.

It can be shown that the Fisher information matrix is simply the normalized variance matrix of the joint distribution $\mathrm{dist}_k(x, x_r)$ for $r \in R$, i.e. the Fisher kernel is the linear kernel over the normalized distances. In the Fisher kernel $\alpha_r k$ cancel out in the derivatives. The mean and the variance of $\mathrm{dist}_k(x, x_r)$ can be approximated by the training data.

The dimensionality of the Fisher vector (the normalized Fisher score) equals with the size of the parameter set of our joint distribution, in our case it depends only on the size of the reference set and the number of representations, $K \cdot |R|$.

Since kernel methods are feasible for regression [22, 26], we also use the methods of this subsection for predicting the numeric evaluation scores.

3.3 Biclustering

We overview the method that performed best for assessing the quality aspects of the DC2010 data [28]. We use Dhillon's information theoretic co-clustering algorithm [10] to cluster pages and terms simultaneously. Important to note that unlike in the original method [10] that uses Kullback-Leibler divergence, we use Jensen-Shannon, the symmetric version in the biclustering algorithm that makes very large difference in classification quality.

In [28] we describe pages by distances from page clusters. To exploit the Fisher kernel we can think of this page clusters as additional samples with a specific distance function. This results sparsity in our previously defined energy function

$$U(x \mid \Omega = \{\alpha, \beta\}) = U(x \mid \Omega = \{\alpha\}) + \sum_{C_i \in C} \beta_i \mathrm{dist}(x, C_i), \tag{4}$$

where C_i corresponds to the ith cluster, therefore the clusters behave as a secondary sample set to R on a cost of expanded dimension.

3.4 Gradient Boosted Trees and Matrix factorization

We apply Gradient Boosting Trees [30] and matrix factorization on the user and C3 data features. We used two different matrix factorization techniques. The first one is a traditional matrix factorization method [17], while the second one is a simplified version of Steffen Rendle's LibFM algorithm [24]. Both techniques use stochastic gradient descent to optimize for mean-square error on the training set. LibFM is particularly designed to use the side information of the evaluators and the pages.

3.5 Evaluation metrics

First, we consider binary classification problems by simply averaging the human evaluations per page and assign them 1 for above 2.5 and 0 for below 2.5. The standard evaluation metrics since the Web Spam Challenges [4] is the area under the ROC curve (AUC) [14]. The use of Precision, Recall and F are discouraged by experiences of the Web spam challenges.

Unlike spam classification, the translation of quality assessments into binary values is not so obvious. We also test regression methods evaluated by Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

4. RESULTS

In this section we measure the accuracy of various methods and their combinations. The detailed results are in Table 1, in four groups. The first group gives the baseline methods. Below, we apply the similarity kernel separate for the corresponding attributes. In the third group we combine multiple similarity functions by the similarity kernel. Finally, in the last group, we average after standardizing the predictions. In Table 2 part of the methods are tested for regression.

4.1 C3 data attributes

For user and item features we experiment with GraphLab Create[2] [18] implementation of Gradient Boosted Tree and matrix factorization techniques. In case of the gradient boosted tree algorithm (GBT) we set the maximum depth of the trees 4, and enabled maximum 18 iterations. To determine the advantage of additional side information over the original matrix factorization technique (MF) we use factorization machine (LibFM) for user and item feature included collaborative filtering prediction. As seen from the tables, matrix factorization (MF) fails due to the too low number of ratings by user and by document but LibFM can already take advantage of the website metadata with performance similar to GBT.

4.2 Linear kernel SVM

Our Bag of words models use the top $30k$ stemmed terms. For TF, TF.IDF and BM25, we show results for linear kernel SVM as it outperforms the RBF and polynomial kernels. We use LibSVM [6] for classification the Weka implementation of SMOReg [22] for regression.

4.3 Fisher kernel methods

The similarity kernel described in Section 3.2 gives the best results both for classification and for regression. For

[2] `http://graphlab.com/products/create/`

Method	Credibility	Presentation	Knowledge	Intentions	Completeness	Avg
Gradient Boosted Tree (**GBT**)	0.6492	0.6558	0.6179	0.6368	0.7845	0.6688
Factorization Machine (**LibFM**)	0.6563	0.6744	0.6452	0.6481	0.7234	0.6695
Marix Factorization (**MF**)	0.5687	0.5613	0.5966	0.5700	0.5854	0.5764
TF linear kernel	0.6484	0.6962	0.6239	0.6767	0.6205	0.6531
TF.IDF linear kernel	0.6571	0.7020	0.5935	0.6824	0.6128	0.6496
BM25 linear kernel (**Lin**)	0.7236	0.7480	0.6278	0.6987	0.6633	0.6923
Bicluster linear kernel	0.6402	0.7467	0.5796	0.6482	0.6382	0.6506
Bicluster Sim kernel	0.6744	0.7718	0.6379	0.6830	0.6560	0.6846
C3 attributes Sim kernel	0.6267	0.7706	0.6327	0.6408	0.6149	0.6571
TF J–S Sim kernel	0.6902	0.7404	0.6758	0.7047	0.6778	0.6978
TF L_2 Sim kernel	0.6335	0.6882	0.6200	0.6585	0.6300	0.6460
TF.IDF J–S Sim kernel	0.7006	0.7546	0.6552	0.7073	0.6791	0.6994
TF.IDF L_2 Sim kernel	0.6461	0.7152	0.6013	0.6902	0.6353	0.6576
BM25 J–S Sim kernel	0.6956	0.7473	0.6351	0.6529	0.6222	0.6706
BM25 L_2 Sim kernel	0.7268	0.7715	0.6741	0.7081	0.6898	0.7141
BM25 L_2 & J–S Sim kernel (**BM25**)	0.7313	0.7761	0.6926	0.7141	0.7003	0.7229
BM25 & C3 Sim kernel	0.7449	0.8029	0.7009	0.7148	0.6993	0.7326
BM25 & Bicluster & C3 (**All**) Sim kernel	0.7457	0.8086	0.7063	0.7158	0.7052	0.7363
Lin + GBT	0.7296	0.8056	0.6589	0.6783	0.6939	0.7133
Lin + LibFM	0.7400	0.7769	0.6622	0.6733	0.6975	0.7100
All Sim kernel + Lin + GBT	0.7549	0.8179	0.6916	0.7098	0.7123	0.7373

Table 1: Detailed performance over the C3 labels in terms of AUC

Method		Credibility	Presentation	Knowledge	Intentions	Completeness	Avg
Gradient Boosted Tree (GBT)	MAE	1.5146	1.3067	1.2250	1.2737	1.4438	1.3528
	RMSE	1.6483	1.4510	1.3658	1.4132	1.6021	1.4961
Factorization Machine (LibFM)	MAE	1.5313	1.3213	1.2303	1.2632	1.4984	1.3689
	RMSE	1.6725	1.4745	1.3744	1.4073	1.6759	1.5209
Matrix Factorization (MF)	MAE	1.7450	1.4093	1.3676	1.2905	1.5794	1.4784
	RMSE	1.9174	1.5912	1.5540	1.4636	1.7583	1.6569
BM25 linear kernel (Lin)	MAE	0.5562	0.7230	0.6052	0.5979	0.5896	0.6144
	RMSE	0.7085	0.9072	0.7784	0.7910	0.7724	0.7915
BM25 L_2 Sim kernel	MAE	0.5678	0.7083	0.6228	0.5946	0.6045	0.6196
	RMSE	0.7321	0.9307	0.8038	0.7878	0.7930	0.8095
Bicluster Sim kernel	MAE	0.5340	0.6868	0.6039	0.5883	0.5813	0.5989
	RMSE	0.6958	0.8906	0.7861	0.7778	0.7624	0.7825
BM25 & Bicluster & C3 All Sim kernel	MAE	0.5403	0.6324	0.5946	0.5952	0.5829	0.5891
	RMSE	0.7106	0.8357	0.7763	0.7879	0.7661	0.7753

Table 2: Detailed performance over the C3 labels in terms of RMSE and MAE

distance, we use L_2 for the C3 attributes as well as TF, TF.IDF and BM25. For the last three, we also use the Jensen–Shannon divergence (J–S) as we suggested in [28]. While the similarity kernel over the bicluster performs weak for classification, it is the most accurate single method for regression.

In the similarity kernel, we may combine multiple distance measures by Equation (3). The All Sim method fuses four representations: J–S and L_2 over BM25 and L_2 for C3 and the bicluster representation. By the linearity of the Fisher kernel, we may use LibSVM [6] for classification and SMOReg [22] for regression.

4.4 Classifier ensembles

Without using the similarity kernel, the best method is the average of the linear kernel over BM25 (Lin) and GBT. The performance is similar to the BM25 L_2 similarity kernel. As a remarkable feature of the similarity kernel, we may combine multiple distance functions in a single kernel. The best method (All Sim) outperforms the best combination not using the similarity kernel (Lin + GBT) by 3.2%. The difference is 7.2% for classifying "knowledge". The same method performs bests for regression too.

The similarity kernel method can also resist noise and learn from small training sets. If we add 10% noise in the training set, the combination of all similarity kernels deteriorates only to an average AUC of 0.7241 from 0.7363 (1.7%). In contrast, the best BM25 SVM result 0.6923 degrades to

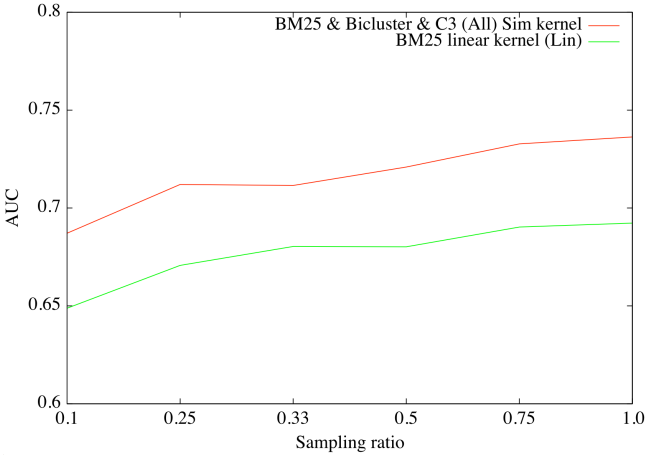


Figure 4: AUC as the function of the size of the training set, given as percent of the full3040, for the baseline BM25 with linear kernel and All with similarity kernel.

0.6657 (3.85%), both with variance 0.004 for ten independent samples. The robustness of the similarity kernel for small training sets is similar to BM25 with linear kernel, as seen in Fig. 4.

5. CONCLUSIONS

Over the C3 data sets, we gave a large variety of methods to predict quality aspects of Web pages, including collaborative filtering and methods that use evaluator and page metadata as well as the content of the page. We achieved best performance by our theoretically justified kernel method over the content of the page and C3 attributes. Our results are promising in that our AUC is stable over 0.7 for all aspects with "presentation" surpassing 0.8. The support vector regression methods also perform with error less than one on the range of 0–4.

6. REFERENCES

[1] J. Abernethy, O. Chapelle, and C. Castillo. WITCH: A New Approach to Web Spam Detection. In *Proc. 4th AIRWeb*, 2008.

[2] R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[3] J. Bennett and S. Lanning. The netflix prize. In *KDD Cup and Workshop in conj. KDD 2007*, 2007.

[4] C. Castillo, K. Chellapilla, and L. Denoyer. Web spam challenge 2008. In *Proc. 4th AIRWeb*, 2008.

[5] C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, and S. Vigna. A reference collection for web spam. *SIGIR Forum*, 40(2):11–24, December 2006.

[6] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20, 1995.

[8] B. Z. Daróczy, D. Siklósi, and A. Benczúr. SZTAKI @ ImageCLEF 2012 Photo Annotation. In *Working Notes of the ImageCLEF 2012 Workshop*, 2012.

[9] K. Dave, S. Lawrence, and D. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proc. 12th WWW*, pages 519–528. ACM, 2003.

[10] I. Dhillon, S. Mallela, and D. Modha. Information-theoretic co-clustering. *Proc. 9th SIGKDD*, pages 89–98, 2003.

[11] M. Erdélyi, A. A. Benczúr, B. Daróczy, A. Garzó, T. Kiss, and D. Siklósi. The classification power of web features. *Internet Mathematics*, 10(3-4):421–457, 2014.

[12] M. Erdélyi, A. Garzó, and A. A. Benczúr. Web spam classification: a few features worth more. In *WebQuality 2011*. ACM Press, 2011.

[13] D. Fetterly and Z. Gyöngyi. Proc. 5th AIRWeb. 2009.

[14] J. Fogarty, R. S. Baker, and S. E. Hudson. Case studies in the use of roc curve analysis for sensor-based estimates in human computer interaction. In *Proc. Graphics Interface*, pp. 129–136, 2005.

[15] Z. Gyöngyi and H. Garcia-Molina. Spam: It's not just for inboxes anymore. *IEEE Computer Magazine*, 38(10):28–34, October 2005.

[16] T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. *Advances in NIPS*, pp. 487–493, 1999.

[17] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[18] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein. Distributed graphlab: a framework for machine learning and data mining in the cloud. *Proc VLDB*, 5(8):716–727, 2012.

[19] A. Olteanu, S. Peshterliev, X. Liu, and K. Aberer. Web credibility: Features exploration and credibility prediction. In *Advances in Information Retrieval*, pages 557–568. Springer, 2013.

[20] R. Pálovics, F. Ayala-Gómez, B. Csikota, B. Daróczy, L. Kocsis, D. Spadacene, and A. A. Benczúr. Recsys challenge 2014: an ensemble of binary classifiers and matrix factorization. In *Proc. Recommender Systems Challenge*, page 13. ACM, 2014.

[21] T. G. Papaioannou, J.-E. Ranvier, A. Olteanu, and K. Aberer. A decentralized recommender system for effective web credibility assessment. In *Proc. 21st CIKM*, pp. 704–713. ACM, 2012.

[22] J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, 1998.

[23] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. simplemkl. *JMLR*, 9:2491–2521, 2008.

[24] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proc. 34th SIGIR*, pp. 635–644. ACM, 2011.

[25] B. D. Ripley and F. P. Kelly. Markov point processes. *Journal of the London Mathematical Society*, 2(1):188–192, 1977.

[26] B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in kernel methods: support vector learning*. MIT Press, Cambridge, MA, USA, 1999.

[27] J. Schwarz and M. Morris. Augmenting web pages and search results to support credibility assessment. In *Proc. SIGCHI*, pp. 1245–1254. ACM, 2011.

[28] D. Siklósi, B. Daróczy, and A. Benczúr. Content-based trust and bias classification via biclustering. In *Proc. Webquality*, pp. 41–47. ACM, 2012.

[29] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Investigation of various matrix factorization methods for large recommender systems. In *Proc. 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, pages 1–8. ACM, 2008.

[30] Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun. A general boosting method and its application to learning ranking functions for web search. In *Advances in NIPS*, pp. 1697–1704, 2008.

Temporally Evolving Models for Dynamic Networks*

Róbert Pálovics Frederick Ayala-Gómez András A. Benczúr

1 Introduction

The research of complex networks and large graphs generated a wide variety of stochastic graph models that try to capture the properties of these complex systems [2, 4, 5, 8, 7]. Most of the well-known models can describe a static graph extracted from a real-world dataset. They are capable of generating an ensemble of graphs, in which all graph instances are similar in terms of specific statistics to the original one. For example, models that capture the power-law degree distribution of real-world networks such as the Albert-Barabási one are dynamic but do not attempt to model the actual temporal evolution of large graphs. Our goal is to give temporal stochastic graph model for the temporal dynamics of these complex systems.

Our models address the link prediction problem introduced by Liben-Nowell and Kleinberg, in a *temporal* setting. More specifically, we try to predict accurately each new link in the graph at the time when it is created in the network. This experimental setting is similar to our method introduced for recommender systems [9]. In Section 3 we explain this setup in case of dynamic graphs. For baseline algorithm, we apply online matrix factorization [6, 10, 11] on temporal network data (see Section 4).

Various node centrality measures capture the "importance" of a node by using the structural properties of the graph [3]. While these metrics are widely investigated, few is known about the evolution of graph centrality in temporal graphs. In our work, we investigate the applicability of node centrality metrics in temporal graphs by examining their temporal behavior and computational complexity. We also use these metrics as side features in our matrix factorization models.

2 Datasets

We perform our experiments on a variety of Twitter, Last.FM and the Koblenz Collection data sets. Twitter is a highly temporal social system with dynamically evolving communities, a mass community with an ever changing graph structure. Our goal is to investigate the dynamics of this community that can be best described as an evolving complex network. The first issue on Twitter is to define a graph that well describes the system. One can define several networks in Twitter. We focus on follower networks, retweet cascades, root retweet networks, @mention graphs and @reply graphs. Our Twitter datasets contain tweets around events of global relevance, including Euromaidan, Maidan, Olympic, Occupy, Yosoy, 15o, 20n, MH17 and Ayotzinapa. Last.fm is an online service in music based social networking. It collects "scrobbles"—a word by Last.fm meaning that when you listen to a song, the name of the song is added to your music profile. We selected a representative, well-connected, yet anonymous random sample of users. These users had their location in UK with age between 14 and 50, inclusive. Only public scrobbles with a daily average activity between 5 and 500 and at least 10 friends that meet the first four conditions. Other temporal network datasets are available in the Koblenz Network Collection for Twitter mentions, arXiv author network, Digg, Flickr and YouTube graphs.

*The publication was supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956).

3 Experimental setting and evaluation metrics

In the dynamic link prediction task, we have to rank the best K links for the given node at the given time instance. Our dataset contains records $< u, v, t >$ of links between users u and v that appear at time t. Our goal is to recommend new links for user u at time t with the constraint that there is only a single link that appears at the given time t. This means that we have to maximize the rank of the given link in the actual predicted list of links. A time sensitive or online link prediction system should retrain its model after each and every training record $< u, v, t >$. We have to generate new top-K recommendation list for *every* single record. The online top-K task is hence different from the standard recommender evaluation settings, since there is always a single neighbor only in the ground truth and the goal is to aggregate the rank of these single neighbors over the entire testing period. For our task, we need carefully selected quality metrics that we describe next. We use our full dataset both for training and testing. We iterate on the records one by one in temporal order. For a given record $< u, v, t >$, we allow the recommender algorithm to use full of the data *before* t in question for training and require a ranked top list of possible neighbors as output. We evaluate the given single actual neighbor v in question against the recommended top list of length K.

For measuring the accuracy of predicting a new link, we face the difficulty that only a single correct answer exists at the given time and the next edge arrives to be tested against an updated model. We propose DCG [9], a modified version of NDCG, the prefered model for batch top-K recommendation [1]. DCG is a slowly decreasing function of the rank and hence measures how close the actual new link appears in the top list.

4 Dynamic adjacency matrix factorization

Batch modeling algorithms may iterate several times over the graph until convergence. In our temporal setting, the model needs to be retrained after each new event and hence reiterations over the earlier parts of the data is ruled out.

In this section, we give an online factorization method for the graph adjacency matrix. Matrix factorization yields a low-rank approximation of the adjacency matrix with entries for non-edges filled with values that we consider an indication for the edge to appear. Links for a given node are predicted by taking the largest values in the corresponding row or column. In our algorithm, we allow a single iteration over the training data only, and this single iteration processes the events in the order of time. We use each record in the dataset as a positive training instance and generate negative training instances by selecting random items for each positive record. We use the regularized matrix factorization method of [12], and use the k-factor model for prediction.

Temporal modeling methods seem more restricted than those that may iterate over the data set several times and one would expect inferior quality by the online methods. Online methods however have the advantage of giving much more emphasis on recent events that we empirically verify in our research.

4.1 Centrality measures as side information

Matrix factorization algorithm may use so-called side information associated with the rows and columns of the matrix. In our experiment we use centrality measures as side information associated with the nodes. We may use directed centrality with different values for rows and columns of the same node. We compare the following metrics in the temporal setting of dynamic networks based on [3]: Harmonic Centrality, PageRank, HITS and SALSA.

5 Conclusion and Further Work

In this paper, we analyze the dynamic network data as a stream of nodes and edges. To predict link formation, the regularized matrix factorization model is proposed. Different centrality measures are used

with online computation over the graph stream to identify the evolution of centrality. As the main lesson learned, we show how recent results in recommender systems can be deployed for the analysis of complex networks.

References

[1] A. Al-Maskari, M. Sanderson, and P. Clough. The relationship between ir effectiveness measures and user satisfaction. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 773–774. ACM, 2007.

[2] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[3] P. Boldi and S. Vigna. Axioms for centrality. *arXiv preprint arXiv:1308.2140*, 2013.

[4] B. Bollobás, O. Riordan, J. Spencer, G. Tusnády, et al. The degree sequence of a scale-free random graph process. *Random Structures & Algorithms*, 18(3):279–290, 2001.

[5] V. Csiszár, P. Hussami, J. Komlós, T. F. Móri, L. Rejtő, and G. Tusnády. When the degree sequence is a sufficient statistic. *Acta Mathematica Hungarica*, 134(1-2):45–53, 2012.

[6] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[7] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 57–65. IEEE, 2000.

[8] M. Kurucz and A. Benczúr. Geographically Organized Small Communities and the Hardness of Clustering Social Networks. *Annals of Information Systems*, pages 177–199, 2010.

[9] R. Pálovics, A. A. Benczúr, L. Kocsis, T. Kiss, and E. Frigó. Exploiting temporal influence in online recommendation. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 273–280. ACM, 2014.

[10] S. Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[11] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[12] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Investigation of various matrix factorization methods for large recommender systems. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, pages 1–8. ACM, 2008.

Temporal Twitter prediction by content and network

Bálint Daróczy[1] Róbert Pálovics[1,2] Vilmos Wieszner[3]
Richárd Farkas[3] András A. Benczúr[1]
[1]Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI)
[2]Technical University Budapest
[3]University of Szeged, Institute of Informatics
{daroczyb,rpalovics,benczur}@ilab.sztaki.hu, {wieszner,rfarkas}@inf.u-szeged.hu

ABSTRACT

In recent years Twitter became *the* social network for information sharing and spreading. By retweeting, users spreading information and build cascades of information pathways. In this paper we investigate the possibility of predicting the future popularity of emerging retweet cascades immediately after the message appears. We introduce a supervised machine learning approach which employs a rich feature set utilizing the textual content of the messages along with the retweet networks of the users. We also propose a temporal evaluation framework focusing on user level predictions in time.

Keywords

Twitter, Retweet prediction, Temporal classification, Language features

1. INTRODUCTION

Twitter, a mixture of a social network and a news media [16], has recently became the largest medium where users may spread information along their social contacts.

In this paper we investigate the temporal influence of messages sent over Twitter. Cha et al. [7] define influence as "...the power of capacity of causing an effect in indirect intangible ways...". In their key observation, the influence of a user is best characterized by the size of the audience who retweets rather than the size of the follower network.

Our goal is to predict the timely success of the information spread, on the individual message level. We analyze how certain messages may reach out to a large number of Twitter users. In contrast to a similar investigation for analyzing the influence of users [3], we investigate each tweet by taking both the author user and the textual content of the message into account.

We characterize the users both by the statistical properties of their follower network and their past retweet counts. The textual content is described by the terms of the normalized text and by several orthographic features along with deeper (psycho)linguistic ones that try to capture the modality of the message in question.

In our experiments we use the data set of [1] that consists of the messages and the corresponding user network of the Occupy movement.

The main contributions of this work is that we carried out an intensive feature engineering both at network and content analysis – instead of focusing on only one of them – and the added value of the two worlds was empirically evaluated. In our results we consider user and network features as defined in [8] and our previous work [19] as baseline and concentrate on the power of content analysis.

1.1 Related results

Social influence in Web based networks is investigated in several results: Bakshy et al. [4] model social contagion in the Second Life virtual world. Ghosh and Lerman [11] compares network measures for predicting the number of votes for Digg posts, who even give an empirical comparison of information contagion on Digg vs. Twitter [17]. In [12, 13], long discussion based cascades built from comments are investigated in four social networks, Slashdot (technology news), Barrapunto (Spanish Slashdot), Meneame (Spanish Digg) and Wikipedia. They propose models for cascade growth and estimate model parameters but give no size predictions.

A number of related studies have largely descriptive focus, unlike our quantitative prediction goals. In [7] high correlation is observed between indegree, retweet and mention influence, while outdegree (the number of tweets sent by the user) is found to be heavily spammed. [16] reports similar findings on the relation among follower, mention and retweet influence. Several more results describe the specific means of information spread on Facebook [5, 2, 6].

Similar to our results, Cheng et al. [8] predict retweet count based on network features. Unlike in our result where we predict immediately after the tweet is published, they consider prediction after the first few retweets. The network features used in their work are similar to the ones in the present paper and in our earlier work [19]. We consider these results as baseline in this paper.

From the content analysis point of view, there has been several studies focusing exclusively on the analysis of the tweet messages' textual content to solve the re-tweet count prediction problem. Besides the terms of the message, Naveed et al. [18] introduced the features of direct message, mention, hashtag, URL, exclamation mark, question mark, positive and negative sentiment, positive and negative emoticons and valence, arousal, dominance lexicon features. Wang et al. [22] proposed deeper linguistic features like verb tense, named entities, discourse relations and sentence similarity.

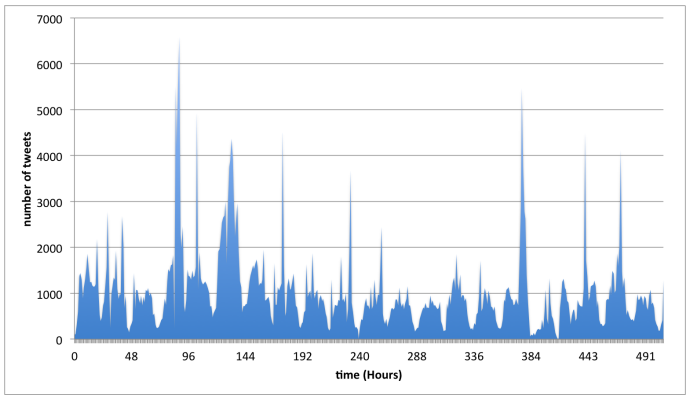


Figure 1: Temporal density of tweeting activity.

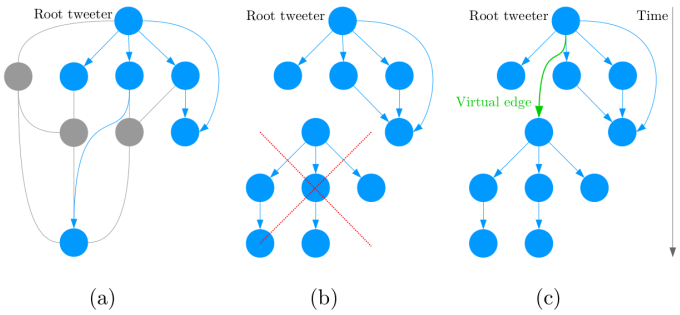


Figure 2: Creation of retweet cascades: Figure (a) shows the computation of the cascade edges. In Figures (b) and (c) we show the possible solutions in case of missing cascade edges.

Table 1: Size of the tweet time series.

Number of users	371,401
Number of tweets	1,947,234
Number of retweets	1,272,443

Table 2: Size of the follower network.

Number of users	330,677
Number of edges	16,585,837
Average in/out degree	37

Gupta et al. [14] addressed the task of scoring tweets according to their credibility. Credibility is a highly related phenomena to social influence. Moreover, this work is related to our ones as it also combines author, network and content features. The feature set to describe the content of a message included the following novel items: the length of the message, swear words, pronouns and self words.

2. DATA SET

The dataset was collected by Aragón et al. [1] using the Twitter API that we extended by a crawl of the user network. Our data set hence consists of two parts:

- *Tweet dataset:* tweet text and user metadata on the Occupy Wall Street movement[1].
- *Follower network:* The list of followers of users who posted at least one message in the tweet dataset.

Table 1 shows the number of users and tweets in the dataset. One can see that a large part of the collected tweets are retweets. Table 2 contains the size of the crawled social networks. Note that the average in- and outdegree is relatively high. Fig. 1 shows the temporal density of tweeting activity.

For each tweet, our data contains

- tweet and user ID,
- timestamp of creation,
- hashtags used in the tweet, and
- the tweet text content.

In case of a retweet, we have all these information not only on the actual tweet, but also on the original *root tweet* that had been retweeted. We define the root tweet as the first occurrence of a given tweet.

3. RETWEET CASCADES

3.1 Constructing retweet cascades

In case of a retweet, the Twitter API provides us with the ID of the original tweet. By collecting retweets for a given original tweet ID, we may obtain the set users who have retweeted a given tweet with the corresponding retweet timestamps. The Twitter API however does not tell us the actual path of cascades if the original tweet was retweeted several times. The information from the Twitter API on the tweet needs to be combined with the follower network to reconstruct the possible information pathways for a given tweet. However it can happen that for a given retweeter, more than one friend has retweeted the corresponding tweet before and hence we do not know the exact information source of the retweeter. The retweet ambiguity problem is well described in [3]. In what follows we consider all friends as possible information sources. In other words for a given tweet we consider all directed edges in the follower network in which information flow could occur (see Fig. 2 (a)).

3.2 Restoring missing cascade edges

For a given tweet, the computed edges define us a *retweet cascade*. However our dataset contains only a sample of tweets on the given hashtags and hence may not be complete: it can happen that a few intermediate retweeters are missing from our data. As a result, sometimes the reconstructed cascade graphs are disconnected. As detailed in Fig. 2 (b) and (c), we handle this problem in two different ways. One possible solution is to only consider the first connected component of the cascade (see Fig. 2 (b)). Another one is to connect each disconnected part to the root tweeter with one virtual cascade edge (see Fig. 2 (c)). In what follows, we work with cascades that contain virtual edges, therefore every retweeter is included in the cascade.

4. FEATURE ENGINEERING

[1]http://en.wikipedia.org/wiki/Occupy_Wall_Street

To train our models, we generate features for each root tweet in the data and then we predict the future cascade size of the root tweet from these feature sets. For a given root tweet, we compute features about

- the author user (*user features*),
- the the follower network of the author (*network features*) and
- the textual content of the tweet itself (*content features*).

Table 3 gives an overview of the feature templates used in our experiments.

4.1 Network Features

We consider statistics about the user and her cascades in the past as well as the influence and impressibility of her followers. We capture the influence and impressibility of a user from previously observed cascades by measuring the following quantities:

- *Number of tweets in different time frames:* for a given root tweet appeared in time t and a predefined time frame τ, we count the number of tweets generated by the corresponding user in the time interval $[t - \tau, t]$. We set τ for 1, 6, 12, 24, 48 and 168 hours.
- *Average number of tweets in different time frames:* We divide the number of tweets in a given time frame by τ.
- *User influence:* for a given user, we compute the number of times one of her followers retweeted her, divided by the number of the followers of the user.
- *User impressibility:* for a given user, we compute the number of times she retweeted one of her followees, divided by the number of followees of the user.

4.2 Content features

The first step of content processing is text normalization. We converted the text them into lower case form except those which are fully upper cased and replaced tokens by their stem given by the Porter stemming algorithm. We replaced user mentions (starting with '@') and numbers by placeholder strings and removed the punctuation marks.

The *content features* are extracted from the normalized texts. The basic feature template in text analysis consists the *terms* of the message. We used a simple whitespace tokenizer rather than a more sophisticated linguistic tokenizer as previous studies reported its empirical advantage [15]. We employed unigrams and bigrams of tokens because longer phrases just hurt the performance of the system in our preliminary experiments.

Besides terms, we extracted the following features describing the *orthography* of the message:

- *Hashtags* are used to mark specific topics, they can be appended after the tweets or inline in the content, marked by #. From the counts of hashtags the user can tips the topic categories of tweet content but too many hashtag can be irritating to the readers as they just make confusion.
- *Telephone number:* If the tweet contains telephone number it is more likely to be spam or ads.
- *Urls:* The referred urls can navigate the reader to text, sound, and image information, like media elements and journals thus they can attract interested readers. We distinguish between full and truncated urls. The truncated urls are ended with three dot, its probably copied from other tweet content, so it was interested by somebody.
- The *like sign* is an illustrator, encouragement to others to share the tweet.
- The presence of *question mark* indicate uncertainty. In Twitter they are usually a rhetorical question rather than a concrete question (people do not search answer on Twitter). The author more likely want to made the reader to think on what contains the message.
- The *Exclamation mark* highlight the part of the tweet, it express emotions and opinions.
- If *Numerical expressions* are present the facts are quantified then it is more likely to have real information content. The actual value of numbers were ignored.
- *Mentions:* If a user mentioned (referred) in the tweet the content of the tweet is probably connected to the mentioned user. It can have informal or private content.
- *Emoticons* are short character sequences representing emotions. We clustered the emoticons into positive, negative and other categories.

The last group of content features tries to capture the *modality* of the message:

- *Swear words* occurring influence the style and attractiveness of the tweet. The reaction for swearing can be ignorance and also reattacking, which is not relevant in terms of retweet cascade size prediction. We extracted the swear word list from `http://www.youswear.com`.
- *Weasel words and phrases*[2] aimed at creating an impression that a specific and/or meaningful statement has been made when in fact only a vague or ambiguous claim has been communicated. We used the weasel word lexicon of [21].
- We employed the linguistic inquiry categories (LIWC) [20] of the tweets' words as well. These categories describe words from emotional, cognitive and structural points of view. For example the "ask" word it is in Hear, Senses, Social and Present categories. Different LIWC categories can have different effect on the influence of the tweet in question.

[2]See `http://en.wikipedia.org/wiki/Wikipedia:Embrace_weasel_words`.

Table 3: Feature set.

user	*number of* {followers, tweets, root tweets}, *average* {cascade size, root cascade size}, *maximum* {cascade size, root cascade size}, *variance* of {cascade sizes, root cascade sizes}, *number of* tweets generated with different time frames, *time average* of the number of tweets in different time frames
network	tweeter's influence and impressibility followers' average influence and impressibility
terms	normalized *unigrams and bigrams*
ortho-graphic	number of # with the values 0, 1, 2 . . . 4 or 4 < number of {like *signs*, ?, !, mentions} number of full and truncated *urls* number of arabic *numbers* and *phone numbers* number of positive/negative/other *emoticons*
modality	number of swear words and weasel phrases union of the *inquiry categories* of the words

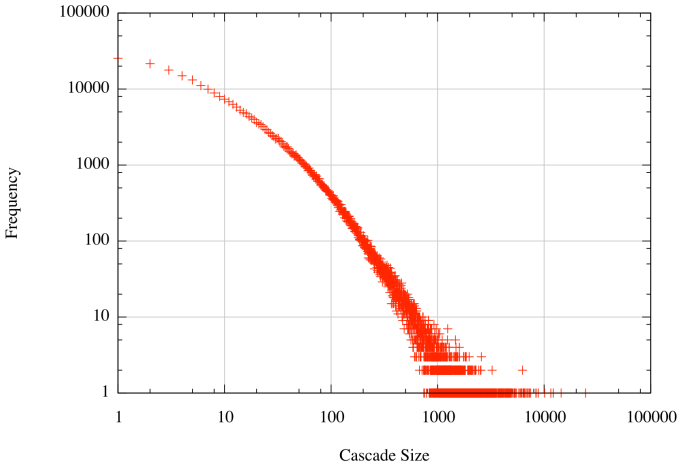


Figure 3: Cascade size distribution.

5. TEMPORAL TRAINING AND EVALUATION

Here we describe the way we generate training and test sets for our algorithms detailed in Section 6. First, for each root tweet we compute the corresponding network and content features. We create daily re-trained models: for a given day t, we train a model on all root tweets that have been generated before t but appeared later than $t - \tau$, where τ is the preset time frame. After training based on the data before a given day, we compute our predictions for all root tweets appeared in that day.

Our goal is to predict cascade size at the time when the root tweet is generated. As the cascade size follows a power law distribution (see Fig. 3), we estimate sizes on the logarithmic scale. In our experiments multi-class classification for ranges of cascade sizes performed better than regression methods for directly predicting the logarithm of the size. We defined three buckets, one with 0. . . 5 (referred as "low"), one with 6. . . 50 ("medium") and a largest one with more than 50 ("high") retweeters participating in the cascade. We trained multiclass random forest classifiers for the three buckets.

We evaluate performance by AUC [10] averaged for the three classes. Note that AUC has a probabilistic interpretation: for the example of the "high" class, the value of the AUC is equal to the probability that a random highly retweeted message is ranked before a random non-highly retweeted one.

By the probabilistic interpretation of AUC, we may realize that a classifier will perform well if it orders the users well with little consideration on their individual messages. Since our goal is to predict the messages in time and not the rather static user visibility and influence, we define new averaging schemes for predicting the success of individual messages.

We consider the classification of the messages of a single user and define two aggregations of the individual AUC values. First, we simply average the AUC values of users for each day (user average)

$$AUC_{\text{user}} = \frac{1}{N}\sum_{i=1}^{N} AUC_i, \tag{1}$$

Second, we are weighting the individual AUC values with the activity of the user (number of tweets by the user for the actual day)

$$AUC_{\text{wuser}} = \frac{\sum_{i=1}^{N} AUC_i T_i}{\sum_i^N T_i} \tag{2}$$

where T_i is the number of tweets by the i-th user.

6. RESULTS

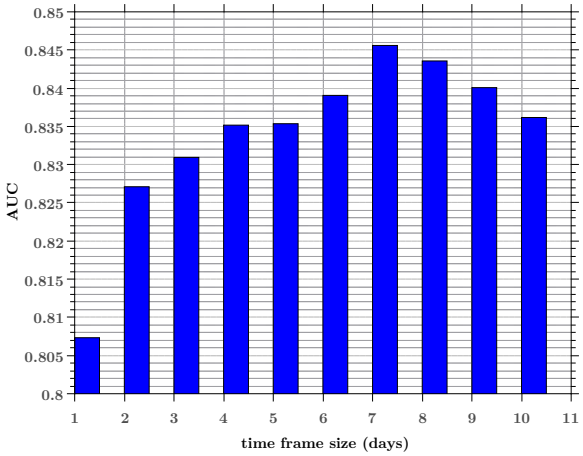


Figure 4: Daily average AUC of classifiers trained with different set of features.

For each day in the testing period, we train a random forest [9] classifier to predict the future retweet size of tweets appearing on that day.

First, we measure classifier performance by computing the average AUC values of the final results for the three size ranges.

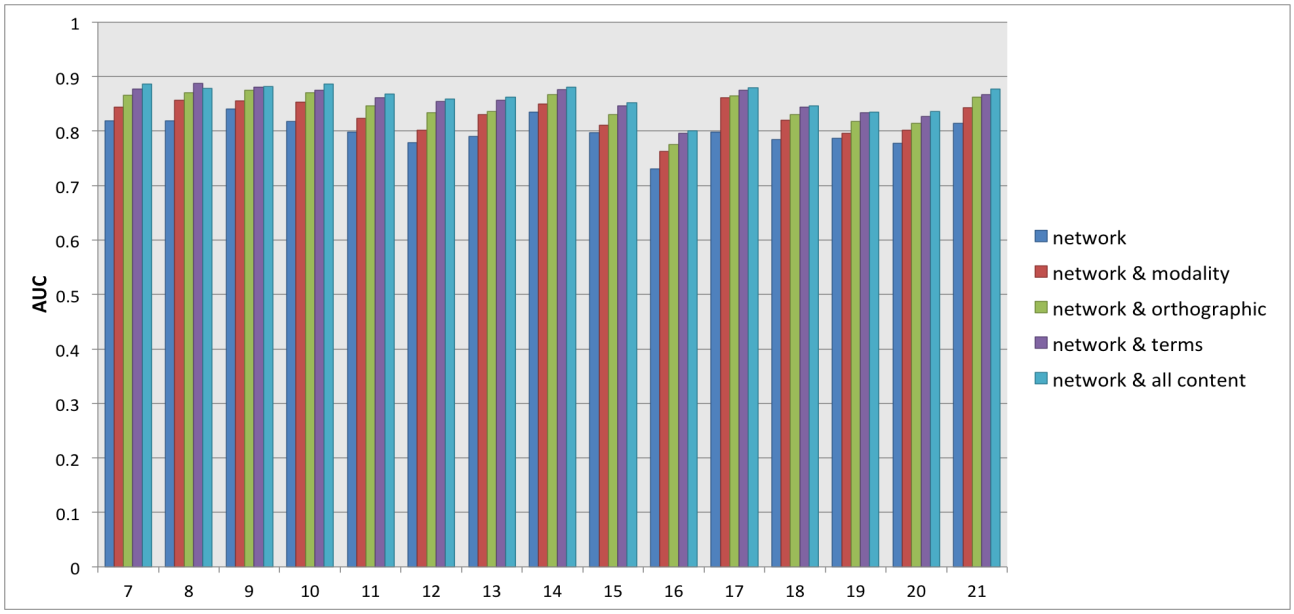


Figure 5: Daily average AUC of classifiers trained with different set of features.

Table 4: Retweet size classification daily average performance of different feature sets

Method \ Retweet range		Low	Medium	High	Weighted Average
network	AUC	0.799	0.785	0.886	0.799
network & modality	AUC	0.827	0.814	0.905	0.827
network & orthographic	AUC	0.844	0.829	0.912	0.843
network & terms	AUC	0.857	0.847	0.914	0.857
network & all content	AUC	0.862	0.849	0.921	0.862

Table 5: Retweet size classification daily average performance of different feature sets evaluated on the user level as defined in equations (1) and (2).

Method \ Retweet range		Low		Medium		High		Average	
		Uniform	Weighted	Uniform	Weighted	Uniform	Weighted	Uniform	Weighted
network	AUC	0.684	0.712	0.752	0.800	0.746	0.796	0.719	0.756
network & modality	AUC	0.700	0.722	0.751	0.796	0.737	0.756	0.726	0.757
network & orthographic	AUC	0.702	0.731	0.753	0.797	0.768	0.782	0.730	0.764
network & terms	AUC	0.705	0.732	0.757	0.800	0.767	0.786	0.733	0.766
network & all content	AUC	0.740	0.783	0.763	0.812	0.769	0.820	0.752	0.797

As mentioned in Section 5, we may train our model with different time frames. In Figure 4 we show the average AUC value with different time frames. As Twitter trends change rapidly, we achieve the best average results if we train our algorithms on root tweets that were generated in the previous week (approximately seven days).

We were interested in how different feature sets affect classifier performance. For this reason we repeated our experiments with different feature subsets. Figure 5 shows our results. For each day, the network features give a strong baseline. The combination of these features with the content result in strong improvement in classifier performance. In Table 4 we summarize the average AUC values for different feature subsets over all four datasets. Our results are consistent: in each case the content related features improve the performance.

Our main evaluation is found in Table 5 where we consider the user level average AUC values as described in Section 5. As expected, since the new evaluation metrics give more emphasis on distinguishing between the tweets of the same user, we see even stronger gain of the modality and orthographic features.

7. CONCLUSIONS AND FUTURE WORK

In this paper we investigated the possibility of predicting the future popularity of a recently appeared text message in Twitter's social networking system. Besides the typical user and network related features, we consider hashtag and linguistic analysis based ones as well. Our results do not only confirm the possibility of predicting the future popularity of a tweet, but also indicate that deep content analysis is important to improve the quality of the prediction.

In our experiments, we give high importance to the temporal aspects of the prediction: we predict immediately after the message is published, and we also evaluate on the user level. We consider user level evaluation key in temporal analysis, since the influence and popularity of a given user is relative stable while the retweet count of her particular messages may greatly vary in time.

Acknowledgments

We thank Andreas Kaltenbrunner for providing us with the Twitter data set [1].

The publication was supported by the KTIA_AIK_12-1-2013-0037 and the PIAC_13-1-2013-0197 projects. The projects are supported by Hungarian Government, managed by the National Development Agency, and financed by the Research and Technology Innovation Fund. Work conducted at the Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI) was supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956), by the Momentum Grant of the Hungarian Academy of Sciences, and by OTKA NK 105645. Work conducted at the Technical University Budapest has been developed in the framework of the project "Talent care and cultivation in the scientific workshops of BME" project. This project is supported by the grant TÁMOP - 4.2.2.B-10/1–2010-0009.

8. REFERENCES

[1] P. Aragón, K. E. Kappler, A. Kaltenbrunner, D. Laniado, and Y. Volkovich. Communication dynamics in twitter during political campaigns: The case of the 2011 spanish national election. *Policy & Internet*, 5(2):183–206, 2013.

[2] E. Bakshy, D. Eckles, R. Yan, and I. Rosenn. Social influence in social advertising: evidence from field experiments. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 146–161. ACM, 2012.

[3] E. Bakshy, J. M. H., W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 65–74. ACM, 2011.

[4] E. Bakshy, B. Karrer, and L. A. Adamic. Social influence and the diffusion of user-created content. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 325–334. ACM, 2009.

[5] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, pages 519–528. ACM, 2012.

[6] M. S. Bernstein, E. Bakshy, M. Burke, and B. Karrer. Quantifying the invisible audience in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 21–30. ACM, 2013.

[7] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.

[8] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, pages 925–936. International World Wide Web Conferences Steering Committee, 2014.

[9] FastRandomForest. Re-implementation of the random forest classifier for the weka environment. `http://code.google.com/p/fast-random-forest/`.

[10] J. Fogarty, R. S. Baker, and S. E. Hudson. Case studies in the use of roc curve analysis for sensor-based estimates in human computer interaction. In *Proceedings of Graphics Interface 2005*, GI '05, pages 129–136, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

[11] R. Ghosh and K. Lerman. Predicting influential users in online social networks. *arXiv preprint arXiv:1005.4882*, 2010.

[12] V. Gómez, H. J. Kappen, and A. Kaltenbrunner. Modeling the structure and evolution of discussion cascades. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 181–190. ACM, 2011.

[13] V. Gómez, H. J. Kappen, N. Litvak, and A. Kaltenbrunner. A likelihood-based framework for the analysis of discussion threads. *World Wide Web*, pages 1–31, 2012.

[14] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. Tweetcred: Real-time credibility assessment of content on twitter. In *Social Informatics*, volume 8851 of *Lecture Notes in Computer Science*, pages 228–243.

2014.

[15] V. Hangya and R. Farkas. Filtering and polarity detection for reputation management on tweets. In *Working Notes of CLEF 2013 Evaluation Labs and Workshop*, 2013.

[16] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[17] K. Lerman and R. Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. In *Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM)*, 2010.

[18] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *Proceedings of the 3rd International Web Science Conference*, WebSci '11. ACM, 2011.

[19] R. Palovics, B. Daroczy, and A. Benczur. Temporal prediction of retweet count. In *Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on*, pages 267–270. IEEE, 2013.

[20] J. Pennebaker, C. Chung, M. Ireland, A. Gonzales, and R. Booth. The development and psychometric properties of liwc2007. Technical report, University of Texas at Austin, 2007.

[21] Gy. Szarvas, V. Vincze, R. Farkas, Gy. Móra, and I. Gurevych. Cross-genre and cross-domain detection of semantic uncertainty. *Computational Linguistics*, 38(2):335–367, 2012.

[22] A. Wang, T. Chen, and M.-Y. Kan. Re-tweeting from a linguistic perspective. In *Proceedings of the Second Workshop on Language in Social Media*, pages 46–55, 2012.

Modeling Community Growth: densifying graphs or sparsifying subgraphs?

Róbert Pálovics[1,2] András A. Benczúr[1,3]
[1]Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI)
[2]Technical University Budapest
[3]Eötvös University Budapest
{rpalovics, benczur}@ilab.sztaki.hu

ABSTRACT

In this paper we model the properties of growing communities in social networks. Our main result is that small communities have higher edge density compared to random subgraphs and their edge number follows power law in the number of nodes. In other words, *the smaller the community, the larger the relative density.*

Our observation resembles the densification law of Leskovec, Kleinberg and Faloutsos who show that the average degree *increases* super-linearly as the size of the network grows. In our settings, however, densification is natural since the average degree of a *random* subgraph grows linearly. In contrary, sublinear growth translates to *increased relative density* in smaller subgraphs.

Our experiments are carried over Twitter retweets and hashtags as well as a detailed music consumption log from Last.fm. In addition to the social network of Twitter followers and Last.fm friends, key in our experiments is that community subgraphs are defined by media use.

We give theoretical results and simulations to explain our findings. The observed edge density can be explained by a mixture of epidemic growth that infects a uniform random neighbor of the community and a low probability selection of a completely new, isolated element. We also explore the relation of graph densification and subgraph sparsification by simulations over graphs of the Stanford Large Network Dataset Collection.

General Terms

Measurement, Theory

Keywords

Communities, Information spread, Power law, Densification law, Twitter, Last.fm, Community subgraphs, Complex networks, Social networks, SNAP

1. INTRODUCTION

1.1 Densification and sparsification

Part of the appeal of Web 2.0 is to find other people who share similar interests. As an example, Last.fm organizes its social network around music recommendation: users may automatically share their listening habits and at the same time grow their friendship. Based on the profiles shared, users may see what artists friends really listen to the most. Companies such as Last.fm use this data to organize and recommend music to people.

While there are several large network datasets available for research, only a few contain temporal information. We exploit the timely information gathered from services of Twitter and Last.fm to obtain microscopic measurements of influence propagating subgraphs of the social network. We define sequences of subgraphs by selecting users that have listened to the same artist, retweeted certain message or used a given hashtag. In this way we obtain evolving communities ordered in time in a fixed social network.

Our main result is a "subgraph sparsification law" of evolving community subgraphs. In time ordered subgraph sequences of the Twitter and Last.fm networks, we measure an increased edge density compared to the average edge density of the whole network. The edge density, i.e. the average degree of a node within the community follows power law of the node count. The exponent is less than two, hence the edge density growth is slower than quadratic and the *relative* density decreases, larger communities are relatively sparser than smaller communities. To understand the distinction, let us consider a random subgraph of the same size n as a selected community. As n approaches the size of the underlying network, the community and random subgraphs will cover roughly the same edges. For smaller n, hence the density of the community is *above* that of the random subgraph. In this sense, small communities that may only choose from a small n intra-community contacts are *relative* denser than the larger ones. Both the absolute and the relative density follow power law, since the number of edges in a random subgraph is quadratic.

We experiment over two large data sets. In case of Last.fm, our experiments are carried over the two-year "scrobble" history and friendship network of 70,000 Last.fm users with public profile. Last.fm's service is unique in that we may obtain a detailed timeline of how the fan community of an artist grows in time over the network.

Twitter, a mixture of a social network and a news media [13], has in the past years became the largest medium

where users may spread information along their social contacts. In our experiments we use the data set of [1] that consists of the messages and the corresponding user network of four global events. We extend the tweet data with the list of followers of users with public profile who posted at least one message in the tweet dataset. The anonymized network with information spreading subgraphs is available at `https://dms.sztaki.hu/en/download/twitter-influence-subgraphs`.

As introduced before, in Last.fm and Twitter community subgraphs, we measure increasing edge density. As in [18], our subgraphs follow the densification law. However, the relative density *decreases* compared to the average edge density of the whole network. Unlike previous models of network growth, in our experiments the network is fixed and as certain information appears in this network, subgraphs are defined as the set of infected nodes. While the average degree is increasing as more nodes join the graph, this may happen for the simple reason that as larger part of a pre-existing network is explored, more connections are found for each node. Our explanation is similar to that of [21] where a sequence of subgraphs is observed as the network is gradually explored.

As a conclusion, the observed edge density can be explained by a mixture of epidemic growth that infects a random neighbor of the community regardless of the age of its infection and a low probability selection of a completely new, isolated element to the community. We also measure the importance of new isolated nodes and show that initially they dominate the communities.

We find an explanation of the community edge density in network models where new connections tend to close short paths. Such models are the forest fire one [18], the triangle closing variants of [15] and, if we add an edge to the prototype as well, the copying model of [12].

While our prime goal is to model the way communities build in social media, our models have surprising connections to densifying graphs [18, 8], and subgraph sampling [21].

Edges in the Last.fm data are timestamped. This gives us the possibility to investigate the original network densification law in case of Last.fm. We further investigate the relation of network densification and subgraph sparsification by epidemic simulations over graphs of the Stanford Large Network Dataset Collection and observe that simulated information spread in these networks follows the same power law edge density as seen in real communities.

Network growth can be considered as community growth in an unobservable hidden background network. For example, people join social networks (Facebook, LinkedIn, etc.) and expose their connections; organizations and companies exposed their relationship by gradually opening their websites in the past decade. Certain networks that are hard to fit into this category include scientific publications; indeed, the epidemic simulations in these graphs give somewhat less self-explaining exponents.

The rest of this paper is organized as follows. First we give a preview of our main observations, followed by the survey of related results. In Section 2 we give our new models for community growth and enumerate some theoretical consequences of different models of the underlying network. In Section 3 we describe our Last.fm and Twitter data that we use in our measurements in Section 4. The relations of the observations and models are discussed in Section 5.

1.2 Summary of main observations

1. "Densifying" community subgraphs with edge number following power law of node number. Note that actually the smaller subgraphs have higher *relative* density compared to a random subgraph of the same size. This difference however vanishes with the community growth, the subgraph "sparsifies".
2. Power law fraction of nodes with at least one edge within the community, with exponent greater than one. This means that initially a large fraction of the nodes are disconnected and these nodes quickly connect to one another.
3. The edge number in a community as the function of the number nodes with at least one edge also follows power law. Surprisingly, the exponent of this process is the same as the Leskovec-Kleinberg-Faloutsos [18] densification exponent and the exponent of an epidemic spread subgraph. In other words, information spreading over a network and the dynamic growth of the network are similar and closely related processes. The network itself can be considered as a community in a hidden social network.
4. Constant expansion: the number of edges leading out from the set of infected nodes is linear as long as the subgraph is not very large.

1.3 Related results

Bonchi [4] summarizes the data mining aspects of research on social influence. He concludes that "another extremely important factor is the temporal dimension: nevertheless the role of time in viral marketing is still largely (and surprisingly) unexplored", an aspect that is key in our result.

Newman reviews the theoretical background of power-law functions and distributions observed in empirical datasets in [20, 7].

As a social media service, Twitter is widely investigated for influence and spread of information. Twitter influence as followers has properties very different from usual social networks [13]. Deep analysis of influence in terms of retweets and mentions is given in [5]. Notion of influence similar to ours is derived in [6, 2] for Flickr and Twitter cascades, respectively. Cha et al. [5] define influence as "... the power of capacity of causing an effect in indirect intangible ways...". In their key observation, the influence of a user is best characterized by the size of the audience who retweets rather than the size of the follower network. We use the Twitter collection of [1] in our experiments.

Our results build on the measurements and theoretical explanations of network densification detailed in [18, 17, 19, 16]. First of all, these results state that graphs densify over time, i.e. the number of edges grow super-linearly while the average distance *shrinks* in evolving real world networks. In contrast to this observation, older network models assumed that evolving graphs have constant average degree and slowly *growing* diameter. They conclude that it is the degree sequence and not the edge sequence that has effect on the diameter of the graph. In [18] two probabilistic generative models are presented, the Community Guided Attachment and the Forest Fire model, that explain edge densification.

Dorogovtsev and Mendes calls edge densification the "accelerated growth" of the network [8]. They introduce theo-

retical relations between the exponent of the power-law degree distribution and the observed temporal edge densification exponent. Their computations are based on the simple assumption that the degree distribution of the graph is a power-law function of the size of the graph.

More empirical observations of densification laws can be found in [10, 22].

Pedarsani et al. investigates densification law in [21]. They state that edge densification laws can be caused by the fact that measurements on real networks are usually carried out on edges samples from the whole network. In other words, they believe that densification may arise as a feature of the common edges sampling procedure to measure dynamic networks. They show that network growth can be a direct consequence of the sampling process, therefore the sampling process itself is a plausible explanation of network densification laws.

Our experiments differ from all three lines of research (Leskovec et al., Dorogovtsev and Mendes, and Pedarsani et al.) in that we investigate a large number of coexisting subgraphs of a network that we may even consider fixed with only the communities evolving inside. Our communities show the "densification" as in the above results, however, similar to the observation of [21], we claim that the small graphs are in fact relative denser compared to the larger ones.

The results of Leskovec et al., in our terminology, consider extra-community edges as phantom nodes and phantom edges, a part of the network that is not covered by the dataset. While in a large network, this part has indeed a minor effect on the properties of edge densification, they play key role in our investigation of evolving communities.

2. MODELS FOR COMMUNITY GROWTH

2.1 Underlying network models

First we shortly summarize three main types of models for the purpose of community growth in an evolving network: concentrated degree, triangle closing and preferential attachment networks.

Certain network models impose constant degree, for example the small world models [23, 11]. Concentrated degree distribution arises in Erős-Rényi graphs [9].

Certain models build the graph by selecting edges that close triangles or short paths as [15]. The copying model [12] also falls in this category since

The main preferential attachment model is the Barabási-Albert one [3]. There the probability of connecting to a node is proportional to its degree, in other words edges connect to subgraphs based on their density.

2.2 Random node selection

We intend to investigate a model with a fixed underlying network. Nodes join after each other to the community. In every step we select a new joining node uniformly at random. In case of Last.fm that means uniform artist listening. In Twitter this model is equivalent to users that post tweets with a certain hashtag independently from each other. In this case, the expected value of the number of edges in the community is power law but with exponent equal to 2. This can be easily proven. Let E and N mean the total number of nodes and edges in the social network. Let user i an user j be part of the community with probability p, independently. The expected total number of nodes and edges in the subgraph is

$$\langle n \rangle = N \cdot p, \qquad \langle e \rangle = E \cdot p \cdot p,$$

therefore

$$\langle e \rangle \sim n^2.$$

It means that when we pick nodes uniform randomly, the average degree within the community is linear function of the subgraph size.

2.3 Epidemic spread

In the concentrated degree distribution models, the increase in the number of edges by epidemic spread is at least one and at most the maximum degree (or an upper bound such that higher degrees are very unlikely). Hence the number of edges in the community $e(n)$ grow linear with the size n.

To model an epidemic spread in the preferential attachment model, we use our observation that the edge expansion is constant, and hence the average degree within the community is equal to the average degree outside. For this reason, in the preferential attachment model, edges are equally likely to connect to any node and the results of the previous subsection apply. Notice that the observation works only under our assumption of constant expansion. In other models where infection may reach high degree nodes fast, we may have a higher probability for an edge connecting into the community.

Finally in the short path closing models, a new node u joining the community will connect to several of the close neighbors of a contact w. Let us select a contact w from the community A and let k denote the expected fraction of "close" contacts of w that are also shared by u. In this intuitive notion, the increase of the number of edges after u joins the community is hence

$$\Delta e(n) = k \cdot d(w, A). \tag{1}$$

If we assume that $d(w, A)$ is the average degree within A, we obtain $\Delta e(n) = k \cdot e(n)/n$, and the solution of the above equation becomes

$$e(n) = \text{const} \cdot n^k, \tag{2}$$

that is the edge density exponent is the same as the short path closing fraction k.

The value of k generalizes the clustering coefficient and must be at least as large in average. In the triangle closing model, all new edges close a triangle and hence k is equal to the clustering coefficient.

In a mixture of epidemic spread and random node selection, the edge density stays below that of epidemic spread. For concentrated degree distributions and preferential attachment graphs, the exponent remains the same one and two, respectively, with only a smaller constant in the edge count. For the short path closing models, if we follow the epidemic spread with probability c, we simply replace k by $c \cdot k$ in equation (1) and hence we may obtain exponents lower than the clustering coefficient.

3. DATA SETS

3.1 Last.fm

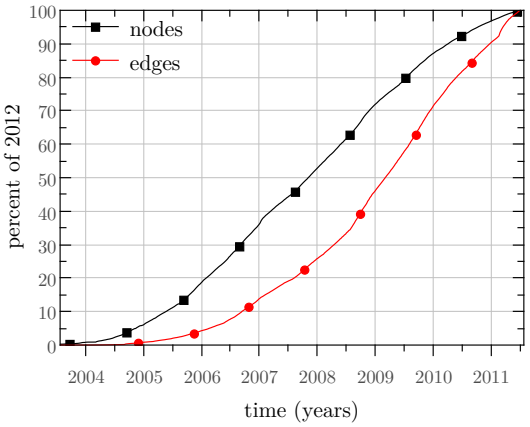


Figure 1: The number of the users and friendship edges in time as the fraction of the values at the time of the data set creation (2012) in the Last.fm dataset.

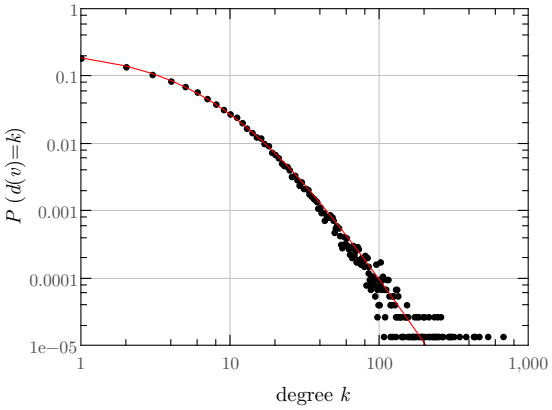


Figure 2: Degree distribution of the Last.fm social network. The distribution follows shifted power law with exponent $\alpha = 3.8$. The estimated shift is $s = 13$.

Last.fm became a relevant online service in music based social networking. The idea of Last.fm is to create a recommendation system based on plugins nearly for all kind of music listening platforms. For registered users it collects, "scrobbles"[1] what they have listened. Each user has its own statistics on listened music that is shown in her profile. Most user profiles are public, and each user of Last.fm may have friends inside the Last.fm social network. We focus on two types of user information,

- the timeline information of users: user u "scrobbled" artist a at time t (u, a, t),
- and the social network of users.

Our data set hence consists of the contacts and the musical taste of the users. For privacy considerations, throughout our research, we selected an anonymous sample of users. Anonymity is provided by selecting random users while maintaining a connected friendship network. We set the following constraints for random selection:

- User location is stated in UK;
- Age between 14 and 50, inclusive;
- Profile displays scrobbles publicly (privacy constraint);
- Daily average activity between 5 and 500.
- At least 10 friends that meet the first four conditions.

The above selection criteria were set to select a representative part of Last.fm users and as much as possible avoid users who artificially generate inflated scrobble figures. In this anonymized data set of two years of artist scrobble timeline, edges of the social network are undirected and timestamped by creation date (Fig. 1). Note that no edges are ever deleted from the network.

The number of users both in the time series and in the network is 71,000 with 285,241 edges. The average degree is therefore 8. The time series contain 979,391,001 scrobbles from 2,073,395 artists and were collected between 01 January 2010 and 31 December 2011. Note that one user can scrobble an artist at different times. The number of unique user-artist scrobbles is 57,274,158.

As the dataset is based on our selection criteria. That means it is not a simple connected part of the network, but a representative part of it. Furthermore, as the edges are timestamped, we not only see a few snapshots of the network, but have a deeper view on the process.

The degree distribution of the underlying social network follows shifted power law distribution

$$P(d(v) = k) = C \cdot (k + s)^{\alpha},$$

with exponent $\alpha = 3.8$ and shift $s = 13$. The relatively large shift is the result of our selection rules.

3.2 Twitter

The dataset was collected by Aragón et al. [1] using the Twitter API that we extended by a crawl of the user network. Our data set hence consists of two parts:

- *Tweet dataset:* tweet text and user metadata on four main global events *15O*[2], *20N*[3] *occupywallstreet*[4], *Yo Soy 132*[5].
- *Follower network:* The list of followers of users who posted at least one message in the tweet dataset.

Table 1 shows the number of users and tweets in case of each dataset. One can see that a large part of the collected tweets are retweets. Table 2 contains the size of the crawled social networks. Note that in all four networks, the average in- and outdegree is relatively high. Fig. 4 shows the in-

[1]The name "scrobbling" is a word by Last.fm, meaning the collection of information about user listening.

[2]http://en.wikipedia.org/wiki/15_October_2011_global_protests
[3]http://en.wikipedia.org/wiki/20-N
[4]http://en.wikipedia.org/wiki/Occupy_Wall_Street
[5]http://en.wikipedia.org/wiki/Yo_Soy_132

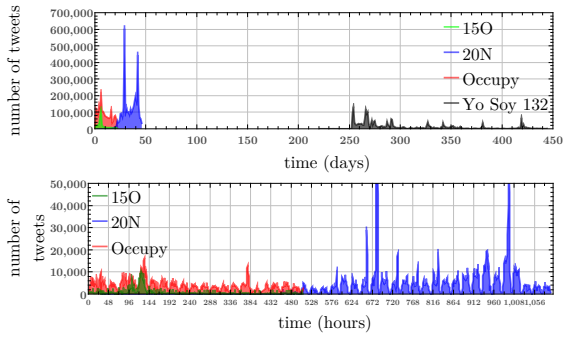


Figure 3: Temporal density of tweeting activity in the four different Twitter datasets.

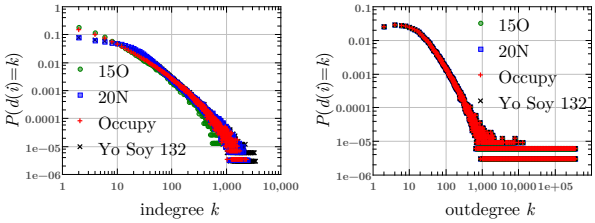


Figure 4: Degree distributions of the Twitter follower networks.

and outdegree distribution of the collected networks. Fig. 3 shows the temporal density of tweeting activity in case of the four different datasets. For each tweet, our data contains

- tweet and user ID,
- timestamp of creation,
- hashtags used in the tweet.

In case of a retweet, we have all these information not only on the actual tweet, but also on the original tweet that had been retweeted.

	15	oc	yo	20
# users	96,935	371,401	395,988	366,155
# tweets	410,482	1,947,234	2,439,109	1,947,234
# hashtags	28,014	93,706	62,008	123,925

Table 1: Sizes of the tweet time series.

	15	oc	yo	20
# users	83,640	330,677	363,452	336,892
# edges	3,093,966	16,585,837	22,054,165	18,809,308
avgdeg.	37	50	61	56

Table 2: Sizes of the follower networks.

3.3 SNAP graphs

We use the following graphs of the Stanford Large Network Dataset Collection[14]:

- **ArXiv** HepPh: Arxiv High Energy Physics paper citation network (phenomenology),
- **ArXiv HepTh**: Arxiv High Energy Physics paper citation network (theory),

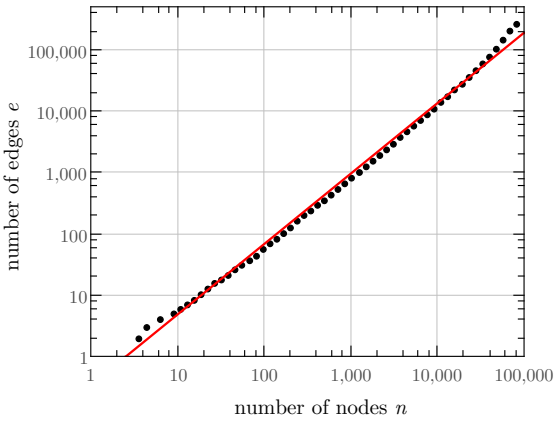


Figure 5: Network densification law in the Last.fm dataset. The number of edges is power law function of the number of nodes in the evolving social network with exponent $\beta = 1.14 - 1.17$.

- **DBLP**: DBLP collaboration network,
- **LiveJournal**: LiveJournal online social network,
- **CAIDA**: The CAIDA AS Relationships Dataset,
- **Google**: Web graph from Google,
- **EU email**: Email network from a EU research institution.

4. EXPERIMENTS

4.1 Network densification

As observed in [18], one common property of complex networks is the edge densification law. As new nodes join in, the number of edges follows a power law of the number of nodes. For Last.fm, we sort the edges by their creation time and then sort the nodes based on this list. Node by node we measure the increase of the number of edges Figure 5. Densification law holds in case of Last.fm with exponent $\beta = 1.14 - 1.17$. Note that regarding to Section 3 no edges were ever deleted from the Last.fm network. Notice that we do not have temporal information on the Twitter follower graph.

4.2 Topical communities

Next we introduce three special community related subsets and define topical communities in Last.fm and Twitter. Let $A(t)$ mean the subset of users in a social network that have adopted a certain topic before time t. As shown in Figure 6, we call a *community subgraph* the graph of users in A.

The "non-zero" component is the subgraph of users that have at least one edge within the community. This component contains all the edges within the community. A contains only isolated nodes besides the "non-zero" component.

The "main component" of A is measured as the one reachable through directed influence edges from the first infected node. With high probability, this is also the largest component. Note that we consider directed reachability, i.e. we do not merge two initial seeds of infected nodes into the same component when they both reach the same new node. Later we investigate the properties of the community subgraph,

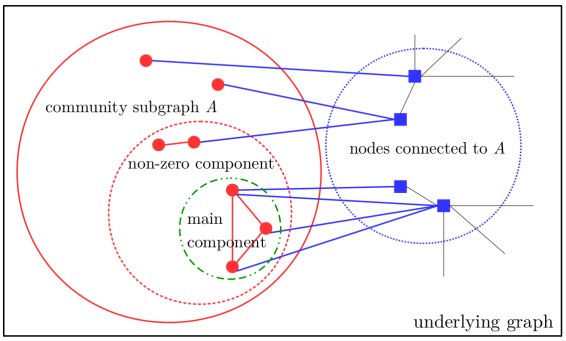


Figure 6: Important subsets of a community subgraph.

the non-zero component, and the main component.

In Last.fm, commmunities are formed by users that have listened to the same artist. $A(t)$ is the subset of users that have scrobbled a given artist at least once before time t.

In case of Twitter a community subgraph is formed by users that have tweeted a given hashtag before time t. In other words we investigate artist subgraphs in Last.fm, and hashtag subgraphs in the Twitter follower network.

In what follows we introduce measurements that result power-law functions related to community subgraphs. Table 3 summarizes the notations and our results in the Last.fm dataset. Table 4 shows the measured exponents for the four Twitter datasets. Next we introduce and investigate these power-law exponents in details. Note that as we have more hashtags than artists, our measurements are more accurate in case of Last.fm than in case of Twitter-. In Table 4 the error of the exponents are roughly 0.05.

4.3 Community subgraph density

To deeper understand the properties of a community subgraph, we set up the following measurement. For each time t a new user adopts the community's topic, we measure the number of edges $e(A, A)$ in the subgraph as the function of the number of users $n = |A|$ in the subgraph. We compute function $e(n)$ for each artist in case of Last.fm and for each different hashtag in Twitter. We average the $e(n)$ curves in case of both social networks. Note that the Twitter follower graph is directed. In that case an edge is part of the subgraph if its source joined earlier to the community than its target. Figure 7 shows our results. In case of Twitter we have four different curves corresponding to the four different datasets. One of our key results is that number of edges is power-law function of the size of the community subgraph

$$e(n) \sim n^{\gamma}. \tag{3}$$

The exponent is 1.52 in Last.fm, and $1.42 - 1.5$ in Twitter communities. Note that we not only averaged the final community subgraphs, but averaged all temporal states of all community subgraphs. Our conclusion is that subgraphs of users with the same activity in a social network show power law growth. Both the number of edges and the average degree are increasing power law function of the number of nodes in the graph.

Figure 8 shows the average degree to the community of the joining node as the function of the community's size. The

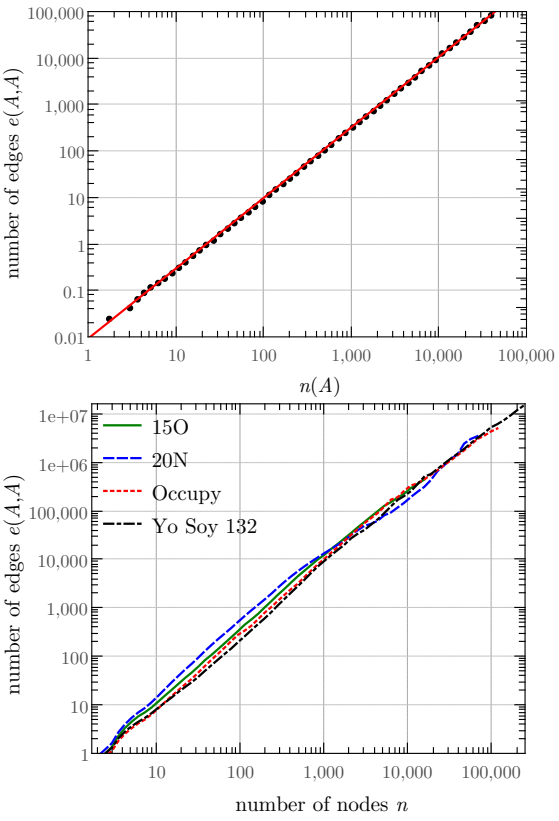


Figure 7: Community subgraph densification in the Last.fm (top) and Twitter (bottom) datasets. The number of edges is power law function of the number of nodes in a community subgraph.Top: Last.fm, Bottom: Twitter.

curves are roughly the derivative of the ones in Figure 7.

4.4 Non-zero degree component

We introduce another power-law result as an explanation of subgraph densification. We can measure the size of the non-zero component z as the function of the size of the subgraph n. That is the number of nodes with non-zero degrees in the subgraph. Figure 9 shows our results. $z(n)$ is a power-law function,

$$z \sim n^{\delta}. \tag{4}$$

Exponent δ is between $1.36 - 1.38$ for Last.fm artists, and 1.1 for Twitter hashtags. Equations (3) and (4) predict that edges in the non-zero component densify with another exponent β_z,

$$e(z) \sim z^{\beta_z}, \qquad \beta_z = \gamma/\delta. \tag{5}$$

We can either compute $\beta_z = \gamma/\delta$ or plot e as the function of z (see Fig. 10). In Last.fm β_z is between 1.15 - 1.17, while it is between 1.31 - 1.38 for Twitter hashtags.

4.5 Main epidemic component

As introduced in Section 4.2, the main component is measured as the one reachable through directed influence edges from the first infected node. Our next measurement (see

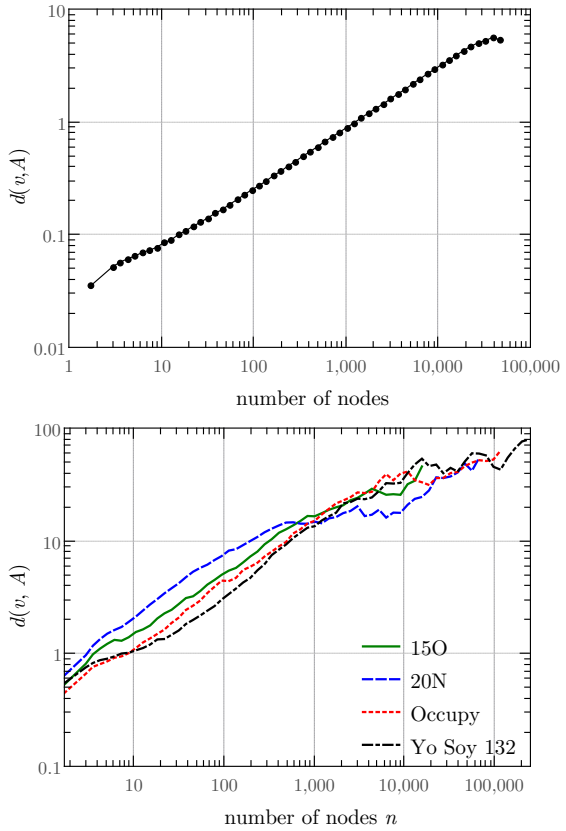


Figure 8: Average degree of the connecting node to the subgraph as the function of the community subgraph size.Top: Last.fm, Bottom: Twitter.

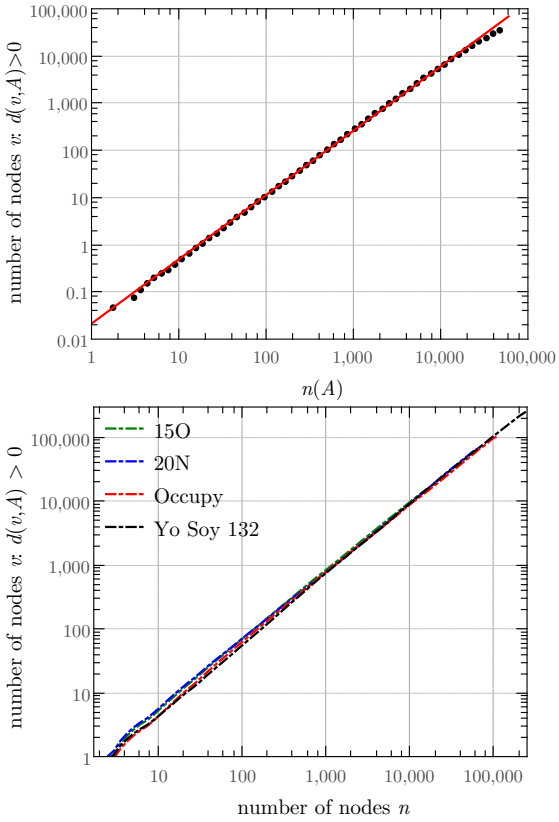


Figure 9: Number of nodes with non-zero degrees as the function of the number of nodes in a community subgraph. Top: Last.fm, Bottom: Twitter.

Fig.10) is that the number of edges ϵ in the main component is power-law function of the size its size m,

$$\epsilon \sim m^{\beta_m}. \quad (6)$$

The corresponding exponent is 1.15 for Last.fm. It is between 1.32 - 1.37 for Twitter networks.

degree distribution α	3.8
network densification β	1.14 - 1.17
subgraph densification γ	1.52
uncorrelated model γ_0	2
non-zero nodes δ	1.36 - 1.38
non-zero component densification β_z	1.15 - 1.17
main component densification β_m	1.15
epidemic β_e	1.14 - 1.15

Table 3: Summary of the most important exponents in th Last.fm dataset.

	γ	δ	β_z	β_m	β_e	β_r
Occupy	1.47	1.1	1.37	1.36	1.35	1.19 - 1.22
Yo Soy 132	1.49	1.1	1.36	1.37	1.27	1.19 - 1.25
20N	1.42	1.1	1.31	1.32	1.27	1.1 - 1.25
15O	1.5	1.07	1.38	1.37	1.32	1.16 - 1.3

Table 4: Exponents in the four Twitter datasets.

4.6 Constant expansion

Figure 11 shows the number of edges leading out from the Last.fm and Twitter communities as the function of the subgraph size. One can observe that in both cases the function is linear as long as long as the subgraph is not very large.

4.7 Epidemic simulations

To investigate the model introduced in Section 2, we simulated epidemic processes in Last.fm, Twitter, and SNAP networks. Starting from a uniform randomly picked node we generated infection processes. At each step we select uniform randomly a node that is not joined to the community, but connected to it in the network (see Fig. 6). Subgraph densification holds for these communities with exponent β_e. Figure 10 shows our results for Last.fm and Twitter networks. Exponents can be found in Table 3 and Table 4. Figure 12 shows our results for SNAP datasets. Table 5 summarizes the exponents for SNAP data. Figure 13 shows for each network the relation of exponent β_e and the average clustering coefficient of the network. Figure 14 shows the number of edges leading out as the function of the epidemic generated community's size.

5. DISCUSSION

In this section we discuss how the network and community densification laws relate to one another and the predictions

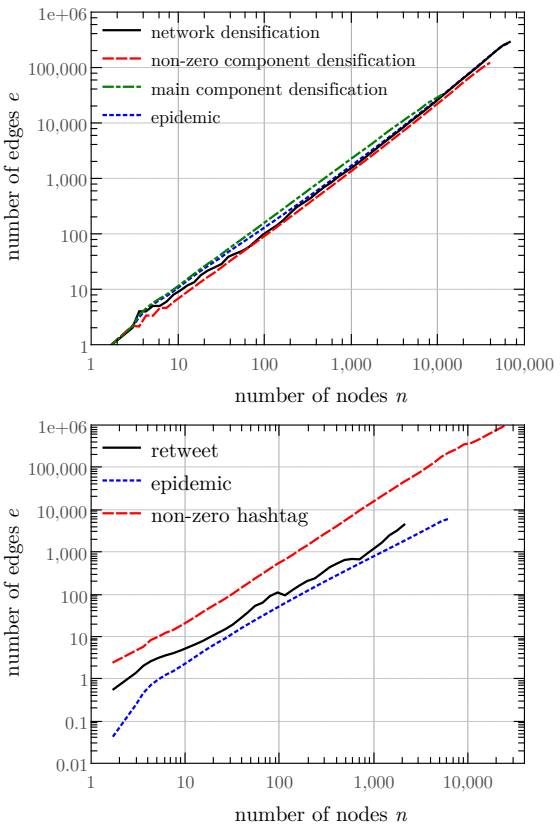


Figure 10: Comparison of different processes with similar exponents. Top: Last.fm, Bottom: Twitter.

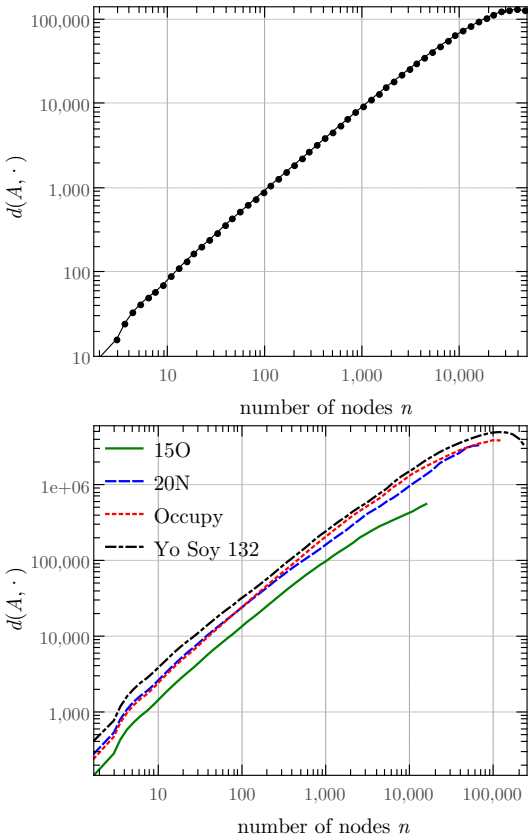


Figure 11: Number of edges leading out from the community subgraph A as the function of the subgraph size. Top: Last.fm, Bottom: Twitter.

network	clustering coefficient	β_e
Last.fm	0.18	1.14
ArXiv HepTh	0.323	1.25
ArXiv HepPh	0.283	1.2
DBLP	0.63	1.06
CAIDA	0.208	1.1
LiveJournal	0.283	1.1
Google	0.5143	1.02
Twitter Occupy	0.12	1.35
EU email	0.0671	1.06

Table 5: β_e and the clustering coefficient in case of different real-world networks.

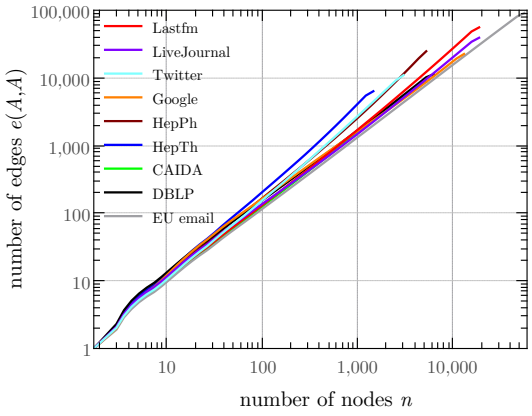


Figure 12: Results of epidemic simulations on various real-world graphs.

of the model in Section 2. Tables 3-4 summarize all power law exponents that we discussed. Here we intend to focus on β, γ and δ.

Figure 16 shows in one plot the result of the epidemic model, the uniform model, and the measured artist subgraph densification law in Last.fm. As introduced in Sections 1-2, the measured curve is between the epidemic model and the uniform random model. This indicates that artist densification in Last.fm is the mixture of an epidemic and a random process. This figure also shows how the relative densification to the random model disappears from the community subgraphs. Larger artist subgraphs are relatively sparser then smaller subgraphs.

Next we compare the values of γ and β_e in case of Last.fm to the exponents measured in case of Twitter. As hashtags can spread with retweets, γ is closer to β_e in hashtag subgraphs than artist subgraphs. In other words information spreading is much stronger in hashtag defined communities then in artist defined ones.

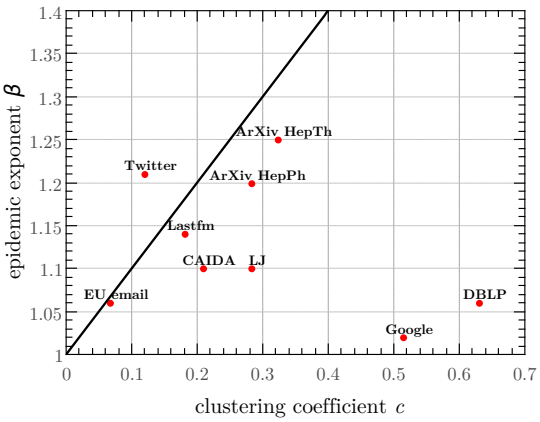


Figure 13: Relation of the epidemic exponent β an the clustering coefficient in case of the 9 different real-world networks.

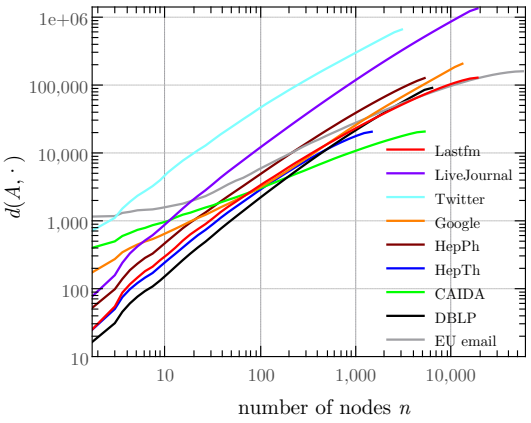


Figure 14: Number of edges leading out from the community subgraph A as the function of the subgraph size.

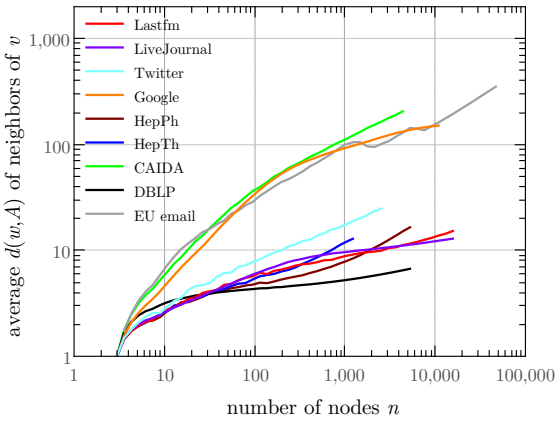


Figure 15: Average degree $d(w, A)$ of neighbors of the recently joined node.

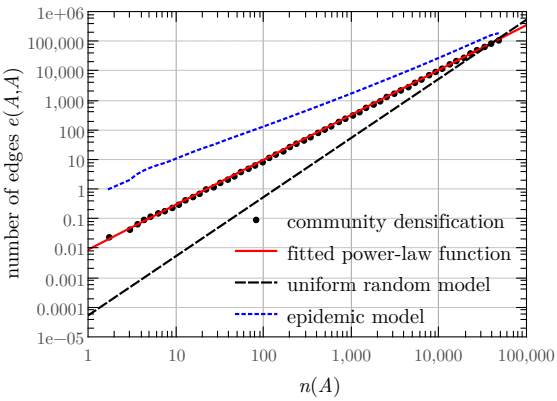


Figure 16: Difference between the measurement and the uniform model in the Last.fm dataset.

Our next observation is that regarding to Tables 3-4,

$$\beta = \beta_e = \beta_z = \beta_m, \tag{7}$$

the four exponents are identical for Last.fm, and similar for Twitter. This result can be seen in Figure 10, where we plotted the curves corresponding to these exponents. Surprisingly, in case of Last.fm, not only the exponents, but the curves are identical. Note that in case of Last.fm we can also observe the network densification law with exponent β. Our results show that epidemic processes over the network are similar to the temporal evolution of the network. Moreover, because of (5) and (7),

$$\gamma = \beta \cdot \delta. \tag{8}$$

This relation between the exponents means that network densification exponent β, and the non-zero exponent δ controls the subgraph densification exponent. Edge density in community subgraphs can be explained by a mixture of epidemic growth that infects a uniform random neighbor of the community and a low probability selection of a completely new, isolated element.

In case of Twitter we computed retweet community subgraphs. As shown in Figure 10, the curve of the epidemic model and the retweet subgraph densification are similar. Note that in contrast to Last.fm, the hashtag curve is over the epidemic curve. However we believe this observation is caused by the quality of the Twitter data. As it is constructed from multiple crawls, we do not have all the edges of the follower network. Moreover, we have less hashtags and retweets than artists in Last.fm.

Next we relate the densification coefficients to the clustering coefficient as in the model of Section 2. As seen in Table 5, for certain networks including Last.fm and the EU email, the two values are very close and for Twitter, even $\beta > k$, indicating a strong tendency to close short paths and connect inside a small community.

For a large number of data sets, however, the clustering coefficient is larger than the densification exponent. As we have no easy-to-define communities in these graphs, the measurements simply indicate that epidemic growth in these networks follow a somewhat different pattern.

In order to investigate graphs with $\beta < k$ further, we identify the reason for the deviation from equation (1) in

Section 2. There we assumed that the degree $d(w, A)$ of the existing member of the community who joins the new member u does not deviate from the average. In particular, $d(w, A)$ should follow the power law $e(n)/n$. In Figure 15 we see that the more a network deviates from the $\beta \approx k$ rule, the quicker a decay in the increase of the average degree happens. The effect of the decayed growth of $d(w, A)$ is lower edge count compared to our model. While the behavior of epidemic spread in these networks is not directly in the scope of this paper, we emphasize this finding as a potential phenomenon that needs further explanation.

6. CONCLUSIONS

In this paper we investigated the properties of growing communities in social networks. We used data from popular social networking sites Last.fm and Twitter to study in details the evolution of communities in large graphs.

We introduced the community subgraph sparsification law. To understand this effect, we carried over numerous of measurements, that resulted various power-law functions between specific quantities related to community subgraphs. We explained the theoretical background and the relation of these power-law exponents. The results of our experiments show that the observed edge density in a community can be explained by a mixture of epidemic growth that infects a uniform random neighbor of the community and a low probability selection of a completely new, isolated element. According to our results epidemic driven community growth is similar to the original network densification: network growth can be considered as community growth in an unobservable social network.

7. ACKNOWLEDGEMENTS

To the Last.fm team for preparing us this volume of the anonymized data set that cannot be efficiently fetched through the public Last.fm API.

To Andreas Kaltenbrunner for providing us with the Twitter data set [1].

Research supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956) and by the grant OTKA NK 105645.
Support from the "Momentum - Big Data" grant of the Hungarian Academy of Sciences.
The work of Robert Palovics reported in this paper has been developed in the framework of the project "Talent care and cultivation in the scientific workshops of BME" project. This project is supported by the grant TAMOP - 4.2.2.B-10/1–2010-0009. Work conducted at the Eötvös University, Budapest was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/-KONV-2012-0013). The research was carried out as part of the EITKIC_12-1-2012-0001 project, which is supported by the Hungarian Government, managed by the National Development Agency, financed by the Research and Technology Innovation Fund and was performed in cooperation with the EIT ICT Labs Budapest Associate Partner Group. (`www.ictlabs.elte.hu`)

8. REFERENCES

[1] P. Aragón, K. E. Kappler, A. Kaltenbrunner, D. Laniado, and Y. Volkovich. Communication dynamics in twitter during political campaigns: The case of the 2011 spanish national election. *Policy & Internet*, 5(2):183–206, 2013.

[2] E. Bakshy, J. M. H., W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 65–74. ACM, 2011.

[3] A.-L. Barabási, R. Albert, and H. Jeon. Mean-field theory for scale-free random network. *Physica A*, 272:173–187, 1999.

[4] F. Bonchi. Influence propagation in social networks: A data mining perspective. *IEEE Intelligent Informatics Bulletin*, 12(1):8–16, 2011.

[5] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.

[6] M. Cha, A. Mislove, B. Adams, and K. P. Gummadi. Characterizing social cascades in flickr. In *Proceedings of the first workshop on Online social networks*, pages 13–18. ACM, 2008.

[7] A. Clauset, C. R. Shalizi, and M. E. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[8] S. Dorogovtsev and J. Mendes. Accelerated growth of networks in handbook of graphs and networks: From the genome to the internet., 2002.

[9] P. Erdős and A. Rényi. On the evolution of random graph. *Math. Inst.*, 1960.

[10] J. S. Katz. Scale-independent bibliometric indicators. *Measurement: Interdisciplinary Research and Perspectives*, 3(1):24–28, 2005.

[11] J. Kleinberg. The Small-World Phenomenon: An Algorithmic Perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, 2000.

[12] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1–10, 2000.

[13] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[14] J. Leskovec. Stanford large network dataset collection. *URL http://snap. stanford. edu/data/index. html*, 2011.

[15] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470. ACM, 2008.

[16] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *The Journal of Machine Learning Research*, 11:985–1042, 2010.

[17] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2006.

[18] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. ACM, 2005.

[19] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.

[20] M. E. Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[21] P. Pedarsani, D. R. Figueiredo, and M. Grossglauser. Densification arising from sampling fixed graphs. *ACM SIGMETRICS Performance Evaluation Review*, 36(1):205–216, 2008.

[22] S. Redner. Citation statistics from more than a century of physical review. *arXiv preprint physics/0407137*, 2004.

[23] D. J. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

Supremum–Norm Convergence for Step–Asynchronous Successive Overrelaxation on M-matrices

Sebastiano Vigna*

April 13, 2014

Abstract

Step-asynchronous successive overrelaxation updates the values contained in a single vector using the usual Gauß–Seidel-like weighted rule, but arbitrarily mixing old and new values, the only constraint being temporal coherence—you cannot use a value before it has been computed. We show that given a nonnegative real matrix A, a $\sigma \geq \rho(A)$ and a vector $\boldsymbol{w} > 0$ such that $A\boldsymbol{w} \leq \sigma\boldsymbol{w}$, every iteration of step-asynchronous successive overrelaxation for the problem $(sI - A)\boldsymbol{x} = \boldsymbol{b}$, with $s > \sigma$, reduces geometrically the $\boldsymbol{w}$-norm of the current error by a factor that we can compute explicitly. Then, we show that given a $\sigma > \rho(A)$ it is in principle always possible to compute such a $\boldsymbol{w}$. This property makes it possible to estimate the supremum norm of the absolute error at each iteration without any additional hypothesis on A, even when A is so large that computing the product $A\boldsymbol{x}$ is feasible, but estimating the supremum norm of $(sI - A)^{-1}$ is not.

Mathematical Subject Classification: 65F10 (Iterative methods for linear systems)

Keywords: Successive overrelaxation; M-matrices; asynchronous iterative solvers

1 Introduction

We are interested in providing *computable absolute bounds in ℓ_∞ norm* on the convergence of a mildly asynchronous version of successive overrelaxation (SOR) applied to problems of the form $(sI - A)\boldsymbol{x} = \boldsymbol{b}$, where A is a nonnegative real matrix and $s > \rho(A)$. A matrix of the form $sI - A$ under these hypotheses is called a *nonsingular M-matrix* [BP94].

We stress from the start that there are no other hypotheses on A such as irreducibility, symmetry, positive definiteness or (weak) 2-cyclicity, and that A is assumed to be very large—so large that computing $A\boldsymbol{x}$ (or performing a SOR iteration) is feasible (maybe streaming over the matrix entries), but estimating $\left\|(sI - A)^{-1}\right\|_\infty$ is not.

*The author was supported by the EU-FET grant NADINE (GA 288956).

Our main motivation is the parallel computation with arbitrary guaranteed precision of various kinds of *spectral rankings with damping* [Vig09], most notably Katz's index [Kat53] and PageRank [PBMW98], which are solutions of problems of the form above with A derived from the adjacency matrix of a very large graph, the only relevant difference being that the rows of A are ℓ_1-normalized in the case of PageRank.

By "computable" we mean that there must be a finite computational process that provides a bound on $\left\|\bar{\boldsymbol{x}} - \boldsymbol{x}^{(t)}\right\|_\infty$, where $\bar{\boldsymbol{x}}$ is the solution and $\boldsymbol{x}^{(t)}$ is the t-th approximation. Such a bound would make it possible to claim that we know the solution up to some given number of significant fractional digits. For example, without further assumptions on A convergence results based on the spectral radius are not computable in this sense and results concerning the residual are not applicable because of the unfeasibility of estimating $\left\|(sI - A)^{-1}\right\|_\infty$.

We are also interested in highly parallel versions for modern multicore systems. While SOR and other iterative methods are apparently strictly sequential algorithms, there is a large body of literature that studies what happens when updates are executed in arbitrary order, mixing old and new values. Essentially, as long as old values come from a finite time horizon (e.g., there is a finite bound on the "oldness" of a value) convergence has been proved for all major standard sequential hypothesis of convergence[1] (for the main results, see the sections about *partial asynchrony* in Bertsekas and Tsitsiklis's encyclopedic book [BT89]).

Again, however, results are always stated in terms of convergence in the limit, and the speed of convergence, which decays as the time horizon gets larger, often cannot be stated explicitly. Moreover, the theory is modeled around message-passing systems, where processor might actually use very old values due to transmission delays. In the multicore, shared-memory system application we have in mind it is reasonable to assume that after each iteration memory is synchronized and all processors have the same view.

Our main motivation is obtaining (almost) "noise-free" scores to perform accurate comparisons of the induced rankings using Kendall's τ [Ken45]:

$$\tau(\boldsymbol{r}, \boldsymbol{s}) := \frac{\sum_{i<j} \operatorname{sgn}(r_i - r_j) \operatorname{sgn}(s_i - s_j)}{\sqrt{\sum_{i<j} \operatorname{sgn}(r_i - r_j)^2} \sqrt{\sum_{i<j} \operatorname{sgn}(s_i - s_j)^2}}.$$

Computational noise can be quite problematic in evaluating Kendall's τ because the signum function has no way to distinguish large and small differences—they are all mapped to 1 or -1 [BPSV08].

Suppose, for example, that we have a graph with a large number n of nodes, and some centrality index that assigns score 0 the first $n/2$ nodes and score 1 the remaining nodes. Suppose we have also another index assigning the same scores, and that this new index is defined by an iterative process, which is stopped at some point (e.g., an iterative solver for linear systems). If the computed values include computational random noise and evaluate τ on the two vectors, we will obtain a τ close to $1/\sqrt{2} \approx 0.707$, even if the ranks are perfectly correlated. On the other

[1] It is a bit surprising, indeed, that the statement that Gauß–Seidel is difficult to parallelize appears so often in the literature. In a sense, an algorithm updating in arbitrary order using possibly old values is not any longer Gauß–Seidel. On the other hand, this is exactly what one expects when asking the question "is Gauß–Seidel parallelizable"?

hand, with a sufficiently small guaranteed absolute error we can proceed to truncate or round the second set of scores, obtaining a result closer to the real correlation.

This scenario is not artificial: when comparing, for instance, indegree with an index computed iteratively (e.g., Katz's index, PageRank, etc.), we have a similar situation. Surprisingly, the noise from iterative computations can even *increase* correlation (e.g., between the dominant eigenvector of a graph that is not strongly connected and Katz's index, as the residual score in nodes whose actual score is zero induces a ranking similar to that induced by Katz's index).

In this paper, we provide convergence bounds in ℓ_∞ norm for SOR iterations for the problem $(sI - A)\boldsymbol{x} = \boldsymbol{b}$, where A is a nonnegative real matrix and $s > \rho(A)$, in conditions of mild asynchrony, without any additional hypothesis on A. Our main result are Theorem 1, which shows that given a $\sigma < s$ and a vector $\boldsymbol{w} > 0$ such that $A\boldsymbol{w} \leq \sigma\boldsymbol{w}$ SOR iterations reduce geometrically the $\boldsymbol{w}$-norm of the error (with a computable contraction factor), and Theorem 2, which shows how to compute such a $\boldsymbol{w}$ using only iterated products of A with a vector. The two results can be viewed as a constructive and computable version of the standard convergence results on SOR iteration based on the spectral radius.

We remark that SOR is actually not useful for PageRank, as shown recently by Greif and Kurokawa [GK11]. The author has found experimentally that the same phenomenon plagues the computation of Katz's index. However, since generalizing from Gauß–Seidel to SOR does not bring any significant increase in complexity in the proof, we decided to prove our results in the more general setting.

2 Step-asynchronous SOR

We now define *step-asynchronous* SOR for the problem $(sI - A)\boldsymbol{x} = \boldsymbol{b}$. In general, *asynchronous* SOR computes new values using arbitrarily old values; in this case, the hypotheses for convergence are definitely stronger. In the *partially asynchronous* case, instead, there is a finite limit on the "oldness" of the values used to compute new values, and while there is a decrease in convergence speed, the hypotheses for convergence are essentially the same of the sequential case (see [BT89] for more details).

Step-asynchronous SOR uses the strictest possible time bound: one step. We thus perform a SOR-like update in arbitrary order:

$$x_i^{(t+1)} = (1-\omega)x_i^{(t)} + \frac{\omega}{s - a_{ii}}\Big(b_i + \sum_{j \in N_i^{(t)}} a_{ij}x_j^{(t+1)} + \sum_{j \in P_i^{(t)} \setminus \{\, i \,\}} a_{ij}x_j^{(t)}\Big). \tag{1}$$

The only constraint is that for each iteration an *update total preorder*[2] $\preceq^{(t)}$ of the indices is given: $i \preceq^{(t)} j$ iff x_i is updated before (or at the same time of) x_j at iteration t, and the set $P_i^{(t)}$ of the indices for which we use the *previous* values is such that for all $j \succeq^{(t)} i$ we have $j \in P_i^{(t)}$, whereas $N_i^{(t)} = n \setminus P_i(t)$ is the set indices for which we use the *next* values. Essentially, we *must* use previous values for all variables that are updated at the same time of x_i or after x_i, but we make

[2] A *total preorder* is a set endowed with a reflexive and transitive total relation. We remark that a choice of a sequence of such preorders is equivalent to a *scenario* in the terminology of [BT89].

no assumption on the remaining variables. In this way we take into account cache incoherence, unpredictable scheduling of multiple threads, and so on.[3]

Matrixwise, the set $N_i^{(t)}$ induces a nonnegative matrix $L^{(t)}$ given by

$$L_{ij}^{(t)} = \left[j \in N_i^{(t)}\right] a_{ij}$$

and a *regular splitting*

$$sI - A = \big(D - L^{(t)}\big) - R^{(t)},$$

where $D = sI - \operatorname{Diag}(A)$ and $R^{(t)}$ is nonnegative with zeros on the diagonal. Then, equation (1) can be rewritten as

$$\big(D - \omega L^{(t)}\big)\boldsymbol{x}^{(t+1)} = (1-\omega)D\boldsymbol{x}^{(t)} + \omega\big(\boldsymbol{b} + R^{(t)}\boldsymbol{x}^{(t)}\big).$$

There is of course a permutation of row and columns (depending on t) such that $L^{(t)}$ is strictly lower triangular, but the only claim that can be made about $R^{(t)}$ is that its diagonal is zero: actually, we could have $L^{(t)} = 0$ and $R^{(t)} = sI - A - D$.

In particular, independently from the choice of $L^{(t)}$, if $\bar{\boldsymbol{x}}$ is a solution we have as usual

$$\big(D - \omega L^{(t)}\big)\bar{\boldsymbol{x}} = (1-\omega)D\bar{\boldsymbol{x}} + \omega\big(\boldsymbol{b} + R^{(t)}\bar{\boldsymbol{x}}\big)$$

and

$$\big(D - \omega L^{(t)}\big)\big(\bar{\boldsymbol{x}} - \boldsymbol{x}^{(t+1)}\big) = (1-\omega)D\big(\bar{\boldsymbol{x}} - \boldsymbol{x}^{(t)}\big) + \omega R^{(t)}\big(\bar{\boldsymbol{x}} - \boldsymbol{x}^{(t)}\big). \tag{2}$$

3 Suitability and convergence in $\boldsymbol{w}$-norm

We now define suitability of a vector for a matrix, which will be the main tool in proving our results. The idea is implicitly or explicitly at the core of several classical proofs of convergence, and is closely related to that of *generalized diagonal dominance*:

Definition 1 *A vector $\boldsymbol{w} > 0$ is* σ-suitable *for A if $A\boldsymbol{w} \leq \sigma\boldsymbol{w}$.*

The usefulness of suitable vectors is that they induce norms norms in which the decrease of the error caused by a SOR iteration for of the problem $(sI - A)\boldsymbol{x} = \boldsymbol{b}$ can be controlled if $s > \sigma$. If A is irreducible, for instance, the dominant eigenvector is suitable for the spectral radius, but it is exactly this kind of hypotheses that we want to avoid.

Definition 2 *Given a vector $\boldsymbol{w} > 0$, the $\boldsymbol{w}$-norm is defined by*

$$\|\boldsymbol{x}\|_\infty^{\boldsymbol{w}} = \max_i \frac{|x_i|}{w_i}.$$

The notation $\|\cdot\|_\infty^{\boldsymbol{w}}$ is used also for the operator norm induced in the usual way. We note a few useful properties—many others can be found in [BT89]:

[3] For example, if we have exactly n parallel updates at the same time we would have, in fact, a Jacobi iteration: in that case, $N_i^{(t)} = \varnothing$ for all i.

Proposition 1 *Given a vector $\boldsymbol{w}$ that is σ-suitable for a nonnegative matrix A, the following statements are true for all vectors $\boldsymbol{x}$:*

1. $|x_i| \leq w_i \|\boldsymbol{x}\|_\infty^{\boldsymbol{w}}$;
2. $\min_i w_i \|\boldsymbol{x}\|_\infty^{\boldsymbol{w}} \leq \|\boldsymbol{x}\|_\infty$;
3. $\max_i w_i \|\boldsymbol{x}\|_\infty^{\boldsymbol{w}} \geq \|\boldsymbol{x}\|_\infty$;
4. $\|\boldsymbol{w}\|_\infty^{\boldsymbol{w}} = 1$;
5. $\|A\|_\infty^{\boldsymbol{w}} = \|A\boldsymbol{w}\|_\infty^{\boldsymbol{w}}$;
6. *if* $\boldsymbol{x} \geq 0$, $\|\boldsymbol{x}\|_\infty^{\boldsymbol{w}} = \min\{\, \alpha \geq 0 \mid \boldsymbol{x} \leq \alpha \boldsymbol{w} \,\}$.
7. $\|A\boldsymbol{x}\|_\infty^{\boldsymbol{w}} \leq \sigma \|\boldsymbol{x}\|_\infty^{\boldsymbol{w}}$; *in particular,* $\rho(A) \leq \|A\|_\infty^{\boldsymbol{w}} \leq \sigma$.

Proof. The first claims are immediate from the definition of $\boldsymbol{w}$-norm. For the last claim,

$$\|A\boldsymbol{x}\|_\infty^{\boldsymbol{w}} = \max_i \left| \frac{\sum_j a_{ij} x_j}{w_i} \right| \leq \max_i \frac{\sum_j a_{ij} |x_j|}{w_i} = \max_i \frac{\sum_j a_{ij} w_j \|\boldsymbol{x}\|_\infty^{\boldsymbol{w}}}{w_i} \leq \sigma \|\boldsymbol{x}\|_\infty^{\boldsymbol{w}}.$$

∎

The next theorem is based on the standard proof by induction of convergence for SOR, but we make induction on the update time of a component rather than on its index, and we use suitability to provide bounds to the norm of the error.

Theorem 1 *Let A be a nonnegative matrix and let $\boldsymbol{w}$ be σ-suitable for A. Then, given $s > \sigma$ step-asynchronous SOR for the problem $(sI - A)\boldsymbol{x} = \boldsymbol{b}$ converges for*

$$0 < \omega < \frac{2}{1 + \max_k \dfrac{\sigma - a_{kk}}{s - a_{kk}}}$$

and letting $\bar{\boldsymbol{x}} = (sI - A)^{-1}\boldsymbol{b}$ we have

$$\left\| \bar{\boldsymbol{x}} - \boldsymbol{x}^{(t+1)} \right\|_\infty^{\boldsymbol{w}} \leq r \left\| \bar{\boldsymbol{x}} - \boldsymbol{x}^{(t)} \right\|_\infty^{\boldsymbol{w}},$$

where

$$r = |1 - \omega| + \omega \max_k \frac{\sigma - a_{kk}}{s - a_{kk}} < 1.$$

Moreover,

$$\left\| \bar{\boldsymbol{x}} - \boldsymbol{x}^{(t+1)} \right\|_\infty^{\boldsymbol{w}} \leq \frac{r}{1 - r} \left\| \boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)} \right\|_\infty^{\boldsymbol{w}}.$$

Proof. Let $\preceq^{(t)}$ be a sequence of update orders, and $P_i(t)$ a sequence of previous-value sets, one for each step t and variable index i, compatible with the respective update orders. We work by induction on the order $\preceq^{(t)}$, proving the statement

$$\left| e_i^{(t+1)} \right| \leq \left(|1 - \omega| + \omega \frac{\sigma - a_{ii}}{s - a_{ii}} \right) w_i \left\| \boldsymbol{e}^{(t)} \right\|_\infty^{\boldsymbol{w}}, \tag{3}$$

where $\boldsymbol{e}^{(t)} = \bar{\boldsymbol{x}} - \boldsymbol{x}^{(t)}$, assuming it is true for all $k \prec^{(t)} i$.
Note that for all i

$$0 < \frac{\sigma - a_{ii}}{s - a_{ii}} < 1,$$

so for $0 < \omega \leq 1$

$$|1-\omega| + \omega\frac{\sigma - a_{ii}}{s - a_{ii}} = 1 - \omega\left(1 - \frac{\sigma - a_{ii}}{s - a_{ii}}\right) < 1,$$

and analogously for

$$1 \leq \omega < \frac{2}{1 + \max\limits_k \dfrac{\sigma - a_{kk}}{s - a_{kk}}}$$

we have

$$|1-\omega| + \omega\frac{\sigma - a_{ii}}{s - a_{ii}} = \omega\left(1 + \frac{\sigma - a_{ii}}{s - a_{ii}}\right) - 1 < \frac{2}{1 + \max\limits_k \dfrac{\sigma - a_{kk}}{s - a_{kk}}}\left(1 + \frac{\sigma - a_{ii}}{s - a_{ii}}\right) - 1 < 1.$$

Writing explicitly (2) for the i-th coordinate, we have

$$\left|e_i^{(t+1)}\right| = \left|(1-\omega)e_i^{(t)} + \frac{\omega}{s - a_{ii}}\Big(\sum_{j \in N_i^{(t)}} a_{ij}e_j^{(t+1)} + \sum_{j \in P_i^{(t)} \setminus \{\, i \,\}} a_{ij}e_j^{(t)}\Big)\right|.$$

Since $j \in N_i^{(t)}$ implies by definition $j \prec^{(t)} i$, we can apply the induction hypothesis on $e_j^{(t+1)}$ to state that $e_j^{(t+1)} \leq w_j \left\|\boldsymbol{e}^{(t)}\right\|_\infty^{\boldsymbol{w}}$. The same bound applies to $e_j^{(t)}$ using the first statement of Proposition 1.
We now notice that σ-suitability implies

$$(A - \mathrm{Diag}(A))\boldsymbol{w} \leq (\sigma I - \mathrm{Diag}(A))\boldsymbol{w},$$

which in coordinates tells us that

$$\sum_{j \neq i} a_{ij}w_j \leq \big(\sigma - a_{ii}\big)w_i.$$

Thus,

$$\begin{aligned}\left|e_i^{(t+1)}\right| &\leq \left(|1-\omega|w_i + \omega\frac{1}{s - a_{ii}}\Big(\sum_{j \in N_i^{(t)}} a_{ij}w_j + \sum_{j \in P_i^{(t)} \setminus \{\, i \,\}} a_{ij}w_j\Big)\right)\left\|\boldsymbol{e}^{(t)}\right\|_\infty^{\boldsymbol{w}} \\ &\leq \left(|1-\omega| + \omega\frac{\sigma - a_{ii}}{s - a_{ii}}\right)w_i\left\|\boldsymbol{e}^{(t)}\right\|_\infty^{\boldsymbol{w}}.\end{aligned}$$

By the very definition of $\boldsymbol{w}$-norm, (3) yields

$$\left\|\boldsymbol{e}_i^{(t+1)}\right\|_\infty^{\boldsymbol{w}} \leq \left(|1-\omega| + \omega \max_k \frac{\sigma - a_{kk}}{s - a_{kk}}\right)\left\|\boldsymbol{e}^{(t)}\right\|_\infty^{\boldsymbol{w}}.$$

For the second statement, we have

$$\begin{aligned}\left\|\boldsymbol{x} - \boldsymbol{x}^{(t)}\right\|_\infty^{\boldsymbol{w}} - \left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|_\infty^{\boldsymbol{w}} &\leq \left\|\boldsymbol{x} - \boldsymbol{x}^{(t+1)} + \boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t)}\right\|_\infty^{\boldsymbol{w}} \\ &= \left\|\boldsymbol{x} - \boldsymbol{x}^{(t+1)}\right\|_\infty^{\boldsymbol{w}} \leq r\left\|\boldsymbol{x} - \boldsymbol{x}^{(t)}\right\|_\infty^{\boldsymbol{w}},\end{aligned}$$

whence

$$\left\|\boldsymbol{x}-\boldsymbol{x}^{(t+1)}\right\|_\infty^{\boldsymbol{w}} \leq r\left\|\boldsymbol{x}-\boldsymbol{x}^{(t)}\right\|_\infty^{\boldsymbol{w}} \leq \frac{r}{1-r}\left\|\boldsymbol{x}^{(t+1)}-\boldsymbol{x}^{(t)}\right\|_\infty^{\boldsymbol{w}}.$$

∎

We remark that the smallest contraction factor is obtained when $\omega = 1$, that is, with no relaxation. This does not mean, however, that relaxation is not useful: convergence might be faster with $\omega \neq 1$; it is just that the error bound we provide features the best constant when $\omega = 1$.

Corollary 1 *With the same hypotheses and notation of Theorem 1, step-asynchronous Gauß–Seidel iterations converge and*

$$\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{(t+1)}\right\|_\infty^{\boldsymbol{w}} \leq \frac{\max_k \dfrac{\sigma-a_{kk}}{s-a_{kk}}}{1-\max_k \dfrac{\sigma-a_{kk}}{s-a_{kk}}}\left\|\boldsymbol{x}^{(t+1)}-\boldsymbol{x}^{(t)}\right\|_\infty^{\boldsymbol{w}}.$$

Corollary 2 *Let A be an irreducible nonnegative matrix and $\boldsymbol{w}$ its dominant eigenvector. Then the statement of Theorem 1 is true in $\boldsymbol{w}$-norm with $\sigma = \rho(A)$.*

A simple consequence is that if we know a σ-suitable vector $\boldsymbol{w}$ for A we can just behave as if the step-asynchronous SOR is converging in the standard supremum norm, but we have a reduction in the strength of the bound given by the ratio between the maximum and the minimum component of $\boldsymbol{w}$:

Corollary 3 *With the same hypotheses and notation of Theorem 1, step-asynchronous Gauß–Seidel iterations converge and*

$$\left\|\bar{\boldsymbol{x}}-\boldsymbol{x}^{(t+1)}\right\|_\infty \leq \frac{\max_i w_i}{\min_i w_i}\frac{\max_k \dfrac{\sigma-a_{kk}}{s-a_{kk}}}{1-\max_k \dfrac{\sigma-a_{kk}}{s-a_{kk}}}\left\|\boldsymbol{x}^{(t+1)}-\boldsymbol{x}^{(t)}\right\|_\infty.$$

Proof. An application of Proposition 1.2 and 1.3. ∎

We remark that

$$\max_k \frac{\sigma-a_{kk}}{s-a_{kk}} \leq \frac{\sigma}{s},$$

so it is possible to restate all results in a simplified (but less powerful) form.

4 Practical issues

In principle it is always better to compute the actual $\boldsymbol{w}$-norm, rather than using the rather crude bound of Corollary 3.[4] On the other hand, computing the $\boldsymbol{w}$-norm requires storing and accessing $\boldsymbol{w}$, which could be expensive.

In practice, it is convenient to restrict oneself to vectors $\boldsymbol{w}$ satisfying $\|\boldsymbol{w}\|_\infty = 1$, as in that case $\|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_\infty^{\boldsymbol{w}}$, and for some $\boldsymbol{x}$ we actually have equality. Then, we

[4] The bound is actually *very* crude, in particular on reducible matrices when σ is close to $\rho(A)$.

can store in few bits an approximate vector $\boldsymbol{w}' \leq \boldsymbol{w}$, which can be used to estimate $\left\|\bar{\boldsymbol{x}}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|_\infty^{\boldsymbol{w}}$, as we have, using the notation of Theorem 1,

$$\left\|\bar{\boldsymbol{x}} - \boldsymbol{x}^{(t+1)}\right\|_\infty \leq \left\|\bar{\boldsymbol{x}} - \boldsymbol{x}^{(t+1)}\right\|_\infty^{\boldsymbol{w}} \leq \frac{r}{1-r}\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|_\infty^{\boldsymbol{w}} \leq \frac{r}{1-r}\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|_\infty^{\boldsymbol{w}'}.$$

A reasonable choice is that of keeping in memory $\lceil -\log_2 w_i \rceil$. Using a byte of storage we can keep track of w_i's no smaller than Moreover, during the evaluation of the norm we just have to multiply by a power of two, which can be done very quickly in IEEE 754 format.

5 Choosing a suitable vector

We now come to the main result: given a nonnegative matrix A and a $\sigma > \rho(A)$, it is possible (constructively) to compute a vector $\boldsymbol{w}$ that is σ-suitable for A. In essence, the computation of a σ-suitable vector for A "tames" the non-normality of the iterative process, at the price of a reduction of the convergence range.

Theorem 2 *Let A be nonnegative and $\sigma > \rho(A)$. Let*

$$\boldsymbol{w}_\sigma^{(k)} = \sum_{i=0}^{k} \left(\frac{A}{\sigma}\right)^i \mathbf{1}$$

and

$$\boldsymbol{w}_\sigma = \lim_{k\to\infty} \boldsymbol{w}_\sigma^{(k)}$$

Then, $A\boldsymbol{w}_\sigma < \sigma\boldsymbol{w}_\sigma$. In particular, $\boldsymbol{w}_\sigma$ is σ-suitable for A, and there is a k such that

$$A\boldsymbol{w}_\sigma^{(k)} \leq \sigma\boldsymbol{w}_\sigma^{(k)},$$

so $\boldsymbol{w}_\sigma^{(k)}$ is σ-suitable for A.

Proof. Consider the matrix $A + \delta\mathbf{1}\mathbf{1}^*$, where $\delta > 0$. Since it is strictly positive, the Perron–Frobenius Theorem tells us that there is a dominant eigenvector $\boldsymbol{w}_\delta > 0$. Moreover, since for $\delta \to \infty$ we have $\rho(A + \delta\mathbf{1}\mathbf{1}^*) \to \infty$, and the spectral radius is continuous in the matrix entries, there must be a δ_σ such that

$$\rho\big(A + \delta_\sigma\mathbf{1}\mathbf{1}^*\big) = \sigma.$$

We have

$$\begin{aligned}
\big(A + \delta_\sigma\mathbf{1}\mathbf{1}^*\big)\boldsymbol{w}_{\delta_\sigma} &= \sigma\boldsymbol{w}_{\delta_\sigma} \\
A\boldsymbol{w}_{\delta_\sigma} + \delta_\sigma\|\boldsymbol{w}_{\delta_\sigma}\|_1\mathbf{1} &= \sigma\boldsymbol{w}_{\delta_\sigma} \\
\frac{\delta_\sigma\|\boldsymbol{w}_{\delta_\sigma}\|_1}{\sigma}\mathbf{1} &= \left(1 - \frac{A}{\sigma}\right)\boldsymbol{w}_{\delta_\sigma} \\
\boldsymbol{w}_{\delta_\sigma} &= \frac{\delta_\sigma\|\boldsymbol{w}_{\delta_\sigma}\|_1}{\sigma}\sum_{i=0}^{\infty}\left(\frac{A}{\sigma}\right)^i\mathbf{1}.
\end{aligned}$$

We now observe that the scaling factor is irrelevant: $\boldsymbol{w}_{\delta_\sigma}$ is an eigenvector, so it is defined up to a multiplicative constant. We can thus just write

$$\boldsymbol{w}_\sigma = \sum_{i=0}^{\infty} \left(\frac{A}{\sigma}\right)^i \mathbf{1}$$

and state that

$$\left(A + \delta_\sigma \mathbf{1}\mathbf{1}^*\right)\boldsymbol{w}_\sigma = \sigma \boldsymbol{w}_\sigma,$$

which implies

$$A\boldsymbol{w}_\sigma = \sigma \boldsymbol{w}_\sigma - \delta_\sigma \|\boldsymbol{w}_\sigma\|_1 \mathbf{1} < \sigma \boldsymbol{w}_\sigma.$$

Thus, as $\boldsymbol{w}_\sigma^{(k)} \to \boldsymbol{w}_\sigma$ when $k \to \infty$, for some k we must have

$$A\boldsymbol{w}_\sigma^{(k)} \leq \sigma \boldsymbol{w}_\sigma^{(k)}.$$

∎

The previous theorem suggests the following procedure. Under the given hypotheses, start with $\boldsymbol{w}^{(0)} = \mathbf{1}$, and iterate

$$\begin{aligned} \boldsymbol{z} &= A\boldsymbol{w}^{(t)} \\ \boldsymbol{w}^{(t+1)} &= \boldsymbol{z}/\sigma + \mathbf{1}. \end{aligned}$$

Note that this is just a Jacobi iteration for the problem $(I - A/\sigma)\boldsymbol{x} = \mathbf{1}$, which is natural, as $\boldsymbol{w}_\sigma$ is just its solution. The iteration stops as soon as

$$\max_i \frac{z_i}{w_i^{(t)}} \leq \sigma, \tag{4}$$

and at that point $\boldsymbol{w}^{(t)}$ is by definition σ-suitable for A, so we can apply Theorem 1. In practice, it is useful to keep the current vector $\boldsymbol{w}^{(t)}$ normalized: just set $s^{(0)} = 1$ at the start, and then iterate

$$\begin{aligned} \boldsymbol{z} &= A\boldsymbol{w}^{(t)} \\ \boldsymbol{u} &= \boldsymbol{z}/\sigma + s^{(t)}\mathbf{1} \\ s^{(t+1)} &= s^{(t)}/\|\boldsymbol{u}\|_\infty \\ \boldsymbol{w}^{(t+1)} &= \boldsymbol{u}/\|\boldsymbol{u}\|_\infty. \end{aligned}$$

We remark that, albeit used for clarity in the statement of Theorem 2, the (exact) knowledge of $\rho(A)$ is not strictly necessary to apply the technique above: indeed, if the procedure terminates $\sigma \geq \rho(A)$ by Proposition 1.

There are a few useful observations about the behavior of the normalized version of the procedure. First, if $\sigma < \rho(A)$ necessarily $s^{(t)} \to 0$ as $t \to \infty$. Second, by Collatz's classical bound [Col42], the maximum in (4) is an upper bound to $\rho(A)$. This happens without additional hypotheses[5] on A because whenever $A\boldsymbol{x} \leq \gamma\boldsymbol{x}$ with $\boldsymbol{x} > 0$ we have

$$\rho(A) \leq \|A\|_\infty^{\boldsymbol{x}} = \|A\boldsymbol{x}\|_\infty^{\boldsymbol{x}} \leq \|\gamma\boldsymbol{x}\|_\infty^{\boldsymbol{x}} = \gamma.$$

[5]We report the following two easy proofs as in most of the literature Collatz's bounds are proved for irreducible matrices using Perron–Frobenius theory.

If, moreover, we compute also the minimum ratio

$$\min_i \frac{z_i}{w_i^{(t)}}, \tag{5}$$

this is a lower bound to $\rho(A)$, again without additional hypotheses on A. Indeed, note that whenever $\beta \boldsymbol{x} \leq A\boldsymbol{x}$ with $\boldsymbol{x} \geq 0$, for every $\delta > 0$ if $\boldsymbol{w}$ is a positive eigenvector of $A + \delta \mathbf{1}\mathbf{1}^*$ we have

$$\beta \boldsymbol{x} \leq A\boldsymbol{x} \leq (A + \delta \mathbf{1}\mathbf{1}^*)\boldsymbol{x} \leq (A + \delta \mathbf{1}\mathbf{1}^*)\|\boldsymbol{x}\|_\infty^{\boldsymbol{w}} \boldsymbol{w} = \rho(A + \delta \mathbf{1}\mathbf{1}^*)\|\boldsymbol{x}\|_\infty^{\boldsymbol{w}} \boldsymbol{w}.$$

The last inequality implies $\beta \leq \rho(A + \delta \mathbf{1}\mathbf{1}^*)$ by Proposition 1.6, and since the inequality is true for every δ it is true by continuity also for $\delta = 0$.

These properties suggest that in practice iteration should be stopped if $s^{(t)}$ goes below the minimum representable floating-point number: in this case, either $\sigma < \rho(A)$, or the finite precision at our disposal is not sufficient to compute a suitable vector because we cannot represent correctly a transient behavior of the powers of A.

If instead the minimum (5) becomes larger than σ, we can safely stop: unfortunately, the latter event cannot be guaranteed to happen when $\sigma < \rho(A)$ without additional hypotheses on A (e.g., irreducibility): for instance, if A has a null row the minimum (5) will always be equal to zero.

Of course, there ain't no such thing as a free lunch. The termination of the process above is guaranteed if $\sigma > \rho(A)$, but we have no indication of how many step will be required. Moreover, in principle some of the coordinates of the suitable vector could be so small to make Theorem 1 unusable. For σ close to $\rho(A)$ convergence can be very slow, as it is related to the convergence of Collatz's lower and upper bounds for the dominant eigenvalue.

Nonetheless, albeit all of the above must happen in pathological cases, we show on a few examples that, actually, in real-world cases computing a σ-suitable vector is not difficult.

We remark that in principle any dyadic product $\boldsymbol{u}\boldsymbol{v}^*$ such that $A + \boldsymbol{u}\boldsymbol{v}^*$ is irreducible will do the job in the proof of Theorem 2. There might be choices (possibly depending on A) for which the computation above terminates more quickly.

6 Examples

6.1 Bounding the error of $(I - A)\boldsymbol{x} = \boldsymbol{b}$

If A is nonnegative matrix with $\rho(A) < 1$, then $I - A$ is invertible and the problem $(I - A)\boldsymbol{x} = \boldsymbol{b}$ has a unique solution, and in the limit we have convergence geometric in $\rho(A)$. However, if we choose a $1 > \sigma > \rho(A)$ (say, $\sigma = (1 + \rho(A))/2$) and a σ-suitable vector $\boldsymbol{w}$, the bounds of Theorem 1 will be valid, so we will be able to control the error in $\boldsymbol{w}$-norm.

6.2 Katz's index

Let M be a nonnegative matrix (in the standard formulation, the adjacency matrix of a graph). Then, given $\alpha < 1/\rho(M)$ Katz's index is defined by

$$\boldsymbol{k}^* = \boldsymbol{v}^*\big(1-\alpha M\big)^{-1} = \boldsymbol{v}^* \sum_{k\geq 0} \alpha^k M^k,$$

where $\boldsymbol{v}$ is a *preference vector*, which is just $\mathbf{1}$ in Katz's original definition [Kat53].[6].

If we want to apply Theorem 1, we must choose a $\sigma > \rho(A)$ and a σ-suitable vector $\boldsymbol{w}$ for A. The vector can then be used to accurately estimate the computation of Katz's index for all $\alpha < 1/\sigma$. This property is particularly useful, as it is common to estimate the index for different values of α, and to that purpose it is sufficient to compute once for all a σ-suitable vector for a σ chosen sufficiently close to $\rho(A)$.

6.3 PageRank

The case of PageRank is similar to Katz's index. We have

$$\boldsymbol{r}^* = (1-\alpha)\boldsymbol{v}^*(1-\alpha P)^{-1} = (1-\alpha)\boldsymbol{v}^* \sum_{k=0}^{\infty} \alpha^k P^k,$$

where $\boldsymbol{v}$ is the preference vector, and $P = \bar{G} + \boldsymbol{d}\boldsymbol{u}^*$ is a stochastic matrix; $\bar{G}$ is the adjacency matrix of a graph G, normalized so that each nonnull row adds to one, $\boldsymbol{d}$ is the characteristic vector of *dangling nodes* (nodes without outlinks, i.e., null rows), and $\boldsymbol{u}$ is the dangling-node distribution, used to redistribute the rank lost through dangling nodes. It is common to use a uniform $\boldsymbol{u}$, but most often $\boldsymbol{u} = \boldsymbol{v}$, and in that case we speak of *strongly preferential* PageRank [BSV09].

We remark that in the latter case it is well known that the *pseudorank*

$$\boldsymbol{p}^* = (1-\alpha)\boldsymbol{v}^* \sum_{k=0}^{\infty} \alpha^k \bar{G}^k$$

satisfies

$$\boldsymbol{r} = \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|_1}.$$

That is, PageRank and the pseudorank are parallel vectors. This is relevant for the computation of several strongly preferential PageRank vectors: just compute a σ-suitable vector for $\bar{G}$ (rather than one for each $\bar{G} + \boldsymbol{d}\boldsymbol{v}^*$, depending on $\boldsymbol{v}$), and compute pseudoranks instead of ranks.

The case of PageRank is however less interesting because, as David Gleich made the author note, assuming the notation of Section 2 and $\omega = 1$

$$\begin{aligned}
\big(1-\alpha P^T\big)\boldsymbol{x}^{(t+1)} - (1-\alpha)\boldsymbol{v} &= \big(D - L^{(t)} - R^{(t)}\big)\boldsymbol{x}^{(t+1)} - (1-\alpha)\boldsymbol{v} \\
&= \big(D - L^{(t)}\big)\boldsymbol{x}^{(t+1)} - R^{(t)}\boldsymbol{x}^{(t+1)} - (1-\alpha)\boldsymbol{v} \\
&= R^{(t)}\boldsymbol{x}^{(t)} + (1-\alpha)\boldsymbol{v} - R^{(t)}\boldsymbol{x}^{(t+1)} - (1-\alpha)\boldsymbol{v} \\
&= R^{(t)}(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t+1)}).
\end{aligned}$$

[6] We must note that actually Katz's index is $\boldsymbol{v}^*\big(1-\alpha M\big)^{-1}M$. This additional multiplication by M is somewhat common in the literature; it is probably a case of *horror vacui*.

	Wikipedia	.uk
nodes	4 206 785	105 896 555
arcs	101 355 853	3 738 733 648
avg. degree	24.093	35.306
giant component	89.00%	64.76%
harmonic diameter	5.24	22.78
dominant eigenvalue	191.11	5676.63

Table 1: Basic structural data about our two datasets.

Since $\left\|R^{(t)}\right\|_1 \leq \alpha$, we can ℓ_1-bound the residual

$$\left\|\left(1-\alpha P^T\right)^{-1}\right\|_1 = \left\|\sum_{k=0}^{\infty} \alpha^k \left(P^T\right)^k\right\|_1 \leq \frac{1}{1-\alpha}$$

we conclude that

$$\left\|\bar{\boldsymbol{x}} - \boldsymbol{x}^{(t+1)}\right\|_1 \leq \frac{\alpha}{1-\alpha}\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right\|_1.$$

It is thus possible, albeit wasteful, to bound the supremum norm of the error using its ℓ_1 norm.

7 Experiments

In this section we discuss some computational experiments involving the computation of PageRank and Katz's index on real-world graphs. We focus on a snapshot of the English version of Wikipedia taken in 2013 (about four million nodes and one hundred million arcs) and a snapshot of the `.uk` web domain taken in may 2007 (about one hundred million nodes and almost four billion arcs).[7] These two graphs have some structural differences, which we highlight in Table 1.

We applied the procedure described in Section 5 to the system associated with PageRank and Katz's index, with $\sigma \in \{\, 1/(1-2^{-i}) \mid 1 \leq i \leq 10 \,\}$ for PageRank and $\sigma \in \{\, \lambda/(1-2^{-i}) \mid 1 \leq i \leq 10 \,\}$ for Katz's index.

In Figure 1 we report the number of iterations that are necessary to compute the i-th suitable vector. The two datasets show the same behavior in the case of PageRank—an exponential increase in the number of iterations as we get exponentially closer to the limit value. The case of Katz is more varied: whereas Wikipedia has a significant growth in the number of iterations (but clearly slower than the PageRank case), `.uk` has a minimal variation across the range (from 2 to 6).

In Figure 2 we draw the (exponentially binned) distribution of values of suitable vectors for a choice of four equispaced values of i. The vectors are normalized in ℓ_∞ norm, that is, the largest value is one.

The shape of the distribution depends both on the graph and on the type of centrality computed, but two features are constant: first, as we approach λ the distribution

[7] Both datasets are publicly available at the site of the Laboratory for Web Algorithmics (`http://law.di.unimi.it/`) under the identifiers `enwiki-2013` and `uk-2007-05`.

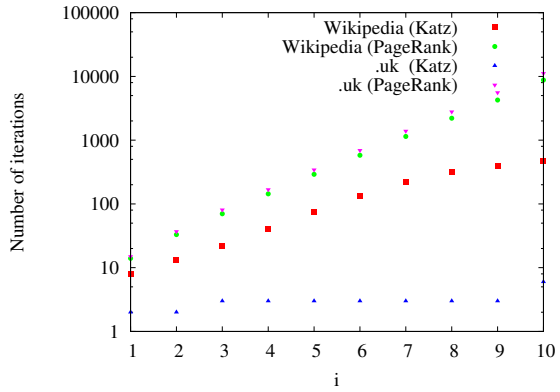


Figure 1: Number of iterations that are necessary to compute a $\lambda/(1-2^{-i})$-suitable vector.

contains smaller and smaller values; second, the smallest value in the PageRank case is several orders of magnitude smaller.

Smaller values imply a larger $\boldsymbol{w}$-norm: indeed, one can think of the elements of an ℓ_∞-normalized suitable vector $\boldsymbol{w}$ as weights that "slow down" the convergence of problematic nodes by inflating their raw error. The intuition we gather from the distribution of values is that bounding the convergence of PageRank is more difficult.

8 Conclusions

We have presented results that make it possible to bound the supremum norm of the absolute error of SOR iterations an M-matrix $sI - A$ even when estimating $\left\|(sI - A)^{-1}\right\|_\infty$ is not feasible. Rather than relying on additional hypotheses such as positive definiteness, irreducibility and so on, our results suggest to compute first a σ-*suitable* positive vector $\boldsymbol{w}$ with the property that SOR iterations converge geometrically in $\boldsymbol{w}$-norm by a computable factor.

While we cannot bound without additional hypotheses the resources (number of iterations and precision) that are necessary to compute $\boldsymbol{w}$, in practice the computation is not difficult, and given an M-matrix $sI - A$ the associated σ-suitable $\boldsymbol{w}$ can be used for all $s > \sigma$.

References

[BP94] Abraham Berman and Robert J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Classics in Applied Mathematics. SIAM, 1994.

[BPSV08] Paolo Boldi, Roberto Posenato, Massimo Santini, and Sebastiano Vigna. Traps and pitfalls of topic-biased PageRank. In William Aiello, Andrei Broder, Jeannette Janssen, and Evangelos Milios, editors, *WAW*

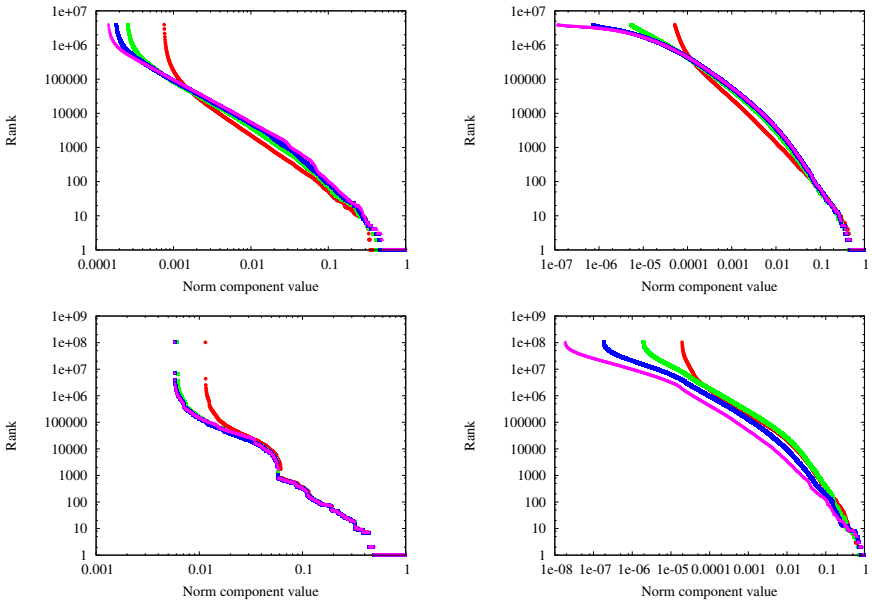


Figure 2: Exponentially binned frequency plots of the values of $\lambda/(1-2^{-i})$-suitable vectors, $i = 1$, 4, 7 and 10.

2006. Fourth Workshop on Algorithms and Models for the Web-Graph, volume 4936 of *Lecture Notes in Computer Science*, pages 107–116. Springer–Verlag, 2008.

[BSV09] Paolo Boldi, Massimo Santini, and Sebastiano Vigna. PageRank: Functional dependencies. *ACM Trans. Inf. Sys.*, 27(4):1–23, 2009.

[BT89] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, Englewood Cliffs NJ, 1989.

[Col42] Lothar Collatz. Einschließungssatz für die charakteristischen Zahlen von Matrizen. *Mathematische Zeitschrift*, 48(1):221–226, 1942.

[GK11] Chen Greif and David Kurokawa. A note on the convergence of SOR for the PageRank problem. *SIAM J. Sci. Computing*, 33(6):3201–3209, 2011.

[Kat53] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[Ken45] Maurice G. Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.

[PBMW98] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report SIDL-WP-1999-0120, Stanford Digital Library Technologies Project, Stanford University, 1998.

[Vig09] Sebastiano Vigna. Spectral ranking. *CoRR*, abs/0912.0238, 2009.

An experimental exploration of Marsaglia's `xorshift` generators, scrambled

Sebastiano Vigna, Università degli Studi di Milano, Italy

Marsaglia proposed recently `xorshift` generators as a class of very fast, good-quality pseudorandom number generators. Subsequent analysis by Panneton and L'Ecuyer has lowered the expectations raised by Marsaglia's paper, showing several weaknesses of such generators, verified experimentally using the TestU01 suite. Nonetheless, many of the weaknesses of `xorshift` generators fade away if their result is scrambled by a non-linear operation (as originally suggested by Marsaglia). In this paper we explore the space of possible generators obtained by multiplying the result of a `xorshift` generator by a suitable constant. We sample generators at 100 equispaced points of their state space and obtain detailed statistics that lead us to choices of parameters that improve on the current ones. We then explore for the first time the space of high-dimensional `xorshift` generators, following another suggestion in Marsaglia's paper, finding choices of parameters providing periods of length $2^{1024} - 1$ and $2^{4096} - 1$. The resulting generators are of extremely high quality, faster than current similar alternatives, and generate long-period sequences passing strong statistical tests using only eight logical operations, one addition and one multiplication by a constant.

Categories and Subject Descriptors: G.3 [**PROBABILITY AND STATISTICS**]: Random number generation; G.3 [**PROBABILITY AND STATISTICS**]: Experimental design

General Terms: Algorithms, Experimentation, Measurement

Additional Key Words and Phrases: Pseudorandom number generators

1. INTRODUCTION

`xorshift` generators are a simple class of pseudorandom number generators introduced by Marsaglia [2003]. In Marsaglia's view, their main feature is speed: in particular, a `xorshift` generator with a 64-bit state space generates a new 64-bit value using just three 64-bit shifts and three 64-bit xors (i.e., exclusive ors), thus making it possible to generate hundreds of millions of values per second.

Subsequent analysis by Brent [2004] showed that the bits generated by `xorshift` generators are equivalent to certain *linear feedback shift registers*. Panneton and L'Ecuyer [2005] analyzed moreover in detail the generators using the TestU01 suite [L'Ecuyer and Simard 2007], finding weaknesses and proposing an increase in the number of shifts, or combination with another generator, to improve quality.

In the first part of this paper we explore experimentally the space of `xorshift` generators with 64-bit state space using statistical test suites. We sample generators at 100 equispaced points of their state space, to easily identify spurious failures. There are 2200 possible full-period `xorshift` generators, due to 275 possible values for its three shift parameters, and eight possible algorithms (see Figure 1); Marsaglia proposes some choice of parameters, that, as we will see, and as already reported by Panneton and L'Ecuyer [2005], are not particularly good. We report results that are actually worse than those of Panneton and L'Ecuyer as we use the entire 64-bit output of the generators. While we can suggest some good parameter choices, the result remains poor.

Thus, we turn to the idea of scrambling the result of a `xorshift` generator using a multiplication, as it is typical, for instance, in the construction of practical hash function due to the resulting *avalanching* behavior (bits of the result depend on several bits of the

This work is supported the EU-FET grant NADINE (GA 288956).
Author's addresses: Sebastiano Vigna, Dipartimento di Informatica, Università degli Studi di Milano, via Comelico 39, 20135 Milano MI, Italy.

	C code	
A_0	`x ^= x << a; x ^= x >> b; x ^= x << c;`	$\mathbf{X}_1$
A_1	`x ^= x >> a; x ^= x << b; x ^= x >> c;`	$\mathbf{X}_3$
A_2	`x ^= x << c; x ^= x >> b; x ^= x << a;`	$\mathbf{X}_2$
A_3	`x ^= x >> c; x ^= x << b; x ^= x >> a;`	$\mathbf{X}_4$
A_4	`x ^= x << a; x ^= x << c; x ^= x >> b;`	$\mathbf{X}_5$
A_5	`x ^= x >> a; x ^= x >> c; x ^= x << b;`	$\mathbf{X}_6$
A_6	`x ^= x >> b; x ^= x << a; x ^= x << c;`	$\mathbf{X}_7$
A_7	`x ^= x << b; x ^= x >> a; x ^= x >> c;`	$\mathbf{X}_8$

Fig. 1. The eight possible `xorshift64` algorithms. The list is actually derived from Panneton and L'Ecuyer [2005], as they correctly remarked that two of the eight algorithms proposed by Marsaglia were redundant, whereas two (A_6 and A_7) were missing. On the right side we report the name of the linear transformation associated to the algorithm as denoted by Panneton and L'Ecuyer [2005]. With our numbering, algorithms A_{2i} and A_{2i+1} are conjugate by reversal. Note that contiguous shifts in the same direction can be exchanged without affecting the resulting algorithm. We normalized such contiguous shifts so that their letters are lexicographically sorted.

input). This can be seen as the composition of a `xorshift` generator with a *multiplicative linear congruential generator*, and is actually suggested in passing in Marsaglia's paper. The third edition of the classic "Numerical Recipes" [Press et al. 2007], indeed, proposes this construction for a basic, all-purpose generator. Since a lot of knowledge has been gathered in the last 50 years on multiplicative constants that give have *good spectral properties*, we use multipliers taken from [L'Ecuyer 1999].

From the wealth of data so obtained we derive generators with better statistical properties than those suggested in "Numerical Recipes". We also investigate several interesting correlations, such as those between the weight of the characteristic polynomial and failures in statistical test suites.

In the last part of the paper, we follow the suggestion about high-dimensional generators contained in Marsaglia's paper, and compute for the first time several choices of parameters that provide full-period `xorshift` generators with a state space of 1024 and 4096 bits. Once again, we propose generators that use a multiplication to scramble the result.

At the end of the paper, we apply our methodology to a number of popular non-cryptographic generators, and we discover that our high-dimensional generators are actually faster and of higher or equivalent statistical quality, as assessed by statistical test suites, than the alternatives.

The software used to perform the experiments described in this paper is distributed by the author under the GNU General Public License. Moreover, all files generated during the experiments are available from the author. They contain a large amount of data that could be further analyzed (e.g., by studying the distribution of p-values over the seeds). We leave this issue open for further work.

2. AN INTRODUCTION TO `xorshift` GENERATORS

The basic idea of `xorshift` generators is that the state space is modified by applying repeatedly a shift and an exclusive-or (xor) operation. In this paper we consider 64-bit shifts and state spaces of 2^n bits, with $n \geq 6$. We usually append n to the name of a family of generators when we need to restrict the discussion to a specific state-space size.

For `xorshift64` generators Marsaglia suggests a number of possible combination of shifts, shown in Figure 1. Not all choices of parameters give a full $(2^{64} - 1)$ period: there are 275 suitable choices of a, b and c and eight variants, totalling 2200 generators.

In linear-algebra terms, if L is the 64×64 matrix on $\mathbf{Z}/2\mathbf{Z}$ that effects a left shift of one position on a binary vector (i.e., L is all zeroes except for ones on the principal subdiagonal) and if R is the right-shift matrix (the transpose of L), each left/right shift/xor can be

described as a linear multiplication by $\big(1 + L^s\big)$ or $\big(1 + R^s\big)$, respectively, where s is the amount of shifting.[1] For instance, algorithm A_0 of Figure 1 is equivalent to the $\mathbf{Z}/2\mathbf{Z}$-linear transformation

$$\mathbf{X}_1 = \big(1 + L^a\big)\big(1 + R^b\big)\big(1 + L^c\big).$$

It is useful to associate with a linear transformation M its *characteristic polynomial*

$$P(x) = \det(M - x).$$

The associated generator has maximum-length period if and only if $P(x)$ is primitive over $\mathbf{Z}/2\mathbf{Z}$. This happens if $P(x)$ is irreducible and if z has maximum period in the ring of polynomial over $\mathbf{Z}/2\mathbf{Z}$ modulo $P(x)$, that is, if the powers z, z^2, ..., z^{2^n-1} are distinct modulo $P(x)$. Finally, to check this condition is sufficient to check that

$$x^{(2^n-1)/p} \neq 1 \mod P(x)$$

for every prime p dividing 2^{n-1} [Lidl and Niederreiter 1994].

The *weight* of $P(x)$ is the number of terms in $P(x)$, that is, the number of nonzero coefficients. It is considered a good property for generators of this kind that the weight is close to $n/2$, that is, that the polynomial is neither too sparse nor too dense [Compagner 1991]. For this reason, if a generator has characteristic polynomial $P(x)$ of degree d with weight W we define the *weight score* of the generator as

$$|W - d/2|,$$

so a low weight score is better.

Note that the family of algorithms of Figure 1 is intended to generate *64-bit values*. This means that the entire output of the algorithm should be used when performing tests. We will see that this has not always been the case in previous literature.

3. SETTING UP THE EXPERIMENTS

In this paper we want explore experimentally the space of a number of `xorshift`-based generators. Our purpose is to identify variants with full period which have particularly good statistical properties, and test whether claims about good parameters made in the previous literature are confirmed.

The basic idea is that of *sampling* the generators by executing a battery of tests starting with 100 different seeds that are equispaced in the state space. For instance, for a 64-bit state space we use the seeds $1 + i\lfloor 2^{64}/100 \rfloor$, $0 \leq i < 100$. The tests produce a number of statistics, and we decided to use the number of failed tests as a measure of low quality. Running multiple tests makes it easy to rule out spurious failures, as suggested also by Rukhin et al. [2001] in the context of cryptographic applications.

We use two tools to perform our tests. The first and most important is TestU01, a test suite developed by L'Ecuyer and Simard [2007] that contains several tests oriented towards the generation of uniform real numbers in $[0\,..\,1)$. We also perform tests using Dieharder, a suite of tests developed by Brown [2013], both as a sanity check and to compare the power of the two suites. Dieharder contains all original tests from Marsaglia's Diehard, plus many more other tests. The suite is more oriented towards the effective values assumed by each bit (e.g., it computes more statistics on subsequences). We refer frequently to the specific type of tests failed: the reader can refer to the TestU01 and Dieharder documentation for more information.

We consider a test failed if its p-value is outside of the interval $[0.001\,..\,0.999]$. This is the interval outside which TestU01 reports a failure by default. Sometimes a much stricter

[1] A more detailed study of the linear algebra behind `xorshift` generators can be found in [Marsaglia 2003; Panneton and L'Ecuyer 2005].

threshold is used (For instance, L'Ecuyer and Simard [2007] use $[10^{-10} \,..\, 1 - 10^{-10}]$ when applying TestU01 to a variety of generators), but since we are going to repeat the test 100 times we can use relatively weak p-values: spurious failures will appear rarely, and we can catch borderline cases (e.g., tests failing on 50% of the seeds) that give us useful information. We call *systematic* a failure that happens for all seeds.

We remark that our choice (counting the number of failures) is somewhat rough; for example, we consider the same failure a p-value very close to 0 and a p-value just below 0.001. Indeed, other, more sophisticated methods might be used to aggreate the result of our samples: combining p-values, for instance, or computing a p-value of p-values [Rukhin et al. 2001]. However, our choice is very easy to interpret, and multiple samples partially compensate this problem (spurious failures will appear in few samples).

Of course, the number of experiments is very large—in fact, our experiments were carried using hundreds of cores in parallel and, overall, they add up to more than a century of computational time. Our strategy is to apply a very fast test to all generators and seeds, in the hope of isolating a small group of generators that behave significantly better. Stronger tests can then be applied to this subset. The same strategy has been followed by Panneton [2004] in the experimental study of `xorshift` generators contained in his Ph.D. thesis.

TestU01 offers three different predefined batteries of tests (SmallCrush, Crush and BigCrush) with increasing computational cost and increased difficulty. Unfortunately, Dieharder does not provide such a segmentation.

Note that Dieharder has a concept of "weak" success and a concept of "failure", depending on the p-value of the test, and we used command-line options to align its behavior with that of TestU01: a p-value outside of the range $[0.001 \,..\, 0.999]$ is a failure. Moreover, we disabled the initial timing tests so that exactly the same stream of 64-bit numbers is fed to the two test suites.

In both cases we implemented our own `xorshift` generator. Some care is needed in this phase, as both TestU01 and Dieharder are inherently 32-bit test suites: since we want to test `xorshift` as a *64-bit* generator, it is important that all bits produced are actually fed into the test. For this reason, we implemented the generation of a uniform real value in $[0 \,..\, 1)$ by dividing the output of the generator by 2^{64}, but we implemented the generation of uniform 32-bit integer values by *returning first the lower and then the upper 32 bits of each 64-bit generated value.*[2]

An important consequence of this choice is that some of the bits are actually not used at all. When analyzing pseudorandom real numbers in the unit interval, there is an unavoidable bias towards high bits, as they are more significant. The very lowest bits have lesser importance and will be in any case perturbed by numerical errors. However, in our case the lowest eleven bits returned by the generator *are not used at all* due to the fact that the mantissa of a 64-bit floating-point number is formbed by 53 bits only. For this reason, we will consistenly run our tests both on a generator and on its reverse.[3]

We remark that in this paper we do not pursue the search for *equidistribution*—the property that all tuples of consecutive values, seen as vectors in the unit cube, are evenly distributed, as done, for instance, by Panneton and L'Ecuyer [2005]. Brent [2010] has already argued in detail that for long-period generators equidistribution is not particularly desirable, as it is a property of the whole sequence produced by the generator, and in the case of a long-period generator only a minuscule fraction of the sequence can be actually used. Moreover, equidistribution is currently impossible to evaluate exactly for long-period non-linear generators, and it is known to be biased towards the high bits [L'Ecuyer and Panneton 2005]: for instance, the `WELL1024a` generator has been designed to be *maximally equidistributed* [Panneton et al. 2006], and indeed it has measure of equidistribution $\Delta_1 = 0$,

[2] If a real value is generated when the upper 32 bits of the last value are available, they are simply discarded.
[3] That is, on the generator obtained by reversing the order of the 64 bits returned.

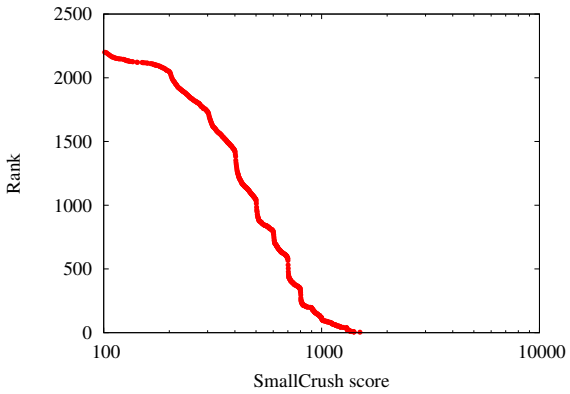


Fig. 2. Score-rank plot of the distribution of SmallCrush scores for the 2200 possible full-period `xorshift64` generators.

but the generator obtained by reversing its bits has $\Delta_1 = 366$: a quite counterintuitive result, as in general we expect all bits to be equally important.

Another major problem with equidistribution is its "on/off" nature: it provides no measure of *how much* a generator is equistributed. This leads to the following pathological behavior: if we take a maximally equidistributed sequence, no matter how long, and we flip the most significant bit of a single element of the sequence, the new sequence will have the *worst possible* Δ_1. For instance, by flipping the most significant bit of a single chosen value out of the output of `WELL1024a` we can turn its equidistribution measure to $\Delta_1 = 4143$. But for any statistical or practical purpose the two sequences are indistinguishable—we are modifying one bit out of $2^5(2^{1024} - 1)$.

We note, however, that since multiplication by an invertible constant induces a permutation of the space of 64-bit values (and thus of t-tuples of such values), the choice of multiplication has the advantage, with respect to other scrambling techniques, of preserving some of the equidistribution properties of the underlying generator; more details will be given in the rest of the paper.

4. RESULTS FOR `xorshift64` GENERATORS

In this section we report the results obtained for the 2200 possible variants of the `xorshift64` generator with full period.

First of all, *all* generators fail at all seeds the MatrixRank test from TestU01's SmallCrush suite. This is somewhat to be expected, as each new value is obtained by applying a linear transformation to the previous one. However, Panneton and L'Ecuyer [2005] report that *half* of the generators fail this test. Unfortunately, the authors do not detail the conditions (seed, implementation of the algorithm, etc.) of the experiments they performed, but TestU01 is available and contains an implementation of the `xorshift64` family. A simple analysis of the code shows that the authors have chosen to use only 32 of the 64 generated bits as output bits, in practice applying a kind of *decimation* to the output of the generator. As we explained in Section 3, we actually feed *all* bits output by the generator to the test suite, which explains why we report a significantly worse performance.

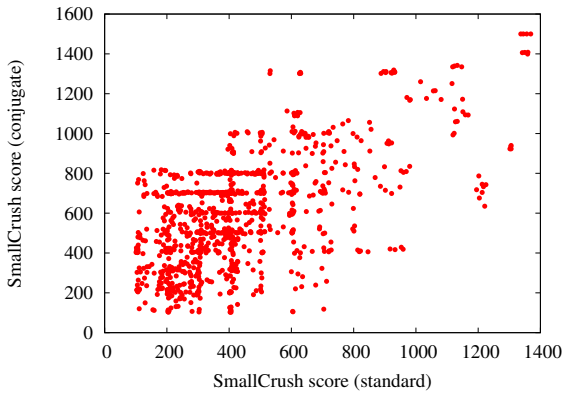


Fig. 3. Scatter plot of the SmallCrush score of conjugate generators.

A score-rank plot of the SmallCrush scores for all generators is shown in Figure 2. The plot associates with abscissa x the number of generators with x or more failures.[4] We observe immediately that *there is a wide range of quality among the generators examined*. Indeed, inspecting more closely we would see that the best generator ($A_4(14, 13, 17)$, score 101) fails MatrixRank systematically and just Collision on one sample. The worst generators (e.g., $A_5(1, 1, 55)$, score 1500), instead, fails systematically *all* SmallCrush tests. The "bumps" in the plot corresponds to new tests failed systematically.

Figure 3 shows instead a scatter plot associating conjugate algorithms, which have just their bits reversed. While there is some weak correlation, we can see the bias towards high bits at work. Some generators fail systematically just a few tests while their reverse fail more than a dozen.

As we already remarked, to avoid high-bits bias we score each conjugate pair jointly, adding up the number of failures in SmallCrush: Table I reports the best four generators, which are the only ones failing systematically just the MatrixRank test. Any other choice fails more than half of the times the BirthdaySpacings test (data not shown here), and all generators with a rank lower than those shown in Table I fail systematically at least one test besides MatrixRank.

The table reports also results for the generator $A_0(13, 7, 17)$ suggested by Marsaglia in his original paper, claiming that it "will provide an excellent period $2^{64} - 1$ RNG, [...] but any of the above 2200 choices is likely to do as well". Clearly, this is not the case: $A_0(13, 7, 17)/A_1(13, 7, 17)$ ranks 655 in the combined SmallCrush ranking and fails systematically several tests. Unfortunately, since this choice of parameters appeared in the original paper, other researchers have used it as well: this is what happens, for example, in the comparison table assembled by L'Ecuyer and Simard [2007] using TestU01 .

We remark that the triples suggested in "Numerical Recipes" are just of average quality when measured using TestU01. The authors suggest as best algorithm $A_3(4, 35, 21)$ (with its conjugate $A_2(4, 35, 21)$, as they notice that it is important to check conjugates), which however with score 389 ranks only 29 in our combined classification. The best result for the triples suggested at page 347 are those for $A_2(11, 29, 14)/A_3(11, 29, 14)$ (score 314, rank 9).

[4] Score-rank plots are the numerosity-based discrete analogous of the complementary cumulative distribution function of scores. They give a much clearer picture than frequency dot plots when the data points are scattered and highly variable.

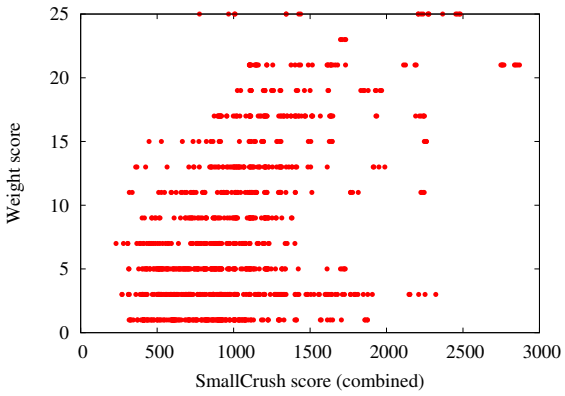


Fig. 4. Scatter plot of the combined SmallCrush score of conjugate xorshift64 generators versus their weight score.

Finally, Figure 4 shows a scatter plot associating the combined (standard plus reverse) score of a generator with its weight score. While there is a very mild correlation (the very lower left and upper right corners are empty), it is definitely not the case that the SmallCrush score of a xorshift64 generator is strictly dependent on its weight score.

SANITY CHECK 1. *Is the result of our experiments dependent on our seed choice? To answer this question, we repeated our experiments on* xorshift64 *generators with SmallCrush on a different set of seeds, namely the integers in the interval* $[1\,..\,100]$. *The scatter plot in Figure 5 and Kendall's*[5] $\tau \approx 0.98$ *between the two rankings we obtain make rather clear that this is not the case. In particular, the four best conjugate pairs in Table I are the same with both seeds.*

To gather more information, we ran the full BigCrush suite and Dieharder on our four best generators, on Marsaglia's choice and on the best choice from "Numerical Recipes": the results are given in Table II and III. Even the four best generators fail now systematically the BirthdaySpacings, MatrixRank and LinearComp tests. The first two generators, however, turn out to perform better than other two. We also notice that BigCrush draws a much thicker line between our four best generators and the other ones, which now fail several more tests. Not surprisingly, Dieharder cannot really separate our four best generators from $A_2(4, 35, 21)/A_3(4, 35, 21)$.

4.1. Equidistribution

As we already discussed in the introduction, in this paper we do not pursue equidistribution of a generator. It is nonetheless interesting to compare the ranking provided by equidistribution properties and that provided by statistical tests. Note that a xorshift64 generator is at least 1-dimensionally equidistributed, that is, every 64-bit value appears exactly once except for zero. We refer to the already quoted paper by Panneton and L'Ecuyer [2005] for a detailed description of the equidistribution statistics Δ_1, the *sum of dimension gaps*: a lower value is better. A *maximally distributed* generator has $\Delta_1 = 0$, and we will refer to Δ_1

[5]We are using the generalization allowing ties and defined in [Kendall 1945], which is often called τ_b, reserving τ for the original coefficient [Kendall 1938]. But this distinction is pointless, as in [Kendall 1938] τ is defined only for rankings with no ties, and the definition given in [Kendall 1945] reduces exactly to the original definition if there are no ties.

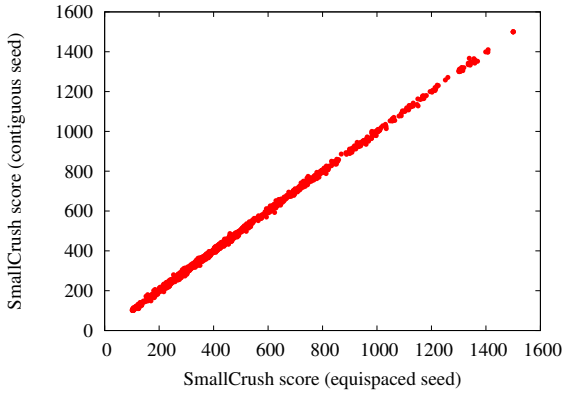


Fig. 5. Scatter plot of the scores obtained on each `xorshift64` generator using the seed $1 + i\lfloor 2^{64}/100 \rfloor$ or the seed $1 + i$, $0 \le i < 100$.

Table I. Best four `xorshift64` generators following SmallCrush.

Algorithm	Failures	Conjugate	Failures	Overall	W
$A_2(11, 31, 18)$	111	$A_3(11, 31, 18)$	120	231	25
$A_2(8, 29, 19)$	155	$A_3(8, 29, 19)$	115	270	35
$A_0(8, 29, 19)$	159	$A_1(8, 29, 19)$	112	271	35
$A_0(11, 31, 18)$	130	$A_1(11, 31, 18)$	150	280	25
$A_2(4, 35, 21)$	209	$A_3(4, 35, 21)$	180	389	25
$A_0(13, 7, 17)$	276	$A_1(13, 7, 17)$	802	1078	25

Note: The only systematic failure is on the MatrixRank test. All other generators have an overall number of failures greater than 300, and fail systematically at least one test besides MatrixRank. $A_0(13, 7, 17)$ is the generator suggested in Marsaglia's original paper, and ranks 655 on 1100 conjugate pairs. $A_3(4, 35, 21)$ is suggested as the best generator in this class in "Numerical Recipes" [Press et al. 2007], and ranks 29.

Table II. The generators of Table I tested with BigCrush.

Algorithm	Failures	Conjugate	Failures	Overall	Systematic
$A_2(11, 31, 18)$	762	$A_3(11, 31, 18)$	750	1512	
$A_2(8, 29, 19)$	747	$A_3(8, 29, 19)$	780	1527	A
$A_0(8, 29, 19)$	749	$A_1(8, 29, 19)$	884	1633	
$A_0(11, 31, 18)$	748	$A_1(11, 31, 18)$	926	1674	
$A_2(4, 35, 21)$	961	$A_3(4, 35, 21)$	1444	2405	$A \cup B$
$A_0(13, 7, 17)$	1049	$A_1(13, 7, 17)$	5454	6503	$A \cup C$

Note: A = {BirthdaySpacings, LinearComp, MatrixRank}. B = {RandomWalk1C, ClosePairsmNP, ClosePairsmNP1, ClosePairsmNP2S}. C = {RandomWalk1H, RandomWalk1J, RandomWalk1M, ClosePairsNJumps, ClosePairsNP, ClosePairsmNP2, ClosePairsmNP2S, CollisionOver, MaxOft, MaxOftAD, Permutation, Run, SampleMean, SampleProd, SerialOver, SumCollector}. This table should be compared with Table I and with the experimental results by L'Ecuyer and Simard [2007].

Table III. The generators of Table I tested with Dieharder.

Algorithm	Failures	Conjugate	Failures	Overall	Systematic
$A_2(11,31,18)$	182	$A_3(11,31,18)$	162	344	dab_monobit2
$A_2(8,29,19)$	179	$A_3(8,29,19)$	181	360	
$A_0(8,29,19)$	176	$A_1(8,29,19)$	182	358	
$A_0(11,31,18)$	181	$A_1(11,31,18)$	186	367	
$A_2(4,35,21)$	189	$A_3(4,35,21)$	187	376	dab_monobit2
$A_0(13,7,17)$	183	$A_1(13,7,17)$	1352	1535	A

Note: A = {dab_filltree, dab_monobit2, diehard_2dsphere, diehard_3dsphere, diehard_operm5, diehard_parking_lot, diehard_squeeze, rgb_minimum_distance, rgb_permutations}. This table should be compared with Table II.

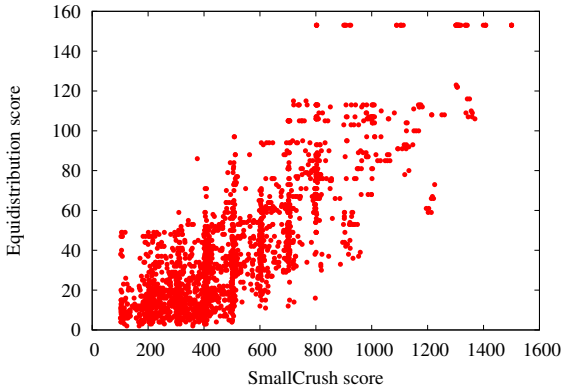


Fig. 6. Scatter plot of the SmallCrush score versus the equidistribution score of `xorshift64` generators.

as to the *equidistribution score*. We computed the equidistribution score for all generators using the implementation of Harase's algorithm [Harase 2011] contained in the `MTToolBox` package from Saito [2013].

SANITY CHECK 2. *It is very easy to introduce hard-to-detect bugs in this kind of computations. However, the Ph.D. thesis of Panneton [2004] reports* Δ_1 *(therein called* V*) for all full-period* `xorshift64` *generators, and we checked that the results are the same. We will use* `MTToolBox` *for other computations, which explains the usefulness of recomputing and checking these values.*

Figure 6 shows that there is some correlation between the SmallCrush score and the equidistribution score of `xorshift64` generators. Nonetheless, in the "interesting" region (the lowest left corner) correlation is not very good—for instance, the generator $A_6(10,7,33)$, which has equidistribution score 2 (the best), fails systematically three types of tests, and ranks 757 in combination with its conjugate: it's actually one of the *worst* generators. Its combined BigCrush score is 6691—even worse than the generator $A_0(13,7,17)$ suggested by Marsaglia.

The explanation is simple: similarly to SmallCrush score, equidistribution has high-bits bias, and a quite strong one [L'Ecuyer and Panneton 2005]. Indeed, Figure 7 reports a scatter plot of the equistribution score of conjugate (i.e., reverse) generators, and the bias towards the high bits is very visibile from the lack of correlation. There are apparently good generators whose reverse is actually worst: for instance, $A_2(12,1,31)$ has score 11 but its conjugate $A_3(12,1,31)$ has score 153—the highest value in the set, whereas $A_0(11,5,32)$

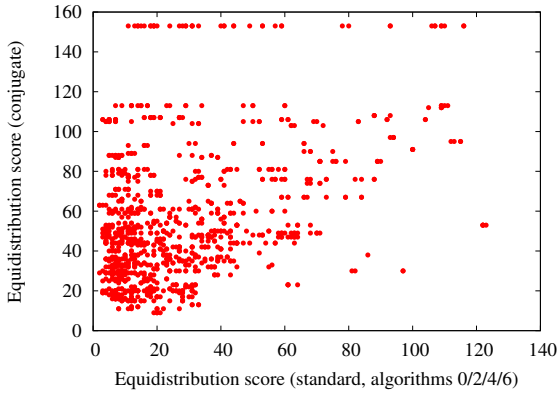


Fig. 7. Scatter plot of the equidistribution score of conjugate `xorshift64` generators.

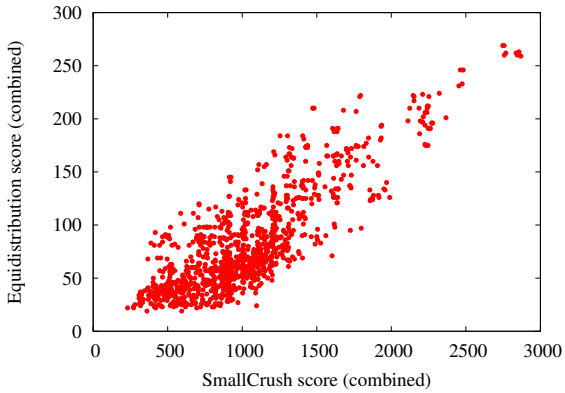


Fig. 8. Scatter plot of the combined SmallCrush score of conjugate `xorshift64` generators versus the combined equidistribution score. Notice the improvement in correlation with respect to Figure 6.

has score 3 (the best is 2) whereas its conjugate $A_1(11, 5, 32)$ has score 106. It is clearly necessary to combine the equidistribution score of a generator and of its reverse.

This approach improves somewhat the quality of the result: Figure 8 shows that there is a better correlation between combined SmallCrush scores and combined equidistribution scores. Nonetheless, even if equidistribution is able to detect bad generators, is not so good at detecting the *very best* generators. We already noticed that only four generators (Table I) fail systematically a single SmallCrush test. These generators, however, are not the best ones by equidistribution score: in display order, they rank 4, 11, 3 and 23 (the other two generators rank 37 and 570, respectively). The first two generators by combined equidistribution score, $A_4(8, 29, 19)$ and $A_6(8, 29, 19)$, rank 20 (combined score 361) and 170 (score 596) in the combined SmallCrush test. When analyzed with the more powerful lens of BigCrush, they have combined scores 3441 and 4082, respectively, and fail systematically almost *twenty* additional tests with respect to the set A of Table II. Definitely, choosing among `xorshift64` generators by equidistribution score alone is not a good idea.

As a final interesting observation, in Figure 9 we correlate the equidistribution score and the weight score of `xorshift64` generators, both in combined and non-combined form. It is somewhat fascinating that two mathematically defined features which are supposed to lead

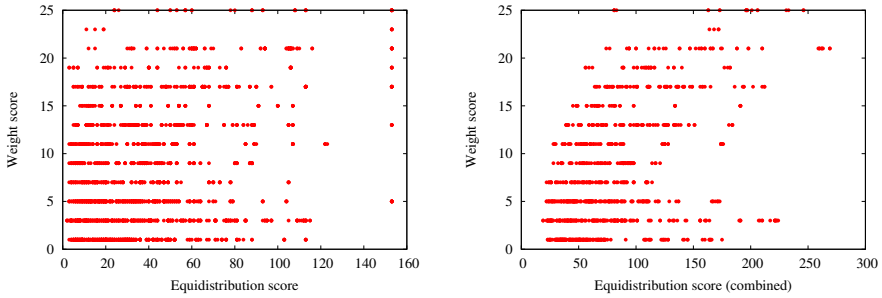


Fig. 9. Scatter plots of equidistribution scores versus weight scores for `xorshift64` generators.

Table IV. The three multipliers used in the rest of the paper. The subscripts recalls the t for which they have good figures of merit.

M_{32}	2685821657736338717
M_8	1181783497276652981
M_2	8372773778140471301

to good generators are entirely uncorrelated. Some mild correlation, as usual, appears if we use a combined equidistribution score.

5. RESULTS FOR `xorshift64*` GENERATORS

Since a `xorshift64` generator exhibits evident linearity artifacts, the next obvious step is to perturb its output using a nonlinear (in $\mathbf{Z}/2\mathbf{Z}$ sense) transformation. A natural candidate is multiplication by a constant, also because such operation is very fast in modern processors. Note that the current state of the generator is multiplied by a constant before returning it, but the state itself is not affected by the multiplication: thus, the period is the same.

We call such a generator `xorshift*`. By choosing a constant invertible modulo 2^{64}, we can guarantee that the generator will output a permutation of the sequence output by the underlying `xorshift` generator.

This approach was noted in passing in Marsaglia's paper, and it is also proposed in a more systematic way in the third edition of "Numerical Recipes" [Press et al. 2007] to create a very fast, good-quality pseudorandom number generator. However, in the latter case the author *first* computes allegedly good triples for `xorshift` using DieHard (with results markedly different from ours, and in strident contrast with TestU01's results, as discussed in Section 4) and *then* chooses a multiplier using a perfectly reasonable criterion (good spectral quality as a linear congruential generator). There is no reason why the best triples for a `xorshift64` generator (which are computed empirically) should continue to be such in a `xorshift64*` generator: and indeed, we will see that this is not the case.

We thus repeated the experiments of the previous section on `xorshift64*` generators. To further understand the dependency on the multiplier, we used three different multipliers, shown in Table IV. The first multiplier, M_{32}, is the one used in "Numerical Recipes", which is suggested by L'Ecuyer [1999] as having good spectral properties. The goodness of the multiplier, however, is established by a *figure of merit* which is a normalized best distance between the hyperplanes of families covering tuples of length t given by successive outputs of the generators. The length t is an additional parameter, and M_{32} has the best figures of merit for $t = 32$. Clearly, if an alternative multiplier provides improvements on both t and the associated figure of merit, we have a hint that it could be chosen instead of M_{32}.

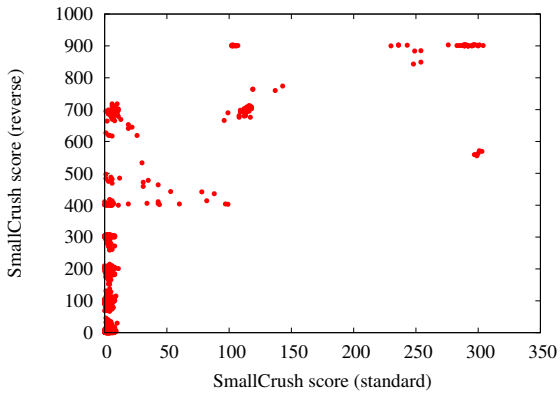


Fig. 10. Scatter plot of the SmallCrush score of `xorshift64*` generators and their reverse.

Lacking that possibility, what if we scramble `xorshift64`'s output with a multiplier that has *better* figures of merit for a *lower* t? We thus also ran experiment with the multiplier $M_8 = 1181783497276652981$, which has a better figure of merit for $t = 8$ [L'Ecuyer 1999], and the multiplier $M_2 = 8372773778140471301$, which has a better figure of merit for $t = 2$ and was kindly provided by Richard Simard.

The landscape is now very different. Indeed, the scatter plot in Figure 10 shows that there is no correlation between the scores assigned by SmallCrush to a generator and its reverse.[6]

Another interesting observation on Figure 10 is that the lower right half is essentially empty. So bad generators have a bad reverse, but there are good generators with a very bad reverse. This suggests that the quality of a `xorshift64*` generator can vary wildly from the low to the high bits.

A score-rank plot of the SmallCrush scores for all generators shown in Figure 11 provides us with further interesting information: almost all generators have no systematic failure, but only about half of the reverse generators have no systematic failure. Moreover, the distribution of standard generators degrades smoothly, whereas the distribution of reverse generators sports again the "bump" phenomenon we observed in Figure 2.

Finally, Figure 12 is the analogous of Figure 4 for `xorshift64*` generators: the mild correlation between combined SmallCrush score and weight score of the underlying `xorshift64` is now completely absent.

Since we need to reduce the number of candidates to apply stronger tests, in the case of M_{32} we decided to restrict our choice to generators with 3 overall failed tests or less, which left us with 152 generators. Similar cutoff were chosen for M_8 and M_2.

These generators were few enough so that we could apply both Crush and Dieharder. Once again, we examine the correlation between the score of a generator and its reverse by means of the scatter plots in Figure 13, which confirm the high-bits bias, albeit less so in the Dieharder case.

In Figure 14 we compare instead the two scores (Crush and Dieharder) available. The most remarkable feature is there are no points in the upper left corner: there is no generator that is considered good by Crush but not by Dieharder. On the contrary, Crush heavily penalizes (in particular because of the score on the reverse generator) a large number of generators.

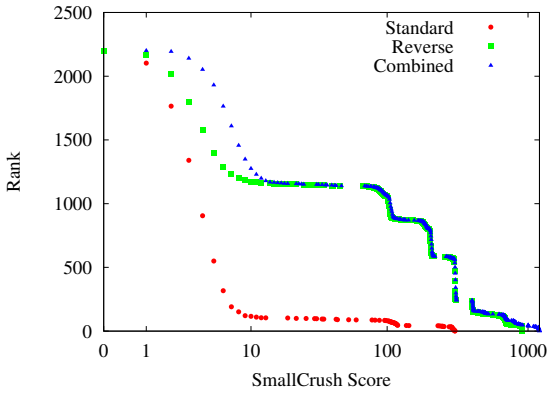


Fig. 11. Score-rank plot of the distribution of SmallCrush scores for the 2200 possible `xorshift64*` generators with multiplier M_{32}.

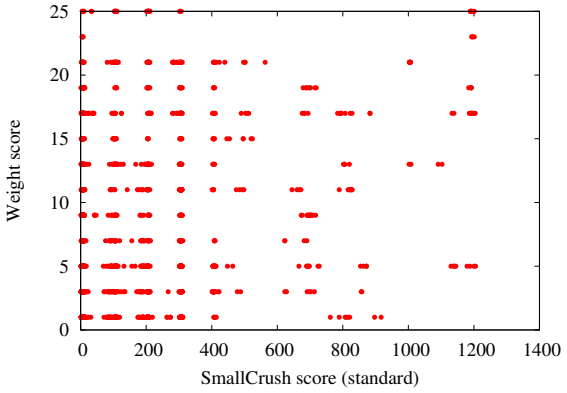


Fig. 12. Scatter plot of combined SmallCrush score or `xorshift64*` generators with multiplier M_{32} versus weights score of the underlying `xorshift64` generators.

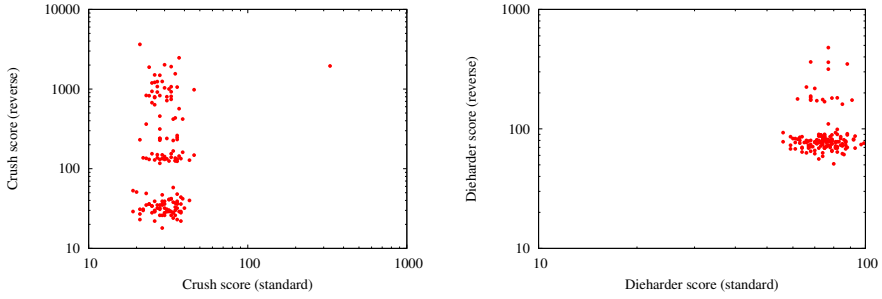


Fig. 13. Scatter plots for Crush (left) and Dieharder (right) scores on `xorshift64*` generators with multiplier M_{32} and their reverse, for the 152 best generators.

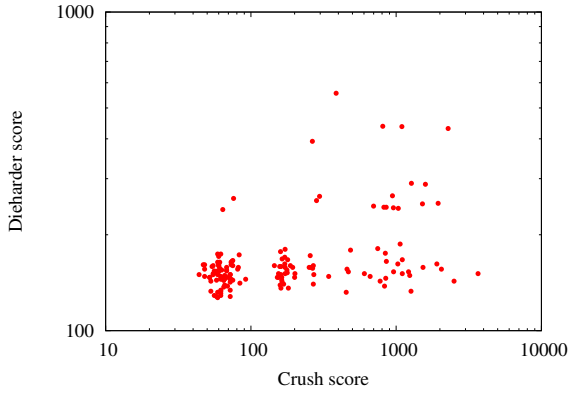


Fig. 14. A scatter plot of Crush and Diehard combined scores of the 152 SmallCrush-best `xorshift64*` generators. The plot is in log-log scale to accommodate some very high values returned by Crush on reverse generators. The "sweet spot" in the lower left corner contains generators that never fail systematically (not even reversed) in both test suites.

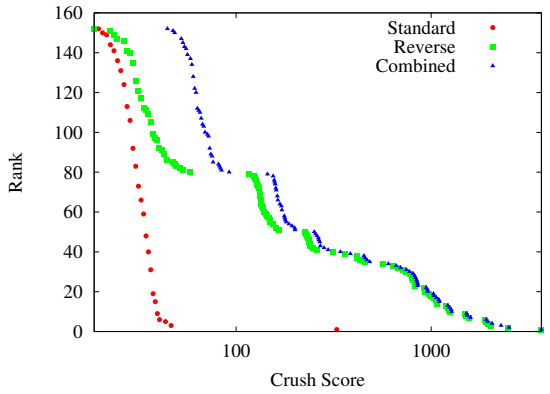


Fig. 15. Score-rank plot of the distribution of Crush scores for the 152 SmallCrush-best `xorshift64*` generators using multiplier M_{32}.

The generators we will select in the end all belong to the small cloud in the lower left corner, where the two test suite agree.

The score-rank plot in Figure 15 shows that our strategy pays off: we started with 152 generators with less than three failures, but analyzing them with the more powerful lens provided by Crush we get a much more fine-grained analysis: in particular, only 73 of them give no systematic failure, and they all belong to the "sweet spot" of Figure 14, that is, they do not give any systematic failure in Dieharder, too.

Finally, we selected for each multiplier the eight best generators with the best Crush score, and applied the BigCrush suite: we obtained several generators failing systematically the MatrixRank test only and shown in Table V (which should be compared with Table II). It is interesting to note that all generators have one additional systematic failure in the standard version with respect to the reversed version. This behavior is not surprising, as multiplication tends to make the lower bits more chaotic and of better quality than the

[6]In general, we report plots only for M_{32}, as the ones for the other multipliers are visually identical.

Table V. Results of BigCrush on the best eight xorshift64* generators found by SmallCrush and Crush in sequence. The generators fail systematically only MatrixRank.

Algorithm	Failures			W
	S	R	+	
M_{32}				
$A_7(11,5,45)$	226	128	354	23
$A_7(17,23,52)$	232	130	362	25
$A_1(12,25,27)$	230	133	363	31
$A_1(17,23,29)$	229	137	366	21
$A_5(14,23,33)$	238	132	370	32
$A_5(17,47,29)$	231	141	372	24
$A_1(16,25,43)$	238	138	376	31
$A_7(23,9,57)$	242	134	376	19
M_8				
$A_5(11,5,32)$	229	122	351	13
$A_2(8,31,17)$	229	126	355	21
$A_5(3,21,31)$	230	141	371	33
$A_3(17,45,22)$	241	133	374	27
$A_4(8,37,21)$	239	136	375	33
$A_3(13,47,23)$	232	144	376	27
$A_3(13,35,30)$	244	136	380	27
$A_4(9,37,31)$	243	141	384	27
M_2				
$A_7(13,19,28)$	228	128	356	23
$A_3(9,21,40)$	228	132	360	35
$A_1(14,23,33)$	234	142	376	29
$A_7(19,43,27)$	239	137	376	23
$A_1(17,47,28)$	240	137	377	25
$A_5(16,11,27)$	234	144	378	25
$A_4(4,35,15)$	230	149	379	35
$A_7(13,21,18)$	238	144	382	31

upper bits. It also suggests that for these generators it is better to extract the lower bits, rather than the high bits, when just a subsequence is needed.

5.1. Equidistribution

Multiplication by an invertible element just permutes the elements of $\mathbf{Z}/2^{64}\mathbf{Z}$ leaving zero fixed, so a `xorshift64*` generator, like the underlying `xorshift64` generator, is at least 1-dimensionally equidistributed. Since `xorshift64*` generators are not linear, analyzing more deeply equistribution scores would require an enormous computing effort, which is not justified in consideration of our observations on `xorshift64` generators. We tried nonetheless to correlate in Figure 16 the SmallCrush score of `xorshift64*` generators with the equidistribution score of the the underlying `xorshift64` generators. No such correlation appears in the generators taken in isolation—there are even generators, such as $A_7(15,1,19)\cdot M_{32}$ with the best SmallCrush combined score (no failure) for which the equidistribution score of the underlying `xorshift64` generator is the worst possible (153). However, once we combine both the SmallCrush scores and the equidistribution score some correlation appears. Once

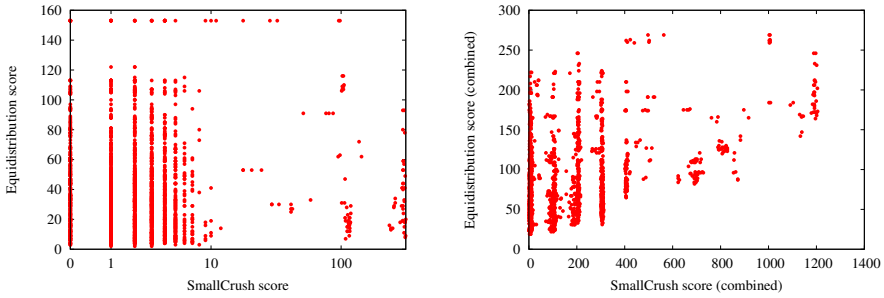


Fig. 16. Scatter plots of SmallCrush scores versus equidistribution scores of the underlying `xorshift64` generator for `xorshift64*` generators using multiplier M_{32}.

again, the combined equidistribution score is more useful to detect bad generators than to find the best ones, as the left part of the plot is quite chaotic.

6. HIGH DIMENSION

Marsaglia [2003] describes a strategy for `xorshift` generators in high dimension: the idea is to use always three low-dimensional shifts, but locating them in the context of a larger matrix of the form

$$M = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & (1+L^a)(1+R^b) \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & (1+R^c) \end{pmatrix}$$

Marsaglia notes that even in this restricted forms there are matrices of full period (he provides examples for 32-bit shifts up to 160 bits). However, we could found no evidence in the literature that this route has been explored for high-dimensional (say, more than 1024 bits of state) generators. The only similar approach is that proposed by Brent [2007] with his `xorgens` generators, which however uses more shifts. The obvious question is thus: are these additional shifts really necessary? We are thus going to look for good, full-period generators with 1024 or 4096 bits of state using 64-bit basic shifts.[7]

The output of such generators will be given by the last 64 bits of the state space. It is well known [Niederreiter 1992] that every bit of the state space satisfies a linear recurrence (defined by the characteristic polynomial) with full period, so *a fortiori* the last 64 bits have full period, too.

Since we already know that some deficiencies of low-dimensional `xorshift` generators are well corrected by multiplication by a constant, we will follow the same approach, thus looking for good `xorshift*` generators of high dimension.[8] Note that since multiplication by an integer invertible in $\mathbf{Z}/2^{64}\mathbf{Z}$ is a permutation of $\mathbf{Z}/2^{64}\mathbf{Z}$, a high-dimension `xorshift*` generator has the same period of the underlying `xorshift` generator.

[7]The reason why the number 4096 is relevant here is that we know the factorization of Fermat's numbers $2^{2^k}+1$ only up to $k = 11$. When more Fermat numbers will be factorized, it will be possible to design `xorshift` or `xorgens` generators with larger state space [Brent 2007]. Note that, however, in practice a period of $2^{1024}-1$ is more than sufficient for any purpose. For example, even if 2^{100} computers were to generate sequences of 2^{100} numbers starting from random seeds using a generator with period 2^{1024}, the chances that two sequences overlap would be less than 2^{-724}.

[8]As in the `xorshift64` case, different choices for the shifts are possible. We will not pursue them here.

We cannot in principle claim full period if we look at a *single* bit of the output of a `xorshift*` generator; but this property can be easily proved by exquisitely combinatorial means:

PROPOSITION 6.1. *Let $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{2^n-2}$ be a list of t-bit values, $t \leq n$, such that every value appears 2^{n-t} times, except for 0, which appears $2^{n-t}-1$ times. Then, for every fixed bit k the associated sequence has period 2^n-1.*

PROOF. Suppose that there is a k and a $p \mid 2^n-1$ such that the k-th bit of $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{2^n-2}$ has period p (that is, the sequence of bits associated with the k-th bit is made by $(2^n-1)/p$ repetitions of the same sequence of p bits). The k-th bit runs through $2^{n-1}-1$ zeroes and 2^{n-1} ones (as there is a missing zero). This means that $(2^n-1)/p \mid 2^{n-1}$, too, as the same number of ones must appear in every repeating subsequence, and since $(2^n-1)/p$ is odd this implies $p = 2^n-1$. □

COROLLARY 6.2. *Every bit of the output of a full-period `xorshift*` generator has full period.*

6.1. Finding good shifts

The first step is identifying values of a, b and c for which the generator has maximum period using the primitivity check on the characteristic polynomial. We performed these computations using the algebra package Fermat [Lewis 2013], with the restriction that $a + b \leq 64$ and that a is coprime with b (see [Brent 2007] for the rationale behind this choices, which significantly reduce the search space). The resulting sets of values are those shown in Table VI and VIII.

For a state space of 1024 bits, we obtain 20 possible parameter choices, which we examined in combination with our three multipliers both through BigCrush and through Dieharder. The results, reported in Table VI and VII, are excellent: with the exception of two pathological choices, no test is failed systematically. For 4096-bit state space (Table VIII and IX) there are 10 possible parameter choices, and no generator fails a test systematically.

SANITY CHECK 3. *Very long and complex computations are prone to implementation, software and hardware errors. In particular, if no verification procedure exists, results of a search on a large state space like the one we just describe are very difficult to assess. We thus decided to compute again the same coefficient using an entirely different algorithm: instead of working on characteristic polynomials, we developed highly optimized Java software that exploits the particular structure taken by powers of the linear transformation M associated with a generator to compute such powers explicitly, and store them using space linear in the size of the state space: as it is known [Marsaglia and Tsay 1985], the generator has full period if and only if M has the same multiplicative period. It is thus sufficient to show that $M^{2^n-1} = 1$ and*

$$M^{(2^n-1)/p} \neq 1$$

for every prime p dividing 2^{n-1}. The computation, in particular for the case of 4096 bits, turned out to be extremely intensive, requiring almost a month of computing time on a 40-core workstation and using more than half a terabyte of in-core memory, as each set of parameters can be checked in parallel, but for each such set we must compute and store the quadratures M^{2^k}, $0 < k < n$, so to be able to evaluate the condition above for all p's. The results confirmed those obtained by using primitive polynomials.

6.2. Equidistribution

Looking at the shape of the matrix defining high-dimensional `xorshift` generators it is clear that if the state space is made of n bits the last $n/64$ output values, concatenated, are equal to the current state. This implies that such generators are at least $n/64$-dimensionally

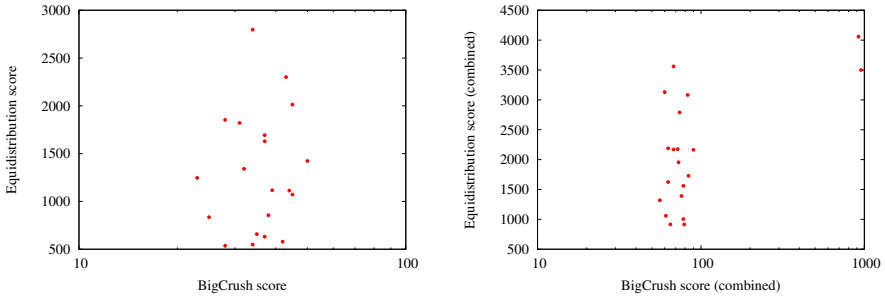


Fig. 17. Scatter plots of BigCrush scores versus equidistribution scores for `xorshift1024*` generators using multiplier M_{32}.

equidistributed (i.e., every $n/64$-tuple of consecutive 64-bit values appears exactly once, except for a missing tuple of zeroes), so `xorshift1024` generators are at least 16-dimensionally equidistributed and `xorshift4096` generators are at least 64-dimensionally equidistributed. Since multiplication by a constant just permutes the space of tuples, the same is true of the associated `xorshift*` generators.

We now repeat the analysis of Section 5.1, always using the `MTTollBox` package, and show the results in Figure 17. No correlation between BigCrush scores of `xorshift1024*` generators and equidistribution scores of the underlying `xorshift1024` appears for the generators taken in isolation, but once we combine both the BigCrush scores and the equidistribution scores some correlation appears, as the two pathological generators have indeed a very high combined equidistribution score. The analogous graphs for `xorshift4096*` generators are omitted; they display no correlation at all.

7. COMPARISON

How do our best `xorshift*` generators score with respect to more complex generators in the literature? We decided to perform a comparison with the popular Mersenne Twister `MT19937` [Matsumoto and Nishimura 1998],[9] with `WELL1024a`/`WELL19937a`, two generators introduced by Panneton et al. [2006] as an improvement over the Mersenne Twister, and with `xorgens4096`, a very recent 4096-bit generator introduced by Brent [2007] we mentioned in Section 6. All these generators are non-cryptographic and aim at fast, high-quality generation. As usual, 100 tests are performed at 100 equispaced points of the state space.

We choose generators from the `xorshift*` family that perform well on both BigCrush and Dieharder, have a good weight score and enough large parameters (which provide faster state change spreading): more precisely, the `xorshift64*` generator $A_1(12, 25, 27) \cdot M_{32}$ (Figure 19), `xorshift1024*` with parameters 31, 11, 30 and multiplier M_8 (Figure 20), and `xorshift4096*` with parameters 25, 3, 49 and multiplier M_2 (Figure 21).

7.1. Quality

Table X compares the BigCrush scores of the generators we discussed. We report also results on the Java standard random number generator, as a reality check with respect to stock generators currently found in computer languages.

The results are quite surprising. A simple 64-bit `xorshift*` generator has less linear artifacts than `MT19937`, `WELL1024a` or `WELL19937a` and, thus, a significantly better score.

[9] More precisely, with its 64-bit version.

Table VI. Results of BigCrush on the `xorshift1024*` generators. The last two generators fail systematically CouponCollector, Gap, HammingIndep, MatrixRank, SumCollector and WeightDistrib.

M_{32}					M_8					M_2				
a, b, c	Failures			W	a, b, c	Failures			W	a, b, c	Failures			W
	S	R	+			S	R	+			S	R	+	
27, 13, 46	25	31	56	275	1, 13, 7	28	19	47	113	3, 26, 35	29	24	53	89
31, 33, 37	28	32	60	79	3, 26, 35	29	22	51	89	27, 13, 46	41	20	61	275
22, 7, 48	37	24	61	223	40, 11, 31	24	33	57	77	25, 8, 15	38	24	62	281
7, 16, 55	37	26	63	65	15, 16, 19	30	32	62	255	31, 10, 27	36	31	67	233
9, 14, 41	23	40	63	167	22, 7, 48	29	33	62	223	9, 5, 60	24	43	67	227
41, 7, 29	28	37	65	265	9, 14, 41	32	30	62	167	1, 13, 7	28	42	70	113
1, 13, 7	34	34	68	113	41, 7, 29	25	38	63	265	15, 16, 19	36	34	70	255
10, 11, 61	32	36	68	155	31, 11, 30	33	32	65	363	2, 11, 61	40	30	70	81
9, 5, 60	44	28	72	227	2, 11, 61	25	41	66	81	41, 7, 29	36	34	70	265
16, 23, 30	37	36	73	59	10, 11, 61	42	25	67	155	9, 14, 41	33	37	70	167
3, 26, 35	45	29	74	89	7, 16, 55	32	35	67	65	22, 7, 48	37	35	72	223
25, 8, 15	42	34	76	281	16, 23, 30	35	34	69	59	31, 11, 30	45	27	72	363
31, 11, 30	35	43	78	363	25, 8, 15	25	45	70	281	7, 16, 55	36	39	75	65
40, 11, 31	38	40	78	77	27, 13, 46	39	32	71	275	31, 33, 37	37	39	76	79
31, 10, 27	34	45	79	233	31, 10, 27	40	32	72	233	10, 11, 61	41	37	78	155
2, 11, 61	43	40	83	81	9, 5, 60	40	36	76	227	16, 23, 30	44	37	81	59
15, 16, 19	45	39	84	255	31, 33, 37	39	39	78	79	40, 11, 31	38	48	86	77
10, 9, 63	39	51	90	69	10, 9, 63	31	49	80	69	10, 9, 63	48	48	96	69
51, 1, 46	31	890	921	111	51, 1, 46	60	896	956	111	51, 1, 46	31	799	830	111
47, 1, 41	50	902	952	99	47, 1, 41	67	907	974	99	47, 1, 41	47	799	846	99

Table VII. Results of Dieharder on `xorshift1024*` generators. No test is failed systematically.

M_{32}					M_8					M_2				
a, b, c	Failures			W	a, b, c	Failures			W	a, b, c	Failures			W
	S	R	+			S	R	+			S	R	+	
31, 33, 37	57	67	124	79	25, 8, 15	67	56	123	281	22, 7, 48	56	76	132	223
31, 11, 30	65	61	126	363	16, 23, 30	77	54	131	59	15, 16, 19	66	67	133	255
16, 23, 30	74	56	130	59	7, 16, 55	66	66	132	65	10, 9, 63	70	71	141	69
41, 7, 29	71	61	132	265	3, 26, 35	60	75	135	89	51, 1, 46	65	78	143	111
9, 14, 41	74	64	138	167	10, 11, 61	63	74	137	155	1, 13, 7	80	64	144	113
10, 9, 63	74	66	140	69	31, 10, 27	74	69	143	233	40, 11, 31	80	67	147	77
22, 7, 48	66	75	141	223	31, 33, 37	86	58	144	79	2, 11, 61	85	65	150	81
51, 1, 46	78	63	141	111	47, 1, 41	82	62	144	99	31, 11, 30	75	75	150	363
27, 13, 46	63	79	142	275	27, 13, 46	78	69	147	275	25, 8, 15	74	77	151	281
25, 8, 15	80	64	144	281	31, 11, 30	85	62	147	363	10, 11, 61	79	76	155	155
3, 26, 35	81	66	147	89	10, 9, 63	65	86	151	69	47, 1, 41	70	86	156	99
2, 11, 61	79	71	150	81	41, 7, 29	84	68	152	265	9, 5, 60	70	86	156	227
40, 11, 31	74	76	150	77	2, 11, 61	88	65	153	81	16, 23, 30	81	76	157	59
31, 10, 27	82	71	153	233	9, 14, 41	77	80	157	167	27, 13, 46	78	80	158	275
47, 1, 41	74	79	153	99	40, 11, 31	82	78	160	77	7, 16, 55	92	70	162	65
9, 5, 60	81	75	156	227	15, 16, 19	85	76	161	255	9, 14, 41	87	80	167	167
10, 11, 61	75	84	159	155	51, 1, 46	92	74	166	111	41, 7, 29	87	81	168	265
15, 16, 19	72	88	160	255	22, 7, 48	90	82	172	223	31, 10, 27	82	87	169	233
7, 16, 55	94	68	162	65	1, 13, 7	79	95	174	113	3, 26, 35	92	79	171	89
1, 13, 7	87	76	163	113	9, 5, 60	97	89	186	227	31, 33, 37	98	88	186	79

Table VIII. Results of BigCrush on xorshift4096* generators.

M_{32}					M_8					M_2				
Algorithm	Failures			W	Algorithm	Failures			W	Algorithm	Failures			W
	S	R	+			S	R	+			S	R	+	
14, 41, 15	33	27	60	241	5, 22, 27	34	35	69	45	11, 9, 25	30	33	63	567
5, 22, 27	34	30	64	45	5, 27, 21	36	35	71	187	5, 27, 21	37	27	64	187
30, 29, 39	33	32	65	177	25, 3, 49	35	37	72	441	25, 3, 49	33	34	67	441
25, 3, 49	30	38	68	441	7, 12, 59	34	39	73	103	19, 34, 19	39	36	75	291
7, 12, 59	43	25	68	103	11, 9, 25	40	34	74	567	23, 26, 29	40	35	75	49
19, 34, 19	34	36	70	291	12, 11, 61	41	33	74	195	30, 29, 39	38	37	75	177
12, 11, 61	32	39	71	195	19, 34, 19	39	35	74	291	12, 11, 61	40	37	77	195
5, 27, 21	34	41	75	187	14, 41, 15	43	34	77	241	14, 41, 15	36	42	78	241
23, 26, 29	36	42	78	49	30, 29, 39	42	37	79	177	7, 12, 59	38	44	82	103
11, 9, 25	35	44	79	567	23, 26, 29	38	43	81	49	5, 22, 27	38	50	88	45

Table IX. Results of Dieharder on `xorshift4096*` generators.

M_{32}					M_8					M_2				
Algorithm	Failures			W	Algorithm	Failures			W	Algorithm	Failures			W
	S	R	+			S	R	+			S	R	+	
25, 3, 49	70	70	140	441	25, 3, 49	67	70	137	441	19, 34, 19	75	64	139	291
12, 11, 61	58	83	141	195	14, 41, 15	72	69	141	241	5, 22, 27	67	77	144	45
30, 29, 39	67	77	144	177	30, 29, 39	70	75	145	177	25, 3, 49	77	71	148	441
5, 22, 27	62	84	146	45	11, 9, 25	73	77	150	567	5, 27, 21	77	71	148	187
11, 9, 25	73	75	148	567	12, 11, 61	75	80	155	195	11, 9, 25	81	76	157	567
19, 34, 19	85	66	151	291	19, 34, 19	89	67	156	291	14, 41, 15	79	78	157	241
14, 41, 15	83	74	157	241	5, 22, 27	93	65	158	45	23, 26, 29	74	84	158	49
7, 12, 59	73	85	158	103	23, 26, 29	72	87	159	49	12, 11, 61	74	85	159	195
23, 26, 29	73	88	161	49	5, 27, 21	75	84	159	187	7, 12, 59	84	79	163	103
5, 27, 21	98	67	165	187	7, 12, 59	90	77	167	103	30, 29, 39	78	89	167	177

Table X. A comparison of generators using BigCrush.

Algorithm	Failures			W	Systematic
	S	R	+		
$A_1(12, 25, 27) \cdot M_{32}$	230	133	363	31	MatrixRank
$A_3(4, 35, 21) \cdot M_{32}$	240	223	463	25	MatrixRank, BirthdaySpacings
`xorshift1024*`	29	22	51	363	—
`xorshift4096*`	33	34	67	441	—
`xorgens4096`	42	40	82	961	—
`MT19937`	258	258	516	6750	LinearComp
`WELL1024a`	441	441	882	407	MatrixRank, LinearComp
`WELL19937a`	235	233	468	8585	LinearComp
`java.util.Random`	4078	9486	13564	—	Almost all

Table XI. A comparison of generators using Dieharder.

Algorithm	Failures		
	S	R	+
$A_1(12, 25, 27) \cdot M_{32}$	86	61	147
$A_3(4, 35, 21) \cdot M_{32}$	76	71	147
`xorshift1024*`	70	75	135
`xorshift4096*`	77	71	148
`xorgens4096`	80	78	158
`MT19937`	72	79	151
`WELL1024a`	81	61	142
`WELL19937a`	86	67	153
`java.util.Random`	4078	9486	13564

Note: No test is failed systematically, except for `java.util.Random` , which fails systematically dab_bytedistrib, dab_dct, diehard_craps, diehard_dna, diehard_operm5, diehard_opso, diehard_oqso, diehard_squeeze, rgb_kstest_test, rgb_lagged_sum, rgb_minimum_distance and rgb_permutations.

High-dimension `xorgens4096` and `xorshift*` generators perform significantly better, in spite of being extremely simpler, and have no systematic failure. The 64-bit `xorshift*` generator suggested by "Numerical Recipes" fails systematically the BirthdaySpacings test, contrarily to our selection. The Java standard generator is, in fact, unusable.[10]

Additionally, Table XI reports the results of Dieharder: the main observation is that at this level of quality Dieharder is unable to make any distinction between the generators, except for the case of the Java generator.

7.2. Escaping zeroland

We show in Figure 18 the speed at which a few of the generators of Table X "escape from zeroland" [Panneton et al. 2006]: purely linearly recurrent generators with a very large state space need a very long time to get from an initial state with a small number of ones to a state in which the ones are approximately half. The figure shows a measure of escape time given by the ratio of ones in a window of 4 consecutive 64-bit values sliding over the first 100 000

[10]Note that we report the number of *failed tests* on our 100 seeds. L'Ecuyer and Simard [L'Ecuyer and Simard 2007] report the number of *types of failed tests* (e.g., failing two distinct RandomWalk tests counts as one) on a single run, so some care must be taken when comparing the results we report and those reported by them.

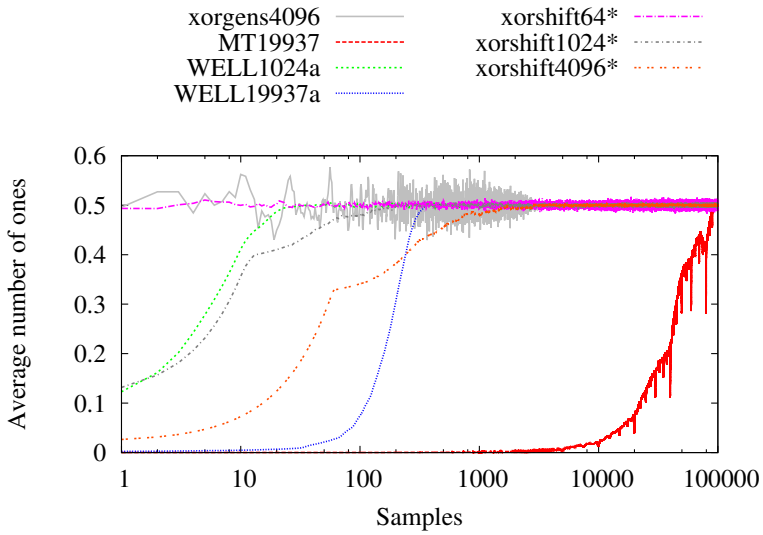


Fig. 18. Convergence to "half of the bits are ones in average" plot.

Table XII. Mean and variance for the data shown in Figure 18.

Algorithm	Mean	Variance
`xorshift64*`	0.5000	0.0039
`xorgens4096`	0.4999	0.0030
`xorshift1024*`	0.4999	0.0035
`WELL1024a`	0.4999	0.0036
`xorshift4096*`	0.4991	0.0110
`WELL19937a`	0.4991	0.0184
`MT19937`	0.3129	0.1689

generated values, averaged over all possible seeds with exactly one bit set (see [Panneton et al. 2006] for a detailed description).

As it is known, `MT19937` needs hundreds of thousands of iterations to start behaving correctly. `xorshift4096*` and `xorgens4096` need a few thousand (but `xorgens4096` oscillates always around 1/2), `WELL19937a` and `xorshift1024*` a few hundreds, whereas `WELL1024a` just a few dozens, and `xorshift64*` is almost unaffected.

Table XII condenses Figure 18 into the mean and variance of the displayed values. Clearly, the multiplication step helps in reducing the correlation between the number of ones in the state space and the number of ones in the output values. Also, the slowness in recovering from states with too many zeroes it directly correlated to the size of the state space—a very good argument against linear generators with too large state spaces.

7.3. Speed

Finally, we benchmark the generators of Table X. Our tests were run on an Intel® Core™ i7-4770 CPU @3.40GHz (Haswell), and the results are shown in Table XIII (variance is undetectable, as we generate 10^{10} values in each test). We also report as a strong baseline results about `SFMT19937`, the *SIMD-Oriented Fast Mersenne Twister* [Saito and Matsumoto

Table XIII. Time to emit a 64-bit integer on an Intel® Core™ i7-4770 CPU @3.40GHz (Haswell).

Algorithm	Speed (ns/64 bits)
xorshift64*	1.60
xorshift1024*	1.36
xorshift4096*	1.36
xorgens4096	1.68
MT19937	4.09
SFMT19937	1.54
WELL1024a	5.18
WELL19937a	8.01

2008], a 128-bit version of the Mersenne Twister based on the SSE2 extended instruction set of Intel processors (and thus not usable, in principle, on other processors).

The highest speed is achieved by the high-dimensional xorshift* generators. Note that the timings in Table XIII include the looping logic, which we approximately benchmarked at 17 ns/iteration. This means that the xorgens4096 generator is actually 27% slower than a xorshift1024* or xorshift4096* generator. SFMT19937 is a major improvement in speed over MT19937, albeit slightly slower than a high-dimensional xorshift* generator; it fails systematically, moreover, the same tests of MT19937.

A xorshift64* generator is actually *slower* than its high-dimensional counterparts. This is not surprising, as the three shift/xors in a xorshift64* generator form a dependency chain and must be executed in sequence, whereas two of the shifts of a higher-dimension generator are independent and can be internally parallelized by the CPU. WELL1024a and WELL19937a are heavily penalized by their 32-bit structure.

```
#include <stdint.h>

uint64_t x;

uint64_t next() {
    x ^= x >> 12; // a
    x ^= x << 25; // b
    x ^= x >> 27; // c
    return x * 2685821657736338717LL;
}
```

Fig. 19. The suggested xorshift64* generator in C99 code. The variable x should be initialized to a nonzero seed before calling next().

8. CONCLUSIONS

After our careful experimental analysis, we reach the following conclusions:

A xorshift1024* generator is an excellent choice for a general-purpose, high-speed generator. The statistical quality of the generator is very high (it has, actually, the best results in BigCrush), and its period is so large that the probability of overlapping sequences is practically zero, even in the largest parallel simulation. Nonetheless, the state space is reasonably small, so that seeding it with high-quality bits is not too expensive,

```
#include <stdint.h>

uint64_t s[ 16 ];
int p;

uint64_t next(void) {
    uint64_t s0 = s[ p ];
    uint64_t s1 = s[ p = ( p + 1 ) & 15 ];
    s1 ^= s1 << 31; // a
    s1 ^= s1 >> 11; // b
    s0 ^= s0 >> 30; // c
    return ( s[ p ] = s0 ^ s1 ) * 1181783497276652981LL;
}
```

Fig. 20. The suggested `xorshift1024*` generator in C99 code. The array `s` should be initialized to a nonzero seed before calling `next()`.

```
#include <stdint.h>

uint64_t s[ 64 ];
int p;

uint64_t next(void) {
    uint64_t s0 = s[ p ];
    uint64_t s1 = s[ p = ( p + 1 ) & 63 ];
    s1 ^= s1 << 25; // a
    s1 ^= s1 >> 3;  // b
    s0 ^= s0 >> 49; // c
    return ( s[ p ] = s0 ^ s1 ) * 8372773778140471301LL;
}
```

Fig. 21. The suggested `xorshift4096*` generator in C99 code. The array `s` should be initialized to a nonzero seed before calling `next()`.

and recovery from states with a large number of zeroes happens quickly. The generator is also blazingly fast (it is actually the fastest generator we tested), providing a 64-bit value in slightly more than a nanosecond. The reasonable state space makes also more likely, in case a large number of generators is used at the same time, that their state can fit the cache. In any case, with respect to other generators, the state space is accessed is a more localized way, as read and write operations happen *at two consecutive locations*, and thus will generate at most one cache miss.

In case memory is an issue, or array access is expensive, a very good general-purpose generator is a `xorshift64*` generator. While the generator $A_1(12, 25, 27) \cdot M_{32}$ fails systematically the MatrixRank test, it has less linear artifacts tham `MT19937`, `WELL1024a` or `WELL19937a`, which fail systematically even more tests. It is a very good choice if memory footprint is an issue and a very large number of generators is necessary. A `xorshift64*` generator can also actually be *faster* than a `xorshift1024*` generator if the underlying language incurs in significant costs when accessing an array: for instance, in Java a `xorshift64*` generator emits a value in 1.62 ns, whereas a `xorshift1024*` generator needs 2.08 ns.

Linear generators with an excessively long period have a number of problems that are not compensated by higher statistical quality. Generating 64 bits with

WELL19937a requires almost ten times the time required by a xorshift1024* generator, with no detectable improvement in the statistical quality of the output by means of test suites; moreover, recovery from state spaces with many zeroes, albeit enormously improved with respect to MT19937, is still very slow, and seeding properly the generator requires almost twenty thousands random bits. In the end, it is in general difficult to motivate state spaces larger than 2^{1024}. Similar considerations are made, for example, by L'Ecuyer and Panneton [2005].

Surprisingly simple and fast generators can produce sequences that pass strong statistical tests. The code in Figure 20 is extremely shorter and simpler than that of MT19937, WELL1024a or WELL19937a. Yet, it performs significantly better on BigCrush. It is a tribute to Marsaglia's cleverness that just eight logical operations, one addition and one multiplication by a constant can produce sequences of such high quality. xorgens generators are similar with this respect, but use several more operations due to the additional shift and to the usage of a *Weyl generator* to hide linear artifacts [Brent 2007].

The t for which the multiplier has a good figure of merit has no detectable effect on the quality of the generator. If our tests, we could not find any significant difference between the behavior of generators based on M_{32}, M_8 or M_2. It could be interesting to experiment with multipliers having very *bad* figures of merit.

Equidistribution is more useful as a design feature than as an evaluation feature. While *designing* generators around equidistribution might be a good idea, as it leads in general to good generators, *evaluation* by equidistribution is a more delicate matter because of high-bits bias and of the failure to detect the generators having the best scores in statistical suites (actually, as we have seen, some of the *worst* generators could be chosen instead).

TestU01 has significantly more resolution than Dieharder as a test suite. In particular in the high-dimension case, TestU01 is able to provide useful information, whereas Dieharder scores flatten down. However, TestU01 should be applied always to the reverse generator, too, to account for its high-bits bias.

REFERENCES

Richard P. Brent. 2004. Note on Marsaglia's Xorshift Random Number Generators. *Journal of Statistical Software* 11, 5 (2004), 1–5.

Richard P. Brent. 2007. Some long-period random number generators using shifts and xors. In *Proceedings of the 13th Biennial Computational Techniques and Applications Conference, CTAC-2006 (ANZIAM J.)*, Wayne Read and A. J. Roberts (Eds.), Vol. 48. C188–C202.

Richard P. Brent. 2010. The myth of equidistribution for high-dimensional simulation. *CoRR* abs/1005.1320 (2010).

Robert G. Brown. 2013. Dieharder: A Random Number Test Suite (Version 3.31). (2013). Retrieved January 8, 2014 from http://www.phy.duke.edu/~rgb/General/dieharder.php

Aaldert Compagner. 1991. The hierarchy of correlations in random binary sequences. *Journal of Statistical Physics* 63, 5-6 (1991), 883–896.

Shin Harase. 2011. An efficient lattice reduction method for $\mathbf{F}_2$-linear pseudorandom number generators using Mulders and Storjohann algorithm. *J. Comput. Appl. Math.* 236, 2 (2011), 141–149.

Maurice G. Kendall. 1938. A New Measure of Rank Correlation. *Biometrika* 30, 1/2 (1938), 81–93.

Maurice G. Kendall. 1945. The treatment of ties in ranking problems. *Biometrika* 33, 3 (1945), 239–251.

Pierre L'Ecuyer. 1999. Tables of linear congruential generators of different sizes and good lattice structure. *Math. Comput* 68, 225 (1999), 249–260.

Pierre L'Ecuyer and François Panneton. 2005. Fast random number generators based on linear recurrences modulo 2: overview and comparison. In *Proceedings of the 37th Winter Simulation Conference*. Winter Simulation Conference, 110–119.

Pierre L'Ecuyer and Richard Simard. 2007. TestU01: A C library for empirical testing of random number generators. *ACM Trans. Math. Softw.* 33, Article 22 (August 2007). Issue 4.

Robert H. Lewis. 2013. Fermat: A Computer Algebra System for Polynomial and Matrix Computation (Version 5.1). (2013). Retrieved January 8, 2014 from http://home.bway.net/lewis/

Rudolf Lidl and Harald Niederreiter. 1994. *Introduction to finite fields and their applications*. Cambridge University Press, Cambridge.

George Marsaglia. 2003. Xorshift RNGs. *Journal of Statistical Software* 8, 14 (2003), 1–6.

George Marsaglia and Liang-Huei Tsay. 1985. Matrices and the structure of random number sequences. *Linear Algebra Appl.* 67 (1985), 147–156.

Makoto Matsumoto and Takuji Nishimura. 1998. Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. *ACM Trans. Model. Comput. Simul.* 8, 1 (1998), 3–30.

Harald Niederreiter. 1992. *Random number generation and quasi-Monte Carlo methods*. CBMS-NSF regional conference series in Appl. Math., Vol. 63. SIAM.

François Panneton. 2004. *Construction d'ensembles de points basé sur une récurrence linéaire dans un corps fini de caractéristique 2 pour la simulation Monte Carlo et l'intégration quasi-Monte Carlo*. Ph.D. Dissertation. Université de Montréal.

François Panneton and Pierre L'Ecuyer. 2005. On the xorshift random number generators. *ACM Trans. Model. Comput. Simul* 15, 4 (2005), 346–361.

François Panneton, Pierre L'Ecuyer, and Makoto Matsumoto. 2006. Improved long-period generators based on linear recurrences modulo 2. *ACM Trans. Math. Softw.* 32, 1 (2006), 1–16.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 2007. *Numerical recipes: the art of scientific computing*. Cambridge University Press.

Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, and San Vo. 2001. *A Statistical Test Suite For Random and Pseudorandom Number Generators for Cryptographic Applications*. National Institute for Standards and Technology, pub-NIST:adr. NIST Special Publication 800-22, with revisions dated May 15, 2001.

Mutsuo Saito. 2013. MTToolBox (Version 0.2). (2013). Retrieved January 8, 2014 from http://msaito.github.io/MTToolBox/en/

Mutsuo Saito and Makoto Matsumoto. 2008. SIMD-Oriented Fast Mersenne Twister: a 128-bit Pseudorandom Number Generator. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*, Alexander Keller, Stefan Heinrich, and Harald Niederreiter (Eds.). Springer, 607–622.

Spectrum Sensing Via Reconfigurable Antennas: Fundamental Limits and Potential Gains

Ahmed M. Alaa, *Student Member, IEEE,* Mahmoud H. Ismail, *Member, IEEE* and Hazim Tawfik

***Abstract*—We propose a novel paradigm for spectrum sensing in cognitive radio networks that provides diversity and capacity benefits using a single antenna at the Secondary User (SU) receiver. The proposed scheme is based on a *reconfigurable antenna*: an antenna that is capable of altering its radiation characteristics by changing its geometric configuration. Each configuration is designated as an antenna *mode* or *state* and corresponds to a distinct channel realization. Based on an abstract model for the reconfigurable antenna, we tackle two different settings for the cognitive radio problem and present fundamental limits on the achievable diversity and throughput gains. First, we explore the *"to cooperate or not to cooperate"* tradeoff between the diversity and coding gains in conventional cooperative and non-cooperative spectrum sensing schemes, showing that cooperation is not always beneficial. Based on this analysis, we propose two sensing schemes based on reconfigurable antennas that we term as *state switching* and *state selection*. It is shown that each of these schemes outperform both cooperative and non-cooperative spectrum sensing under a global energy constraint. Next, we study the *"sensing-throughput"* trade-off, and demonstrate that using reconfigurable antennas, the optimal sensing time is reduced allowing for a longer transmission time, and thus better throughput. Moreover, state selection can be applied to boost the capacity of SU transmission.**

***Index Terms*—cognitive radio; cooperative spectrum sensing; diversity; ergodic capacity; reconfigurable antennas; spectrum sensing**

I. Introduction

COGNITIVE Radio (CR) is a promising technology offering a significant enhancement in wireless systems spectrum efficiency via dynamic spectrum access [1]. In a CR network, unlicensed secondary users (SUs) can opportunistically occupy the unused spectrum allocated to a licensed primary user (PU). This is achieved by means of PU signal detection. Detection of PU signal entails sensing the spectrum occupied by the licensed user in a continuous manner. Thus, the process of *spectrum sensing* is mandatory for a CR system as it helps preserving the Quality-of-Service (QoS) experienced by the licensed PU. Energy detection (ED) is one of the simplest spectrum sensing techniques as it can be implemented using simple hardware and does not require Channel State Information (CSI) at the SU receiver [2]–[3]. Generally, the performance of a spectrum sensing technique severely degrades in slow fading channels. To combat this effect, Cooperative Spectrum Sensing (CSS) schemes have been proposed to take advantage of the spatial diversity in wireless channels [4]–[6]. In CSS, hard or soft decisions from different CR users are combined to make a global decision at a central unit known as the *Fusion Center* (FC). CSS has been widely accepted in the literature as a realizable technique for extracting spatial diversity. The other alternative would be using multiple antennas, which is constrained by the space limitation in SU recievers [6]–[14].

The authors are with the Department of Electronics and Electrical Communications Engineering, Cairo University, Gizah, 12613, Egypt (e-mail: {aalaa, mismail, htawfik}@eece.cu.edu.eg).

Manuscript received XXXX XX, 201X; revised XXXX XX, 201X.

A. Background and Motivation

Although CSS achieves a diversity gain that is equal to the number of cooperating users, it encounters a significant cooperation overhead: several decisions taken at SU terminals have to be fed back to the FC via a dedicated reporting channel [5]; global information (including the number of cooperating SU terminals) must be provided to each SU in order to calculate the optimal detection threshold [6]; hard decisions taken locally at each SU cause loss of information, which degrades the performance at low signal-to-noise ratio (SNR) [7]; and finally, the existence of multiple SUs is not always guaranteed. In addition, in this work, we show that there exists a trade-off between the coding gain and the diversity order achieved in both cooperative and non-cooperative schemes, and demonstrate that cooperation is actually not beneficial in the low SNR regime. Motivated by these disadvantages, we tackle the following question: *can we dispense with secondary users cooperation and still achieve an arbitrary diversity gain?* To answer this question, we propose a novel spectrum sensing scheme that can indeed achieve an arbitrary diversity order for a single SU and still uses a single antenna. The scheme is based on the usage of *reconfigurable antennas*; a class of antennas capable of changing its geometry, hence changing the current distribution over the volume of the antenna and thus altering one of its propagation characteristics: operating frequency, polarization or radiation pattern. Each geometrical configuration thus leads to a different mode of operation leading to different realizations of the perceived wireless channel. Switching between various antenna modes could be done using microelectromechanical (MEMS) switches [15], nano-electromechanical switches (NEMS), or solid state switches [16].

In [15], the concept of an electrically reconfigurable antenna was first introduced based on RF MEMS switches. Many research efforts followed this concept and proposed actual designs for antennas that can alter their geometric configuration [17]–[21]. The usage of reconfigurable antennas in wireless communications was studied in various contexts. For example, based on an abstract conceptual model, diversity benefits of reconfigurable antennas in MIMO systems were discussed in [20]. Also, in [22], a new class of space-time codes, termed

as *state-space-time* codes was introduced, where it was shown that reconfigurable antennas can offer diversity benefits but has no impact on the achieved degrees of freedom. Moreover, reconfigurable antennas were employed in the context of interference alignment in [16], where desirable channel fluctuations were created by switching the antenna modes over time.

B. Summary of Contributions

In this paper, we propose a single user CR system that employs a reconfigurable antenna at the SU transceivers. By switching the antenna *radiation states* over time, we can manipulate the wireless channel thus creating artificial channel fluctuations that turn a slow fading channel into a fast fading one. Capitalizing on this property, we show that we can dispense with the spatial diversity achieved through cooperation without encountering any degradation in the sensing performance. Besides, the proposed scheme has the following advantages: 1) the full coding and diversity gains are captured at any SNR, 2) the space limitation problem that inhibits the usage of multiple antennas is solved by using a single compact antenna, 3) unlike multiple antenna systems, only one RF chain is needed, 4) the availability of CSI at the SU can be used to even boost the achieved coding gain, and 5) diversity is achieved with no cooperation overhead, which usually involves setting up a dedicated reporting channel; feeding back information from the FC to the SU terminals; and maintaining synchronization between the SU devices.

Another approach for sensing using reconfigurable antennas is to select the "best" state instead of randomly switching among various states. When the CSI is available at the SU, the receiver can select the state that offers the strongest channel gain. Therefore, in addition to the previously stated advantages, state selection offers an additional SNR gain, that we term as the *selection gain*. Based on a comprehensive diversity analysis, we obtain the achievable diversity orders in the conventional and proposed schemes as a function of the detection threshold based on Neyman-Pearson (NP) and Bayes tests.

While there exists many antenna switching techniques with different ranges of switching delays [18], some classes of switching devices, such as those based on mechanical switches, may exhibit significant switching delays that may affect the performance of the proposed schemes. Thus, we quantify the impact of an arbitrary switching delay on the performance of the proposed schemes in both the NP and Bayesian tests.

Moreover, we revisit a well known trade-off in CR systems, which is the "Sensing-throughput trade-off". In a frame-structured CR system, each frame duration is divided into sensing and transmission periods. An optimal sensing time that compromises between the detection performance and the achieved throughput was calculated in [23]–[25]. We show that using reconfigurable antennas, and given a constraint on the PU detection probability, the SU throughput is improved as a longer period of the frame can be dedicated to transmission rather than sensing, in addition to the reduction of the false alarm probability, which means better channel utilization.

Finally, we show that reconfigurable antennas are not only beneficial in the sensing phase, but can also offer significant capacity gains in the transmission phase (when the SU accesses the channel). To that end, we obtain closed-form expressions for the average transmission capacity using state selection, and taking into consideration the impact of switching delay.

The rest of the paper is organized as follows: Section II presents the signal model adopted in the spectrum sensing problem and relevant derivations for the false alarm and missed detection probabilities. In Section III, we discuss the "to cooperate or not to cooperate" tradeoff, identifying the drawbacks of the cooperative scheme. Spectrum sensing via reconfigurable antennas is introduced in Section IV, and the diversity orders obtained in sensing based on NP and Bayes criterion are derived. In section V, the impact of reconfigurable antennas on the sensing-throughput tradeoff is studied, showing the achievable throughput gains. In addition, the gains achieved in SU transmission and the optimal switching strategy are analyzed. Finally, we draw our conclusions in Section VI.

II. Spectrum Sensing Signal Model

A. System Model and Notations

In this section, we formulate the spectrum sensing problem for the conventional and proposed schemes, and clarify the notations of *diversity order* and *coding gain*.

The diversity order d_* for a performance metric P_* with an average SNR of $\overline{\gamma}$ is defined as [6]

$$d_* = -\lim_{\overline{\gamma}\to\infty} \frac{\log P_*}{\log \overline{\gamma}}.$$

The performance metric P_* usually represents either the probability of error, the false alarm probability or the missed detection probability. The metric P_* corresponds to the missed detection probability P_{md} in the NP optimization problem, and corresponds to the error probability P_e if the optimization problem adopts the Bayesian criterion. As for the coding gain, it is defined as the multiplication factor of the average SNR in P_* as $\overline{\gamma}$ tends to infinity. Thus, if $P_* \asymp \frac{1}{(A\overline{\gamma})^d}$ as $\overline{\gamma} \mapsto \infty$, the coding gain is given by A and the diversity order is d, where $\asymp$ denotes asymptotic equality. The diversity order affects the slope of the P_* curve when plotted versus the average SNR (in dB), while the coding gain shifts the P_* curve along the SNR curve. In spectrum sensing using energy detection, the coding gain is indeed sensitive to the average energy involved in detection. Hence, the average energy can be used to quantify the shift of the P_* curve. Without loss of generality, we are interested in evaluating the asymptotic missed detection and error probabilities at high SNR only in order to obtain the diversity order and coding gain using the previous definitions. It is important to note, however, that both gains characterize the performance for all ranges of SNR.

Now, it is required to compare the detection performance of non-cooperative sensing, cooperative sensing, and reconfigurable antenna based schemes. Hereunder, we present the system model for the three schemes under study.

1) Non-cooperative scheme: A conventional non-cooperative spectrum sensing scheme involves one SU that observes M samples for spectrum sensing. According to the sampling theorem, for a sensing period of T and a signal with bandwidth W, the number of samples is $M = 2\,TW$ [26]. It is assumed that the instantaneous SNR is γ and the primary signal i^{th} sample is $S_i \sim \mathcal{CN}(0,1)$ [7], where $\mathcal{CN}(\mu,\sigma^2)$ denotes the complex Gaussian distribution with mean μ and variance σ^2. The additive white noise is $n_i \sim \mathcal{CN}(0,1)$. Thus, the i^{th} sample received at the SU receiver is a binary hypothesis given by [7]

$$r_i = \begin{cases} n_i \sim \mathcal{CN}(0,1), & \mathcal{H}_o \\ \sqrt{\gamma}\, S_i + n_i \sim \mathcal{CN}(0,1+\gamma), & \mathcal{H}_1 \end{cases} \quad (1)$$

where $\mathcal{H}_o$ denotes the absence of the PU, while $\mathcal{H}_1$ denotes the presence of the PU. After applying such signal to an energy detector, the resulting test statistic is $Y = \sum_{i=1}^{M} |r_i|^2$, which follows a central chi-squared distribution for both $\mathcal{H}_o$ and $\mathcal{H}_1$. The false alarm and detection probabilities are given by [7]

$$P_F(M,\lambda) = P(Y > \lambda|\mathcal{H}_o) = \frac{\Gamma(M, \frac{\lambda}{2})}{\Gamma(M)},$$

and

$$P_D(M,\lambda,\gamma) = P(Y \leq \lambda|\mathcal{H}_1,\gamma) = \frac{\Gamma\left(M, \frac{\lambda}{2(1+\gamma)}\right)}{\Gamma(M)}, \quad (2)$$

where λ is the detection threshold, $\Gamma(.,.)$ is the upper incomplete gamma function, and $\Gamma(.)$ is the gamma function. We assume Rayleigh fading with an average SNR of $\overline{\gamma}$ and that the instantaneous SNR is constant over the M observed samples (slow fading). Different observations perceive different SNR values. The SNR varies according to the exponential probability density function (pdf)

$$f_\gamma(\gamma) = \frac{1}{\overline{\gamma}} e^{-\frac{\gamma}{\overline{\gamma}}}, \gamma \geq 0. \quad (3)$$

Because the detection probability is a function of the slow fading channel gain, we obtain the average detection probability as

$$\overline{P}_D = \int_0^\infty \frac{\Gamma\left(M, \frac{\lambda}{2(1+\gamma)}\right)}{\Gamma(M)} \frac{1}{\overline{\gamma}} e^{-\frac{\gamma}{\overline{\gamma}}}\, d\gamma. \quad (4)$$

In order to evaluate the average detection probability, we can rewrite the integrands in (4) in terms of the Meijer-G function $G_{p,q}^{m,n}\left(\begin{smallmatrix} a_1,\ldots,a_p \\ b_1,\ldots,b_q \end{smallmatrix} \middle| z\right)$ [27, Sec. 7.8] as

$$\Gamma\left(M, \frac{\lambda}{2(1+\gamma)}\right) = G_{1,2}^{2,0}\left(\begin{smallmatrix} 1 \\ M, 0 \end{smallmatrix} \middle| \frac{\lambda}{2(1+\gamma)}\right),$$

and

$$e^{-\frac{\gamma}{\overline{\gamma}}} = G_{0,1}^{1,0}\left(\begin{smallmatrix} - \\ 0 \end{smallmatrix} \middle| \frac{\gamma}{\overline{\gamma}}\right).$$

The Meijer-G representation allows us to write the integral in (4) as

$$\overline{P}_D = \int_0^\infty G_{0,1}^{1,0}\left(\begin{smallmatrix} - \\ 0 \end{smallmatrix} \middle| \frac{\gamma}{\overline{\gamma}}\right) G_{1,2}^{2,0}\left(\begin{smallmatrix} 1 \\ M, 0 \end{smallmatrix} \middle| \frac{\lambda}{2(1+\gamma)}\right) d\gamma. \quad (5)$$

With the aid of [27, Eq. 7.811.1], the integral is approximated at high SNR as

$$\overline{P}_D \approx \frac{\lambda e^{\frac{1}{\overline{\gamma}}}}{2\overline{\gamma}\Gamma(M)} G_{1,3}^{3,0}\left(\begin{smallmatrix} 0 \\ M-1, -1, 0 \end{smallmatrix} \middle| \frac{\lambda}{2\overline{\gamma}}\right), \quad (6)$$

which can be further reduced into the form of [27, Sec. 7.8]

$$\overline{P}_D = \frac{2\, e^{\frac{1}{\overline{\gamma}}}}{\Gamma(M)} \left(\frac{\lambda}{2\overline{\gamma}}\right)^{\frac{M}{2}} K_M\left(\sqrt{\frac{2\lambda}{\overline{\gamma}}}\right), \quad (7)$$

where $K_M(.)$ is the M^{th} order modified bessel function of the second kind.

2) Cooperative Scheme: A cooperative CR network consists of N SUs, each senses the PU signal and reports its decision to an FC. The FC employs an ***n-out-of-N*** fusion rule to take a final global decision. We let l be the test statistic denoting the number of votes for the presence of a PU. Hence, the conditional pdfs follow a *binomial distribution* [5] where $P(l|\mathcal{H}_o) = \binom{N}{l}\ P_F^l\ (1-P_F)^{N-l}$, and $P(l|\mathcal{H}_1) = \binom{N}{l}\ \overline{P}_D^l\ (1-\overline{P_D})^{N-l}$, where P_F is the local false alarm probability, and $\overline{P}_D$ is the local detection probability averaged over the pdf of the SNR. Based on the fusion rule mentioned above, the global false alarm and detection probabilities P_{F_G} and P_{D_G} are

$$P_{F,G} = \sum_{l=n}^{N} \binom{N}{l} P_F^l \left(1-P_F\right)^{N-l},$$

$$P_{D,G} = \sum_{l=n}^{N} \binom{N}{l} \overline{P}_D^l \left(1-\overline{P}_D\right)^{N-l}. \quad (8)$$

3) Single user spectrum sensing using a reconfigurable antenna: In the proposed scheme, we assume a single SU that employs a reconfigurable antenna to sense the PU signal. Establishing the exact mathematical models for the relation between an antenna mode and the corresponding channel realization can be a daunting task. We postulate that reconfigurable antennas have an arbitrary number of possible configurations/modes (i.e., radiation patterns), and that the corresponding induced wireless channels are independent from one another (all possible radiation patterns are spatially uncorrelated). For a reconfigurable antenna with Q radiation modes, we assume that $E_i(\Omega)$ and $E_j(\Omega)$ are the 3D radiation patterns corresponding to modes i and j respectively, and Ω is the solid angle describing the azimuth and elevation planes. Note that the solid angle ranges from 0 to 4π steradian. The spatial correlation coefficient between the two radiation patterns is given by [21]

$$\rho_{i,j} = \frac{\int_{4\pi} E_i(\Omega) E_j^*(\Omega) d\Omega}{\sqrt{\int_{4\pi} |E_i(\Omega)|^2 d\Omega \int_{4\pi} |E_j(\Omega)|^2 d\Omega}}.$$

A reconfigurable antenna is designed such that all radiation patterns are orthogonal, i.e. $\rho_{i,j} \approx 0$, $\forall i,j \in \{1,2,3,\ldots,Q\}$. For a rich scattering environment, the equivalent channel realizations encountered by different antenna states are i.i.d (independent and identically distributed) and follow a Rayleigh distribution. Various designs for antennas with pattern diversity

already exist [17]–[20]. The application of reconfigurable antennas with orthogonal patterns for MIMO systems was investigated in [22]. Moreover, in [16] and [28], blind interference alignment was proposed based on reconfigurable antennas with independent channels for each state. In [29], independent channel realizations were also exploited while studying the benefits of applying reconfigurable antennas in the MIMO Z interference channel. The impact of independent channel realizations perceived for different states result in a diversity gain that is similar to the spatial diversity gain attained in multiple antenna systems [30]. A conceptual model for the reconfigurable antenna that resembles an antenna selection scheme is adopted in this paper. The analyses we present herein are abstract in the sense that they do not consider a specific antenna design. Fig. 1 depicts the SU receiver employing a reconfigurable antenna with Q available antenna modes.

In a slow fading channel, reconfigurable antennas with Q modes can offer Q different channel realizations. Thus, the i^{th} sample received at the SU receiver is a binary hypothesis given by

$$r_i = \begin{cases} n_i \sim \mathcal{CN}(0,1), & \mathcal{H}_o \\ \sqrt{\gamma_j}\, S_i + n_i \sim \mathcal{CN}(0, 1+\gamma_j), & \mathcal{H}_1 \end{cases} \tag{9}$$

where $\gamma_j \in \{\gamma_1, \gamma_2, \cdots, \gamma_Q\}$ is the channel realization observed by the i^{th} sample. The set of Q channel gains are independent identically distributed (i.i.d.) Rayleigh random variables. It is assumed that the antenna states are switched Q times within the sensing period such that channel realization j is observed by l_j samples where $\sum_{j=1}^{Q} l_j = M$. We designate this scheme as *state switching spectrum sensing*. As an alternative, if the CSI is available at the receiver, the SU could possibly select the strongest channel for the entire sensing interval, and we call this scheme *state selection spectrum sensing*. Generally, the test statistic resulting at the output of the energy detector when the PU is active can be written as

$$Y = \sum_{j=1}^{L} (1+\gamma_j)\, x_j,$$

where L is the number of antenna states involved in sensing ($L \leq Q$), γ_j is one of Q independent channel realizations $\{\gamma_1, \gamma_2, \ldots, \gamma_Q\}$ assigned to the l_j samples, and x_j is a chi-square distributed random variable with $2l_j$ degrees of freedom (the sum of l_j normally distributed random variables). For state selection, L = 1 and $l_1 = M$ as only the highest channel gain is selected. For state switching, $L \leq Q$ and $\sum_{j=1}^{L} l_j = M$. Thus, the probability of missed detection is given by

$$P_{md}(\gamma_1, \ldots, \gamma_Q) = P\left(\sum_{j=1}^{L} (1+\gamma_j)\, x_j \leq \lambda | \mathcal{H}_1, \gamma_1, \ldots, \gamma_Q \right), \tag{10}$$

where the threshold λ is adjusted such that the false alarm probability P_F = α in the NP test, or adjusted to minimize the error probability in the Bayesian test. It is obvious that the probability of missed detection is the cumulative distribution function (CDF) of the linear combination of chi-square random variables. An extremely accurate approximation for the CDF of the sum of weighted chi-square random variables was proposed in [31]. Based on Eqs. (20)–(23) in [31], the probability of missed detection will be given by the minimum of two functions $H(w)$ and $G(w)$ of an auxiliary parameter w as follows

$$P_{md} = \min\{H(w), G(w)\},$$

where

$$w = \frac{\lambda}{M + \sum_{j=1}^{Q} l_j \gamma_j},$$

$$G(w) = \sum_{j=1}^{2M} w\, \frac{1+\gamma_j}{\lambda} \times \frac{\Upsilon\left(\frac{\lambda}{2w(1+\gamma_j)}, \frac{\lambda}{1+\gamma_j}\right)}{\Gamma(\frac{\lambda}{2w(1+\gamma_j)})},$$

and

$$H(w) = \frac{\Upsilon\left(M, \frac{\lambda}{\sqrt[M]{\prod_{i=1}^{Q}(1+\gamma_j)^{l_j}}}\right)}{\Gamma(M)}.$$

Thus, the missed detection probability in terms of the channel realizations is given by (11) where $\Upsilon(\cdot,\cdot)$ is the lower incomplete gamma function. Eq. (11) is general for any antenna state switching pattern. For state selection, the same result still applies with $l_k = M$, where $k = \max_j \gamma_j$ and $l_{k'} = 0, k' \neq k, k' \in \{1, 2, \cdots, Q\}$.

B. Equivalence of NP and Bayesian Optimization to Diversity Order Maximization

The only design parameters in the spectrum sensing problem are the detection thresholds. Usually, the thresholds are selected such that the detection performance is optimized in terms of either the NP or Bayesian criteria. Obtaining the optimal detection threshold is essential for calculating the diversity order achieved by the SU receiver. However, the problem of obtaining the detection thresholds that maximize the detection or minimize the error probabilities is not always mathematically tractable, especially in the cooperative scheme [7]. In this subsection, we formulate an equivalent problem for obtaining these optimal thresholds and we show that maximizing (minimizing) a performance metric P_* is equivalent to maximizing the diversity order d_* at assymptotically high SNR. Thus, as an alternative approach, one can obtain closed-form expressions for the diversity order d_* in terms of the detection thresholds and get the thresholds that maximize d_* instead of maximizing (minimizing) P_*, which is usually a mathematically tractable problem. This is formulated in the following two lemmas.

Lemma 1: *Based on the NP criterion, maximizing the high SNR asymptotic probability of detection under a false alarm probability constraint is equivalent to maximizing the diversity order of the detection probability.*

proof See Appendix A.

Lemma 2: *Based on the Bayes detection criterion, minimizing the high SNR asymptotic probability of error is equivalent to maximizing the diversity order of the error probability.*

Proof See Appendix B.

$$P_{md}(\gamma_1,\ldots,\gamma_Q)=\min\left\{\frac{\Upsilon\left(M,\frac{\lambda}{\sqrt[M]{\prod_{i=1}^{Q}(1+\gamma_i)^{l_i}}}\right)}{\Gamma(M)},\ \sum_{i=1}^{2M} w\,\frac{1+\gamma_i}{\lambda}\times\frac{\Upsilon\left(\frac{\lambda}{2w(1+\gamma_i)},\frac{\lambda}{1+\gamma_i}\right)}{\Gamma\left(\frac{\lambda}{2w(1+\gamma_i)}\right)}\right\}. \tag{11}$$

In the next section, we utilize these equivalent problems to compare the performance of the cooperative and non-cooperative schemes.

III. To Cooperate or Not to Cooperate

Although cooperation is widely adopted as a means of improving the performance of spectrum sensing via diversity gain, it can actually be shown that cooperative spectrum sensing does not outperform the non-cooperative scheme for the whole SNR range. Deciding whether to cooperate or not to cooperate should then depend on the operating average SNR. Specifically, for a fixed total energy constraint, the non-cooperative scheme offers a better detection performance at low SNR. This is because, at low SNR, the impact of SNR loss in the cooperative scheme due to hard decisions taken locally at each SU is higher than the gain offered by cooperation [1]. On the other hand, a large diversity gain is observed at high SNR making cooperation favorable. Therefore, cooperation would not be beneficial at low SNR ranges where it is required to improve the detection performance. In addition to that, the knowledge of the number of cooperating users at each SU is essential to achieve full diversity order. Thus, even at high SNR, cooperative schemes may fail to capture full diversity gain if global network information are not provided to local SUs. In the following two subsections, we compare the two schemes and evaluate their performance in terms of diversity and coding gains, both for NP and Bayes tests.

A. Non-cooperative scheme analysis

Considering the NP test, the asymptotic expansion of $K_M(x)$, which appears in the P_d expression in (7), as $x\mapsto 0$ is given by [14]

$$K_M(x)\asymp x^{-M}\Big(2^{M-1}\Gamma(M)-\frac{2^{M-3}\Gamma(M)x^2}{M-1}+\frac{2^{M-6}\Gamma(M)x^4}{(M-1)(M-2)}+\ldots\Big).$$

Note that $\sqrt{\frac{2\lambda}{\overline{\gamma}}}\mapsto 0$ and $e^{\frac{1}{\gamma}}\mapsto 1$ as $\overline{\gamma}\mapsto\infty$. The asymptotic expansion of the detection probability is consequently given by

$$\overline{P}_D\asymp 1-\frac{\lambda}{2\overline{\gamma}(M-1)}+\frac{\lambda^2}{8\overline{\gamma}^2(M-1)(M-2)}+\ldots.$$

Thus, at large average SNR, the first two terms dominate and $\overline{P}_D=1-\frac{\lambda}{2\overline{\gamma}(M-1)}+\mathcal{O}(\overline{\gamma}^{-2})$. Hence, the average missed detection probability is $\overline{P}_{md}=1-\overline{P}_D\approx\frac{\lambda}{2\overline{\gamma}(M-1)}$. As defined in Section II, the diversity order d and coding gain A are, respectively, given by

$$d_{md}=-\lim_{\overline{\gamma}\to\infty}\frac{\log\left(\frac{\lambda}{2\overline{\gamma}(M-1)}\right)}{\log\overline{\gamma}}=1,$$

and

$$A=\frac{M-1}{\lambda}. \tag{12}$$

Eq. (12) shows the diversity order and coding gain in terms of the threshold λ. It is clear that for the non-cooperative NP test, any choice of the local threshold does not affect the diversity order and the optimal threshold is selected such that it satisfies the constraint on P_F. The coding gain, on the other hand, depends on the number of samples involved in energy detection as well as the local threshold λ. The more samples involved in detection, the higher coding gain is achieved. On the other hand, large thresholds corresponding to strict false alarm constraints result in small coding gains. Note that for an α-level NP test, the local threshold is decided by the value of α when setting $P_F=\alpha$.

Now considering the Bayes optimization problem, the optimal threshold is given by the following lemma. **Lemma 3:** *The optimal threshold that minimizes the average probability of error in non-cooperative spectrum sensing is given by*

$$\lambda_{opt}=\mu^{\frac{1}{M-1}}\exp\left(-\mathcal{W}_{-1}\left(\frac{-\mu^{\frac{1}{M-1}}}{2(M-1)}\right)\right),$$

at high SNR, where $\mu=\frac{P(\mathcal{H}_1)}{P(\mathcal{H}_o)}\times\frac{2^{M-2}\Gamma(M-1)}{\overline{\gamma}}$ and $\mathcal{W}_{-1}(.)$ is the *Lambert W* function [33].

Proof See Appendix C.

In order to investigate the impact of the threshold on the diversity order, we calculate the diversity order achieved with a non-optimal threshold in the following Lemma: **Lemma 4:** *For conventional spectrum sensing with a detection threshold of $\lambda=\theta\lambda_{opt}$ where $\theta\in\mathbb{R}$ and λ_{opt} is the optimal Bayes threshold given by Lemma 3, the achieved diversity order for the Bayes optimization problem is $d_e=\min\{\theta,1\}$. The corresponding false alarm and missed detection probabilities are given by Eq. (13).*

Proof See Appendix D.

As stated in Lemma 4, for any threshold with $\theta>1$ (or equivalently $\lambda>\lambda_{opt}$), the maximum diversity order is achieved. However, given the expression of P_{md} in Eq. (13), the coding gain is $A_{md}=\frac{1}{\theta}$ if $\theta\geq 1$, and $A_F=\left(\frac{1}{\theta}\right)^{\frac{1}{\theta}}$ if $\theta\leq 1$. Thus, it is clear that the *coding gain* decreases with the increase of θ. Thus, the optimum Bayesian threshold corresponds to the *minimum λ that achieves the maximum diversity order* $d_{e_{max}}=d_{md}$. Because d_F is an increasing function of θ, we can obtain the optimum threshold by equating d_F to d_{md} instead of minimizing P_e, which is not mathematically

[1]No SNR degradation would be encountered if SUs send soft decisions to the FC. However, this is not practically feasible as the reporting channel is usually limited [5].

$$P_F \asymp \frac{1}{\Gamma(M)} \left(\theta(M-1) \log \left(\frac{M-1}{\Gamma(M-1)^{\frac{1}{M-1}}} \overline{\gamma}^{\frac{1}{M-1}} \right) \right)^{M-1} \left(\frac{\Gamma(M-1)^{\frac{1}{M-1}}}{(M-1)\overline{\gamma}^{\frac{1}{M-1}}} \right)^{\theta(M-1)},$$

and

$$P_{md} \asymp \frac{\theta}{\overline{\gamma}} \log \left(\frac{M-1}{\Gamma(M-1)^{\frac{1}{M-1}}} \overline{\gamma}^{\frac{1}{M-1}} \right). \tag{13}$$

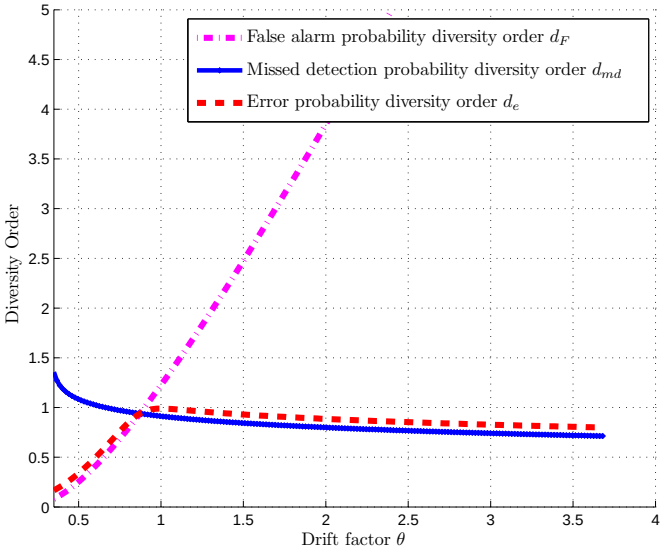


Fig. 1. Diversity orders (d_F, d_{md} and d_e) versus the drift factor θ for the conventional spectrum sensing scheme.

tractable. The behavior of the achieved diversity order versus the factor θ, that we denote as the *drift factor*, is depicted in Fig. 2. It is shown that the optimal threshold corresponding to $\theta = 1$ represents the intersection of d_F and d_{md}. This implies the following proposition.

Proposition 1. The optimal Bayesian threshold can be obtained by solving the transcendental equation

$$d_F(\lambda) = d_{md}(\lambda).$$

B. Cooperative scheme analysis: the good, the bad, and the ugly

In cooperative sensing, local thresholds are employed by individual SU receivers to take local hard decisions, while a global threshold (an integer number) is used by the fusion center to take the final decision. In this subsection, we relate the local and global thresholds, λ and n, to the coding gain and diversity order. Next, we select the thresholds so that the global false alarm probability $P_{F,G} = \alpha$ and the diversity order is maximized, which corresponds to the NP test. Then, we select the thresholds that maximize the error probability diversity order, which corresponds to the Bayesian test. We characterize the performance of energy constrained CSS as being multifaceted with three basic aspects: a "*good*" aspect, which is achieving diversity order of N at assymptotically high SNR; a "*bad*" aspect, which is the poor coding gain causing performance degradation at low SNR; and an "*ugly*" aspect, which is the inability to achieve the full diversity order when the SUs do not know the number of cooperating SUs N. In this case, cooperation does not reach the maximum possible diversity gain in addition to having a poor coding gain, questioning its usefulness. Hereunder, we present a comprehensive study for the performance of the cooperative scheme.

Based on (8), the global missed detection probability is given by

$$P_{md,G}(n,\lambda) = \sum_{l=0}^{n-1} \binom{N}{l} \overline{P}_{md}^{N-l}(\lambda) \; (1-\overline{P}_{md}(\lambda))^l. \tag{14}$$

It is obvious that $\overline{P}_{md} \mapsto 0$ as $\overline{\gamma} \mapsto \infty$. The last term in the series in (14) dominates and the asymptotic value of $P_{md,G}$ becomes

$$P_{md,G}(n,\lambda) \asymp \binom{N}{n-1} \left(\frac{\lambda}{2\,\overline{\gamma}\,(M-1)} \right)^{N-n+1}. \tag{15}$$

Thus, by rearranging (15) in the form of $(A\overline{\gamma})^{-d}$, the diversity order d_{md} and coding gain A_{md} in terms of the local and global thresholds are given by

$$d_{md,G} = N - n + 1,$$

$$A_{md,G} \propto \binom{N}{n-1}^{\frac{-1}{N-n+1}} \frac{M-1}{\lambda}.$$

Clearly, the global threshold that maximizes the diversity order is n = 1, which is known as the OR rule [5]. Hence, if only one SU votes for the presence of a primary user, the fusion center adopts its decision. The local threshold λ is chosen such that $P_{F,G}$ = α.

Based on the above analysis, it can be concluded that cooperative spectrum sensing with N SU receivers can offer a diversity order of N. The larger N is, the higher the diversity order is, but the more information is lost due to hard decisions taken locally at each SU. This is demonstrated by the fact that the coding gain $A_{md,G} \propto M$ at $n = 1$, which is as low as $\frac{1}{N}$ of the total number of samples (NM) involved in detection, but the diversity gain will be maximized and $d_{md,G} = N$. In the low SNR region, information loss due to poor coding gain is more critical and we do not benefit from multiuser diversity. Thus, for a fixed total energy constraint, it is better not to cooperate when the SNR is low as assigning the total energy to a single SU leads to a better detection performance.

To demonstrate the tradeoff between coding and diversity gains, we compare a cooperative network with N SU terminals and M samples per terminal with a non-cooperative network with a single SU and NM samples. Note that the total sensed energy is constant in both cases to ensure a fair comparison. Let the local thresholds in the multiple and single-user cases

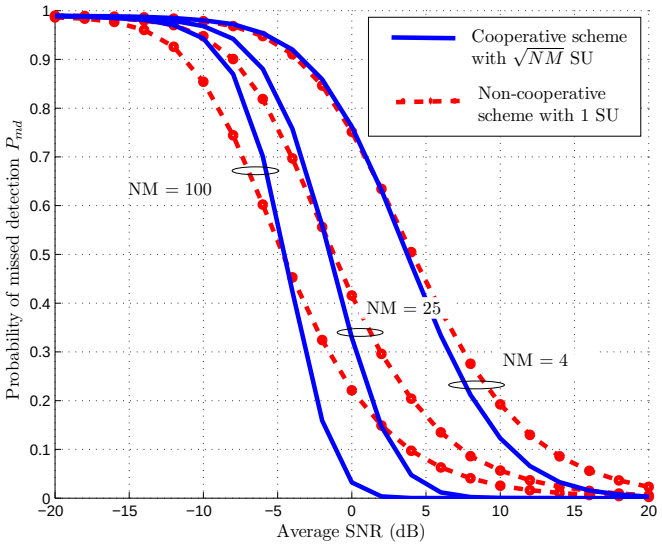


Fig. 2. To cooperate or not to cooperate tradeoff.

be $\lambda_{N,M}$ and $\lambda_{1,NM}$, respectively. Based on the above results, the coding gain would be $\frac{M-1}{\lambda_{N,M}}$ in the cooperative scheme and $\frac{NM-1}{\lambda_{1,NM}}$ in the non-cooperative scheme. Thus, the coding gain of the non-cooperative scheme is boosted by a factor of N. This factor is reduced as $\lambda_{N,M}$ and $\lambda_{1,NM}$ are not generally equal.

Fig. 3 depicts the tradeoff under study. Simulations were carried out for cooperative and non-cooperative schemes and the missed detection probability is plotted versus the average SNR. The NM product is fixed for both schemes and is set to 4, 25 and 100. This product represents the total energy constraint involved in detection. For each value of NM, the cooperative scheme employs $\sqrt{NM}$ SU terminals and $\sqrt{NM}$ samples per terminal [2]. On the other hand, the non-cooperative scheme employs 1 SU using NM samples. By applying the NP test and setting α = 0.01, it is found that at NM = 100, the non-cooperative scheme outperforms the cooperative scheme by 3 dB at low SNR. Thus, it is better not to cooperate if the operating SNR is less than -5 dB, which is the SNR value corresponding to the intersection of the P_{md} curves for both schemes. The SNR gain is reduced in the NM = 25 scenario and nearly vanishes when NM = 4. On the other hand, the cooperative scheme offers large gains in the high SNR region. For instance, at P_{md} = 0.03 and NM = 100, cooperation outperforms non-cooperative sensing by an SNR gain of 7 dB due to the multiuser diversity. The larger N is, the more gain one gets at high SNR, but at the expense of the coding gain for a fixed energy constraint.

For the Bayesian optimization problem, we obtain the global false alarm probability by taking the dominant term of the binomial expansion in (8)

$$P_{F,G}(n,\lambda) \asymp \binom{N}{n} \left(\frac{\Gamma(M,\frac{\lambda}{2})}{\Gamma(M)}\right)^n,$$

[2]Any combination of the number of SU terminals and the number of samples that keeps the NM product constant can be used in the analysis.

Based on the series expansion $\frac{\Gamma(M,\frac{\lambda}{2})}{\Gamma(M)} = \sum_{i=0}^{M-1} \frac{\lambda^i}{2^i \Gamma(i+1)} e^{\frac{-\lambda}{2}}$ [6], we can approximate the false alarm probability as

$$P_{F,G}(n,\lambda) \approx \binom{N}{n} \left(\frac{\lambda^{M-1}}{2^{M-1}\Gamma(M)}\right)^n e^{\frac{-\lambda}{2}n}. \quad (16)$$

We substitute λ in (14) and (16) with the locally optimal threshold multiplied by the factor θ. Our objective is to obtain the value of θ that maximizes the diversity order of the global error probability. The global false alarm and detection probabilities in terms of θ are given in (17).

From (17), it is obvious that $d_{md,G} = N - n + 1$, while $d_{F,G} = n\theta$. Thus, the diversity order of the error probability is

$$d_{e,G} = \min\{N - n + 1, n\theta\}.$$

We investigate the achievable diversity order in two different scenarios as follows:

- **The number of cooperating users N is unknown at SU receivers:** In this case, we aim at selecting the global threshold n and the local threshold $\theta\lambda_{opt}$, such that θ is not a function of N. The optimal thresholds are obtained based on the following optimization problem

$$\max_{n,\theta} \ \min\{n\theta, N - n + 1\}$$
$$\text{s.t. } n\theta = N - n + 1.$$

  Because the number of SUs is unknown at each SU, we select a locally optimal threshold for each SU by setting $\theta = 1$. Combining this fact with *Proposition 1*, we obtain the optimal global threshold by solving the equation $n = N - n + 1$, which yields a global threshold of $n = \lfloor \frac{N+1}{2} \rfloor$ [3]. Thus, the corresponding diversity order is

$$d_e = \min\left\{\lfloor\frac{N+1}{2}\rfloor, \lceil\frac{N+1}{2}\rceil\right\} = \lfloor\frac{N+1}{2}\rfloor.$$

  Thus, the "ugly" face of CSS appears when global information are not provided to local SUs. Note that for N = 2, cooperation without global knowledge of N yields no diversity gain at all.
- **The number of cooperating users N is known at SU receivers:** It is obvious that $d_{md,G}$ is maximized by setting n = 1. Applying *Proposition 1*, the optimal value of θ is N. The corresponding diversity order $d_{e,G} = N$, thus the full diversity order is achieved in this case.

It is worth mentioning that global knowledge of N is also needed in the NP test. However, the lack of knowledge of N in the NP problem has no effect on the diversity order. Instead, it turns the problem into a *discrete hypothesis detection problem* [32], where only discrete values of $P_{F,G} = \alpha$ are realizable. As mentioned earlier, tolerating a larger α comes at the expense of the coding gain and not the diversity order.

To sum up, whether to cooperate or not to cooperate depends on several factors. If the operating SNR is low, it is better not to cooperate as the coding gain is severely degraded in the cooperative systems impacting performance at low SNR.

[3]Throughout this paper, the operator $\lfloor . \rfloor$ is the flooring operator, while $\lceil . \rceil$ is the ceiling operator.

$$P_{F,G} \asymp \binom{N}{n} \left(\frac{\left(2(M-1)\theta \log\left(\frac{M-1}{\Gamma(M-1)^{\frac{1}{M-1}}} \overline{\gamma}^{\frac{1}{M-1}} \right) \right)^{M-1}}{2^{M-1}\Gamma(M)} \right)^{n} \left(\frac{\Gamma(M-1)^{\frac{1}{M-1}}}{M-1} \right)^{\theta n(M-1)} \frac{1}{\overline{\gamma}^{\theta n}},$$

$$P_{md,G} \asymp \binom{N}{n-1} \left(\frac{\theta \log\left(\frac{M-1}{\Gamma(M-1)^{\frac{1}{M-1}}} \overline{\gamma}^{\frac{1}{M-1}} \right)}{\overline{\gamma}} \right)^{N-n+1}. \quad (17)$$

Moreover, if the number of SUs is not known, we can not achieve the full diversity order in the Bayesian test. For small number of cooperating users (e.g., N = 2), the system will not offer significant diversity gain and cooperation may not be worth it. Stemming from this analysis, we study the performance of the proposed single reconfigurable antenna schemes in the next section. Such schemes are capable of overcoming all the drawbacks of cooperation and achieving the full diversity and coding gains thus offering a superior performance compared to the conventional schemes for the entire SNR range.

IV. Spectrum Sensing Via Reconfigurable Antennas

As stated earlier, reconfigurable antennas can artificially induce fluctuations in the slow fading channel. This would create temporal diversity for a single SU network, which can offer a gain similar to the spatial diversity gain in the cooperative scheme. We investigate two basic schemes for spectrum sensing using a reconfigurable antenna: a *state switching* scheme (when the CSI is unknown) and a *state selection* scheme (when the CSI is available). Based on the signal model presented in Section II, we derive the optimal test statistic for spectrum sensing with an arbitrary selection of antenna modes over time, where each mode j is selected for l_j sensing samples.

Lemma 5: *For spectrum sensing using reconfigurable antennas with arbitrary antenna state selection over time, let* $Z_j = \sum_{i=l_{j-1}+1}^{l_{j-1}+l_j} |r_i|^2, j \in \{1,2,\cdots,Q\}, l_o = 0$, L *is the number of antenna states invoked within the sensing period* ($L \leq Q$), *and* η *is an arbitrary detection threshold. The Likelihood Ratio Test (LRT) reduces to*

$$\sum_{j=1}^{L} \frac{\gamma_j}{1+\gamma_j} Z_j \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \eta$$

proof See Appendix E.

Note that the LRT described in Lemma 5 requires the knowledge of the channel realizations corresponding to different antenna states, and involves a test statistic that is calculated via *weighted energy detection* rather than simple energy detection. If the CSI is not available at the SU (i.e., the set of channel realizations $\{\gamma_1, \gamma_2, \cdots, \gamma_Q\}$ is unknown), the test in Lemma 5 denotes a *hypothesis detection problem with unknown parameters* [32]. Because the test statistic depends on the unknown parameters, no Uniformly Most Powerful (UMP) test exists, and we adopt a suboptimal test that involves simple energy detection without assigning weights to energy samples. In the state switching scheme, we blindly select an arbitrary number of channels over the sensing period such that $L \leq Q$ and $\sum_{j=1}^{L} l_j = M$. On the other hand, if the CSI is available at the SU, we adopt the state selection scheme instead, where the strongest channel realization is selected for the entire sensing period (i.e. $L = 1, l_k = M$, and $k = \max_j \gamma_j$).

A. Optimal sensing based on NP Criterion

1) Spectrum Sensing via State Switching: The missed detection probability for an arbitrary antenna mode switching pattern is given by (11). Given that $\Upsilon(M,x) \asymp \frac{x^M}{M}$ as $\overline{\gamma} \to \infty$ [27], the asymptotic values of $H(w)$ and $G(w)$ are $\frac{\lambda^M}{\Gamma(M+1)\prod_{j=1}^{Q}(1+\gamma_j)^{l_j}}$ and $\sum_{j=1}^{2M} w \frac{1+\gamma_j}{\lambda}$, respectively, which implies that $\min\{G(w), H(w)\} = H(w)$ at high SNR. Thus, one can calculate the diversity order based on $P_{md} \approx H(w)$. The asymptotic missed detection probability will then be given by

$$P_{md}(\gamma_1, \ldots, \gamma_Q) \asymp \frac{\lambda^M}{\Gamma(M+1)\prod_{j=1}^{Q}(1+\gamma_j)^{l_j}}. \quad (18)$$

By averaging the missed detection probability in (18) over the pdf of Q independent Rayleigh channel realizations we get

$$\overline{P}_{md} = \frac{\lambda^M}{\Gamma(M+1)} \int_{\gamma_1=0}^{\infty}\int_{\gamma_2=0}^{\infty}\cdots\int_{\gamma_Q=0}^{\infty} \frac{1}{\prod_{j=1}^{Q}(1+\gamma_j)^{l_j}} \times \frac{1}{\overline{\gamma}^Q} e^{\frac{-\sum_{j=1}^{Q}\gamma_j}{\overline{\gamma}}} d\gamma_1 d\gamma_2 \ldots d\gamma_Q,$$

which can be reduced to

$$\overline{P}_{md} = \frac{\lambda^M}{\Gamma(M+1)} \prod_{j=1}^{Q} \int_{\gamma_j=0}^{\infty} \frac{1}{(1+\gamma_j)^{l_j}} \frac{1}{\overline{\gamma}} e^{\frac{-\gamma_j}{\overline{\gamma}}} d\gamma_j. \quad (19)$$

It can be easily shown that the integral in (19) is given by

$$\overline{P}_{md} = \frac{\lambda^M}{\Gamma(M+1)} \prod_{j=1}^{Q} \overline{\gamma}^{-l_j} e^{\frac{1}{\overline{\gamma}}} \Gamma\left(1-l_j, \frac{1}{\overline{\gamma}}\right).$$

At large SNR, $e^{\frac{1}{\overline{\gamma}}} \to 1$ and $\Gamma(1-l_j, \frac{1}{\overline{\gamma}}) \asymp \frac{\overline{\gamma}^{l_j-1}}{l_j-1}$ yielding

$$\overline{P}_{md} \asymp \frac{\lambda^M}{\Gamma(M+1)} \times \frac{1}{\overline{\gamma}^Q \prod_{j=1}^{Q}(l_j-1)}. \quad (20)$$

Optimizing the coding gain depends on the choice of the number of samples l_i associated to an antenna realization γ_i. It is obvious from (20) that minimizing the missed detection probability is achieved by maximizing the quantity $\prod_{i=1}^{Q}(l_i-1)$. We can obtain the optimum values of the l_i's via a simple *Lagrange optimization problem* as

$$\max \prod_{i=1}^{Q}(l_i-1)$$

$$\text{s.t.} \sum_{i=1}^{Q} l_i = M.$$

By constructing the auxiliary function $\Theta(l_1,l_2,\ldots,l_Q,\Lambda) = \prod_{i=1}^{Q}(l_i-1) + \Lambda\ (\sum_{i=1}^{Q} l_i - M)$ (where Λ is the lagrange multiplier) and solving for $\bigtriangledown_{(l_1,l_2,\ldots,l_Q)}\Theta(l_1,l_2,\ldots,l_Q,\Lambda) = 0$ (where $\bigtriangledown$ is the gradient operator), we obtain the optimum solution as

$$l_1 = l_2 = \ldots = l_Q = \lfloor \frac{M}{Q} \rfloor.$$

Thus, the optimum antenna switching pattern is to change the antenna radiation mode every $\lfloor \frac{M}{Q} \rfloor$ samples. Note that this result is intuitive as all channel realizations are independent and identically distributed, which means that the optimal antenna mode switching pattern is obtained when employing every mode for an equal time interval during the sensing period.

From (20), the achieved diversity order is

$$d_{md} = -\lim_{\overline{\gamma}\to\infty} \frac{\log \overline{P}_{md}}{\log \overline{\gamma}} = Q.$$

Note that if the number of samples is less than the number of antenna states, only M channel realizations can be employed during the sensing period. Thus, the diversity order is generally given by

$$d_{md} = \min\{M, Q\}.$$

The threshold λ is selected such that $P_F = \alpha$, where it has no impact on the diversity order. The average PU signal energy input to the energy detection is given by $Var\left\{\sum_{j=1}^{L}\sum_{i=l_{j-1}+1}^{l_{j-1}+l_j}\sqrt{\gamma_j}S_i\right\} = \sum_{j=1}^{L}\sum_{i=l_{j-1}+1}^{l_{j-1}+l_j}\overline{\gamma} = M\overline{\gamma}$. Thus, the coding gain is proportional to the total number of samples involved in detection, and the full coding gain is achieved.

2) Spectrum Sensing via State Selection: In the non-cooperative scheme, knowledge of the CSI at the SU can provide neither coding nor diversity gain to the detection performance. In the proposed scheme, the CSI is utilized to *select* the "best" antenna mode (the mode with largest channel gain) rather than *switch* the antenna modes over time. This resembles *selection combining* in multiple antenna systems. Thus, an SNR gain is obtained that is termed as a *selection gain*. The pdf of the maximum of Q Rayleigh distributed channel gains is given by [34]

$$f_{\gamma_{max}}(\gamma_{max}) = \frac{Q}{\overline{\gamma}} e^{\frac{-\gamma_{max}}{\overline{\gamma}}}(1-e^{\frac{-\gamma_{max}}{\overline{\gamma}}})^{Q-1}.$$

In order to simplify the analysis, we focus on the dominant fading density at assymptotically large $\overline{\gamma}$, which can be written as [16]

$$f_{\gamma_{max}}(\gamma_{max}) \approx \frac{Q}{\overline{\gamma}^{Q}} e^{\frac{-\gamma_{max}}{\overline{\gamma}}} \gamma_{max}^{Q-1},$$

and the probability of missed detection as a function of the instantaneous channel gain is obtained from (11) by setting $l_k = M$, where $k = \max_j \gamma_j$, and γ_k is the corresponding channel realization

$$P_{md} = \frac{\Upsilon\left(M, \frac{\lambda}{2(1+\gamma_k)}\right)}{\Gamma(M)}.$$

The average missed detection probability is thus given by

$$\overline{P}_{md} = \frac{Q}{\overline{\gamma}^{Q}} \int_{\gamma_k=0}^{\infty} \frac{\Upsilon\left(M, \frac{\lambda}{2(1+\gamma_k)}\right)}{\Gamma(M)} e^{\frac{-\gamma_k}{\overline{\gamma}}} \gamma_k^{Q-1} d\gamma_k. \quad (21)$$

For simplicity, assume that $1+\gamma_k \approx \gamma_k$. The integrands in (21) can be represented in terms of the Meijer-G function as

$$\overline{P}_{md} = \frac{Q}{\Gamma(M)\overline{\gamma}^{Q}} \int_0^{\infty} \gamma_k^{Q-1} e^{-\frac{\gamma_k}{\overline{\gamma}}} G_{1,2}^{1,1}\left(\begin{matrix} 1 \\ M, 0 \end{matrix} \middle| \frac{\lambda}{2\gamma_k} \right) d\gamma_k.$$

Using the property $G_{p,q}^{m,n}\left(\begin{smallmatrix} a_1,\ldots,a_p \\ b_1,\ldots,b_q \end{smallmatrix} \middle| z \right) = G_{q,p}^{n,m}\left(\begin{smallmatrix} 1-b_1,\ldots,1-b_q \\ 1-a_1,\ldots,1-a_p \end{smallmatrix} \middle| z^{-1} \right)$, the average missed detection probability will be given by the following integral

$$\overline{P}_{md} = \frac{Q}{\Gamma(M)\overline{\gamma}^{Q}} \int_0^{\infty} \gamma_k^{Q-1} e^{-\frac{\gamma_k}{\overline{\gamma}}} G_{2,1}^{1,1}\left(\begin{matrix} 1-M, 1 \\ 0 \end{matrix} \middle| \frac{2\gamma_k}{\lambda} \right) d\gamma_k.$$

Using [27, Eq. (7.813)], the average missed detection probability is

$$\overline{P}_{md} = \frac{Q}{\Gamma(M)} G_{3,1}^{1,2}\left(\begin{matrix} 1-Q, 1-M, 1 \\ 0 \end{matrix} \middle| \frac{2\overline{\gamma}}{\lambda} \right),$$

which can be represented as

$$\overline{P}_{md} = \frac{\mathcal{K}_1}{\overline{\gamma}^{Q}}\ {}_1F_2(Q; Q+1, -M+Q+1; \frac{\lambda}{2\overline{\gamma}})$$

$$+\frac{\mathcal{K}_2}{\overline{\gamma}^{M}}\ {}_1F_2(M; M+1, -M+Q+1; \frac{\lambda}{2\overline{\gamma}}), \quad (22)$$

where $\mathcal{K}_1$ and $\mathcal{K}_2$ are constants, ${}_pF_q(a_1,...,a_p;b_1,...,b_q;z)$ is the generalized hypergeometric function, and ${}_pF_q(a_1,...,a_p;b_1,...,b_q;z) \to 1$ as $z \to 0$. Thus, it can be easily concluded that the diversity order of the state selection scheme will be given by

$$d = \min\{M, Q\}.$$

Note that this is the same diversity order of the state switching scheme. Thus, availability of the CSI at the SU in state selection sensing offers no diversity gain compared to state switching. Selecting the best channel state every sensing period, on the other hand, offers an SNR gain (coding gain) that we define as the *selection gain*. The ratio between the average SNR in the state selection scheme relative to the state switching scheme is given by

$$\text{Selection gain} = \frac{E\{\gamma_k\}}{E\{\gamma\}} = H_Q, \quad (23)$$

where H_Q is the Q^{th} harmonic number defined as $H_Q = 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{Q}$ [34]. For large number of antenna states, the selection gain tends to

$$\text{Selection gain} \approx \log(Q) - \psi(1),$$

where $\psi(.)$ is the *digamma function* and $-\psi(1)$ is the *Euler-Mascheroni* constant. Thus, the coding gain obtained from state selection grows logarithmically with the number of antenna states.

B. Optimal sensing based on Bayes Criterion

1) Spectrum Sensing via State Switching: The achieved diversity order in this case will be obtained according to the following lemma:

Lemma 6: *The achieved diversity order for the proposed scheme using a threshold of $\theta\lambda_{opt}$ is $d_e = \min\{\theta \min\{M, Q\}, \min\{M, Q\}\}$, where $\theta \in \mathbb{R}$.*

Proof See Appendix F.

As stated in Lemma 6, spectrum sensing using a reconfigurable antenna with Q modes can achieve a diversity order of Q. This is equivalent to the diversity order of a cooperative scheme with Q SUs. Even if the SU is using a suboptimal threshold of $\theta\lambda_{opt}$, the achieved diversity order is θQ which is Q times larger than the diversity order achieved by the conventional scheme that employs a threshold of $\theta\lambda_{opt}$.

2) Spectrum Sensing via State Selection: By observing Eq. (17), the missed detection diversity order is given by $\min\{Q, M\}$. The same diversity analysis applied for the state switching scheme in Lemma 6 can be carried out for the state selection scheme. In fact, both schemes have the same diversity order and the same optimal threshold at high SNR. Similar to the NP problem, the state selection scheme offers an extra coding gain as the average SNR is boosted by a factor of H_Q.

C. Impact of Switching Delay

In this subsection, we quantify the impact of switching delay on the detection performance of state switching and state selection schemes. Let D be the equivalent number of samples that a particular switching device needs to change from one antenna state to the other. We assume that throughout those D samples, the old channel realization is perceived by the SU receiver. A new channel realization appears after D samples, which means that the maximum achievable switching rate is $\frac{1}{DT_s}$, where T_s is the system sampling period. [4]

1) Impact on state switching scheme: In the state switching scheme with D delay samples, the achieved diversity order is

$$d = \min\{Q, \frac{M}{D}\}.$$

The SU tries to rapidly switch the antenna modes such that maximum number of channel realizations is utilized in sensing. The limited switching speed affects the achieved diversity order negatively. The number of samples l_j assigned to a channel realization j must be greater than D. The maximum number of channel realizations that can appear within M sensing samples is thus $\frac{M}{D}$. If $Q > \frac{M}{D}$, we can not achieve the maximum diversity order. In fact, if the sensing period is limited compared to the switching delay, the diversity gain offered by reconfigurable antennas becomes less significant. If $M = D$, the system behaves like the conventional non-cooperative scheme.

2) Impact on state selection scheme: If the SU requires D samples to select the maximum channel realization, the achieved SNR gain is perceived for $M - D$ samples only. In this case, the selection gain tends to

$$\text{Selection gain} = \frac{D}{M} + \frac{M - D}{M} H_Q.$$

Moreover, the diversity order is also impacted as the effective sensing period that is subject to the selected channel is $M - D$ samples only. Hence, the diversity order becomes

$$d = \max\{1, \min\{M - D, Q\}\}.$$

Again, at $M = D$, the system acts in an identical way to the legacy single antenna non-cooperative scheme as all samples experience an arbitrary channel without selection. Thus the dominating diversity order is either 1, when $M = D$, or $\min\{M - D, Q\}$ otherwise. The switching delay degrades the diversity order of the state switching scheme, and both the diversity order and selection gain of the state switching scheme. The design of the reconfigurable antenna should take into account the possible values of the sensing period. It is essential to employ high speed switching devices with switching times that are significantly smaller than the sensing period. If the switching speed is inevitably low, one has to extend the sensing period such that diversity and coding gain benefits of the reconfigurable antenna are attained. However, this will be at the expense of the system throughput.

D. Performance Evaluation

In this subsection, we evaluate the performance of the proposed schemes and compare them with the conventional cooperative and non-cooperative schemes. It is important to note that all the parameter settings used in the simulations discussed in this section are selected arbitrarily for numerical and simulation convenience. However, the analyses and explanations presented in the paper are generic and suit any practical values for the system parameters. For all curves, Monte Carlo simulations are carried out with 1,000,000 runs. In Fig. 4, we plot the error probability curves for the non-cooperative, the cooperative (with N being known and unknown), the state switching as well as the state selection (with Q = 15 antenna states) schemes. An overall energy constraint is imposed by fixing the total number of samples to 30. It is shown that the cooperative scheme with 15 cooperating SUs outperforms the non-cooperative scheme at high SNR as it achieves a diversity order of 15. However, cooperation performs worse at SNR values below -5 dB due to the poor coding gain. When the number of SUs is unknown, a diversity order of $\lfloor \frac{15+1}{2} \rfloor$ = 8 is only achieved. Thus the offered diversity gain at high SNR

[4]Various switching devices experience different ranges of time delay. For instance, a MEMS switch may have a switching time of $10-20$ μs [18]. Other electronic switching devices, such as PIN diodes or field-effect transistors (FETs), can offer a much faster switching speed [21].

is generally less than that offered when N is known. State switching and selection are shown to outperform cooperative and non-cooperative schemes at any SNR. For state switching, a diversity order of $\min\{15, 30\}$ = 15 is achieved, which is the same diversity order of the cooperative scheme, leading both curves to have the same slope. However, the state switching scheme uses 30 samples for sensing, which maintains the same coding gain of the non-cooperative scheme. It is shown that state switching acts like a non-cooperative scheme at low SNR, and provides a diversity gain at high SNR. As for the state selection scheme, it attains the same diversity order of $\min\{15, 30\}$ = 15, and in addition, offers a coding gain of $H_{15} = 1 + \frac{1}{2} + \ldots + \frac{1}{15} \approx 5$ dB. Thus, an SNR gain of about 5 dB compared to state switching is obtained via antenna state selection. Similar simulations are carried out for the NP test with 100 samples, false alarm probability of 0.05, and 10 antenna states. Fig. 5 shows that state switching and selection act in a similar manner to that depicted by Fig. 4. Again, state selection scheme outperforms all other schemes, while state switching still offers a better performance than cooperative and non-cooperative schemes. Although achieving the selection gain requires channel estimation and appropriate reconfigurable antenna design (with large number of independent states), it is still less complex than the cooperation scenario.

Fig. 6 demonstrates the impact of switching delay on the sensing performance based on the NP test for Q = 10 states. For the state switching scheme, switching delay has no impact on the coding gain. However, the diversity order is reduced when the delay is introduced. For a total number of sensing samples M = 100, we study the effect of the switching delay with values $D = \{30, 50, 95, 100\}$ samples. For those delay values, the delay-free diversity order of 10 is reduced to be $d_{md} = \min\{10, \frac{100}{30}\} \approx 3$, $\min\{10, \frac{100}{50}\} = 2$, $\min\{10, \frac{100}{95}\} \approx 1$, and $\min\{10, \frac{100}{100}\} = 1$, respectively. This is demonstrated by the degradation of the slope of the solid curves in Fig. 6 as delay increases. When the delay samples are equal to the sensing samples, state switching performs like the non-cooperative scheme with legacy antenna. When a very large delay of 95 samples is encountered, the SU does not achieve any diversity gain (it will be shown later that state selection is less sensitive to large delay scenarios). At low SNR, all curves coincide as switching delay has no impact on the coding gain. Contrarily, the diversity order of the state switching scheme is less sensitive to delay and its coding gain degrades with increasing delay. For a delay of 30 samples, the full diversity order is achieved as $\max\{1, \min\{100 - 30, 10\}\} = 10$. However, the selection gain drops from $H_{10} = 4.667$ dB to $\frac{7}{10}H_{10} + \frac{3}{10} = 3.711$ dB. Similarly, a delay of 50 samples degrades the coding gain but preserves the diversity order. This is depicted in Fig. 6 by the three dashed curves corresponding to delays of D = 0, 30, and 50 samples. The three curves have the same slope (same diversity order) but different coding gains. When the delay becomes as large as 95 samples, the diversity order drops to $\max\{1, \min\{100 - 95, 10\}\} = 5$, which is reflected in Fig. 6 by a significant change in the slope of P_{md}. It is worth mentioning that for 95 delay samples, state switching

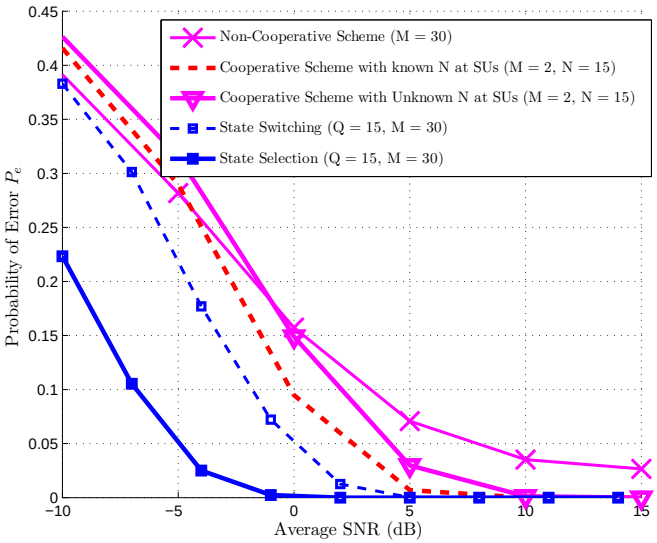


Fig. 3. Performance of various schemes based on the Bayesian test.

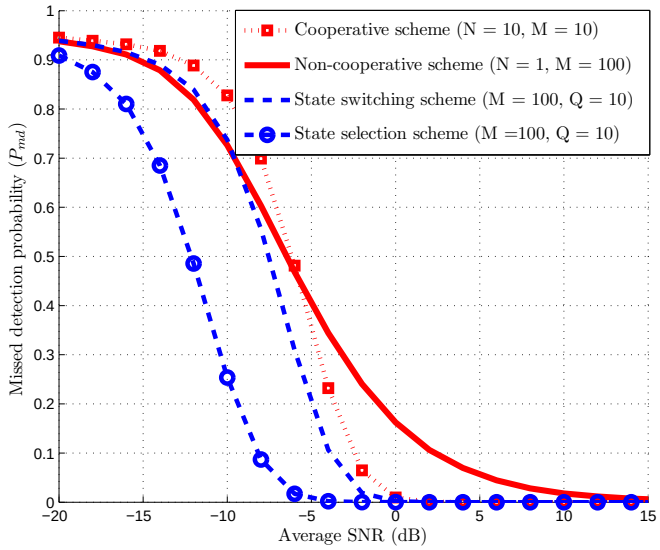


Fig. 4. Performance of various schemes based on NP test with $\alpha = 0.05$

.

does not achieve any diversity gain, which is not the case in state selection. Thus, state selection loses its diversity gain advantages only for significantly large switching delays, but at the expense of the CSI estimation complexity. Fig. 7 shows the impact of delay on the error probability in the Bayesian test, and it is easy to interpret the results in a similar manner.

V. Sensing-throughput trade-off: Throughput Gain in Reconfigurable Antenna Schemes

In this section, we revisit the fundamental tradeoff between sensing capability and achievable throughput of the secondary networks. We will show that there exists an optimal sensing time for which the highest throughput for the secondary network is achieved with sufficient protection for the PU. Next, we will show that by adopting state selection spectrum sensing, this optimal sensing time is reduced, thus allowing for an even

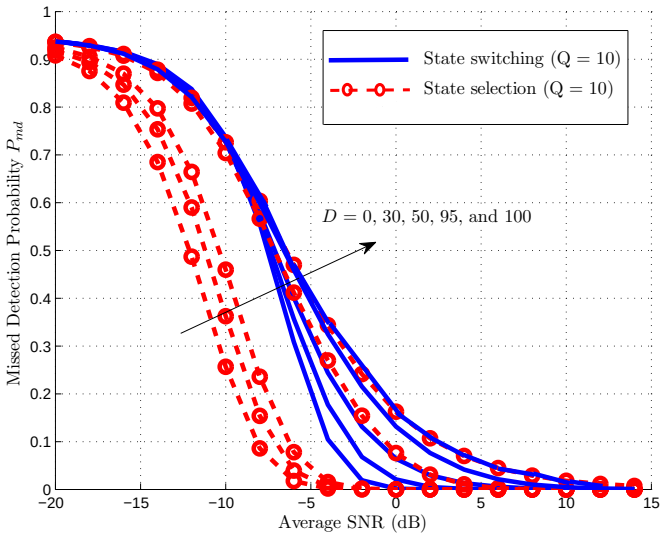


Fig. 5. Impact of switching delay on proposed schemes based on the NP test with M = 100 and $\alpha = 0.05$.

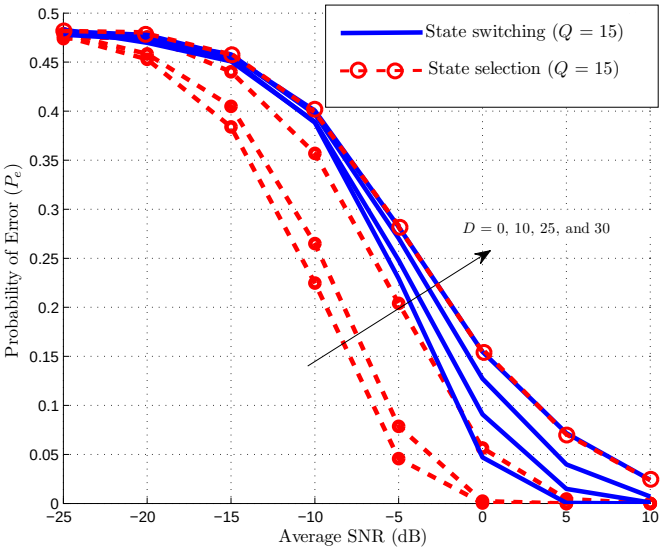


Fig. 6. Impact of switching delay on proposed schemes based on the Bayesian test.

higher throughput given the same PU protection constraints. Furthermore, we show that the SU transmitter and receiver can utilize reconfigurable antennas to maximize the secondary channel capacity by selecting the "best" antenna states at both secondary parties. Thus, not only do reconfigurable antennas improve the performance in the detection phase, but they can also be utilized to enhance the channel capacity in the transmission phase as well. Finally, we investigate the effect of switching delay on the achievable capacity and quantify the possible degradation caused by such delay.

The sensing-throughput tradeoff was studied thoroughly in [23]–[25]. We are concerned here with the impact of reconfigurable antenna spectrum sensing on throughput given a constraint on the detection probability. In the next subsections, we compare the reconfigurable antenna state selection scheme with the conventional one. We omit CSS from our discussion

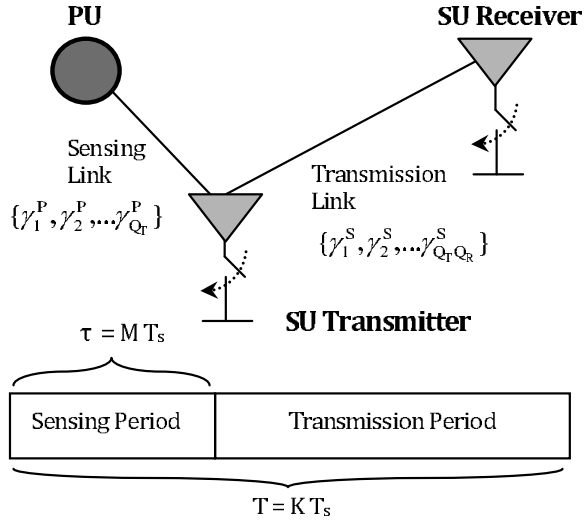


Fig. 7. Sensing and transmission stages in a CR system.

for fair comparison, as the throughput achieved by CSS is divided among the cooperating users. Besides, we only consider state selection and not state switching, as the constraints on detection probability are usually given at low SNR [23], which takes away any advantage of state switching. In addition to that, it is obvious that state switching can not improve the ergodic capacity as it has no CSI.

A. Problem Formulation

As depicted by fig. 8, we assume a frame structured secondary network consisting of an SU transmitter and an SU receiver. The frame is divided into a sensing period of length τ and a transmission period of $T-\tau$. The SU transmitter senses the PU signal for a period of τ and if the PU is absent, the SU transmitter sends data to the SU receiver in a period of $T-\tau$. For a sampling period of T_s, we have $\tau = MT_s$ and $T = KT_s$, where M is the number of samples used in sensing and K is the total number of samples in the frame. We assume that the SU transmitter employs a reconfigurable antenna with Q_T states, while the SU receiver has a reconfigurable antenna with Q_R states. The SU transmitter is engaged in two phases:

- **Sensing phase:** where the SU transmitter senses the PU signal after applying state selection and selects the strongest channel out of the Q_T channel realizations $\{\gamma_1^P, \gamma_2^P, \ldots, \gamma_{Q_T}^P\}$ between the SU transmitter and the PU.
- **Transmission phase:** where the SU transmitter and receiver apply state selection jointly and select the strongest channel out of Q_TQ_R possible channel realizations $\{\gamma_1^S, \gamma_2^S, \ldots, \gamma_{Q_TQ_R}^S\}$.

Thus, the SU transmitter selects the best antenna state for sensing and then switches to the best state for transmission. We assume the availability of full CSI at the SU parties. If switching delay is considered, then D samples are wasted to switch between the different modes.

B. Normalized throughput maximization

The average throughput for the secondary network as a function of the sensing period is given by [23]

$$R(\tau) = \left(1 - \frac{\tau}{T}\right)\left\{C_o P(\mathcal{H}_o)(1-P_F(\tau)) + C_1 P(\mathcal{H}_1) P_{md}(\tau)\right\}. \quad (24)$$

If γ_p is the channel between SU receiver and the PU, and γ_s is the secondary transmission channel, then $C_o = \log(1+\gamma_s)$ and $C_1 = \log(1+\frac{\gamma_s}{1+\gamma_p})$. Because $P(\mathcal{H}_1)$ is usually less than $P(\mathcal{H}_o)$ and $C_1 < C_o$, a reasonable approximation for $R(\tau)$ is adopted in [23]–[24] as

$$R(\tau) \approx C_o P(\mathcal{H}_o)\left(1 - \frac{\tau}{T}\right)(1 - P_F(\tau)). \quad (25)$$

From (25), we note that two factors affect the average secondary throughput. First, as the sensing time increases, the throughput decreases as less time is dedicated to transmission within a frame. Second, a high value for the false alarm probability degrades the throughput as it implies that we waste opportunities to access the channel. The average normalized throughput is defined as $\tilde{R}(\tau) = \frac{R(\tau)}{C_o P(\mathcal{H}_o)}$, which can be expressed as

$$\tilde{R}(\tau) = \frac{T-\tau}{T}(1 - P_F(\tau)).$$

The optimal sensing time is obtained by maximizing $\tilde{R}(\tau)$ while keeping $\overline{P}_D(\tau)$ above a certain threshold

$$\max \quad \tilde{R}(\tau)$$

$$\text{s.t. } \overline{P}_D(\tau) \geq p_d. \quad (26)$$

It is easy to prove that $\tilde{R}(\tau)$ has a unique maximum by proving its unimodality. The derivative of $\tilde{R}(\tau)$ with respect to τ is given by

$$\frac{\partial \tilde{R}(\tau)}{\partial \tau} = \underbrace{\frac{-1}{T}(1 - P_F(\tau))}_{A_1} + \underbrace{(1 - \frac{\tau}{T})\left(-\frac{dP_F(\tau)}{d\tau}\right)}_{A_2}. \quad (27)$$

Notice that the term A_1 is always negative as $P_F(\tau)$ is always less than 1. Also, as $P_F(\tau)$ decreases with increasing τ, then A_1 is a monotonically decreasing function of τ. As for the term A_2, it is always positive because $P_F(\tau)$ is a monotonically decreasing function in τ, which means that $-\frac{dP_F(\tau)}{d\tau}$ is always positive. Moreover, as $\tau < T$, then $(1 - \frac{\tau}{T})$ is also positive and A_2 is positive for all τ. Finally, it can be shown that $-\frac{dP_F(\tau)}{d\tau}$ is a monotonically decreasing function of τ, thus A_2 is also monotonically decreasing in τ. Now, the sum of the two monotonic functions A_1 and A_2 is positive if $|A_2| > |A_1|$ and negative otherwise. Therefore $\tilde{R}(\tau)$ is unimodal and has an extremum point at $|A_2| = |A_1|$.

It is shown in [23] that the optimal solution to (26) is achieved with equality constraint. Assume that for the conventional spectrum sensing scheme, the optimal number of sensing samples is M_{opt}. For this number of samples, the detection probability satisfies the equality constraint $\overline{P}_D(\tau) = p_d$. For state selection spectrum sensing with Q_T antenna states, we have shown that a coding gain of $10\log(H_{Q_T})$ dB is obtained.

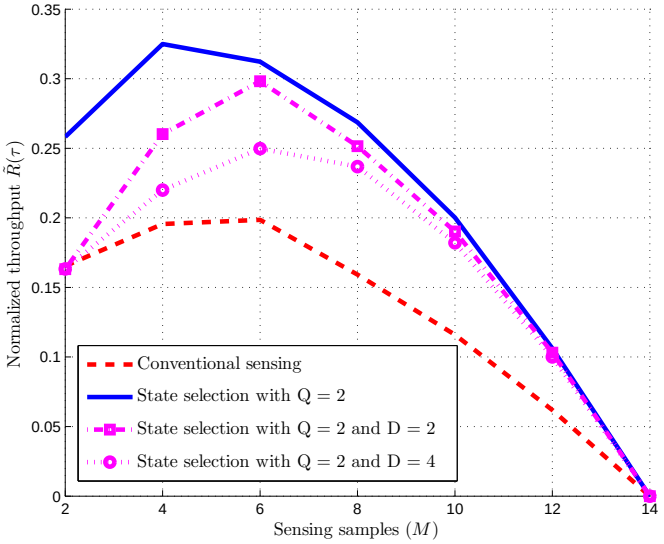


Fig. 8. Optimal sensing time in conventional and state selection schemes (SNR = 0 dB and $\overline{P}_D$ = 0.9).

Thus, to satisfy the constraint of $\overline{P}_D(\tau) = p_d$ with state selection at low SNR, we only need $\frac{M}{H_{Q_T}}$ samples for sensing. If the optimal sensing time for the conventional scheme is M_{opt} and the corresponding false alarm probability is $P_{F,c}$, and if the false alarm probability of the state selection scheme with $\frac{M_{opt}}{H_{QT}}$ sensing samples is $P_{F,s}$, then the normalized throughput gain is

$$\text{Normalized througput gain} = \frac{1 - \frac{M_{opt}}{KH_{QT}}}{1 - \frac{M_{opt}}{K}} \times \frac{1 - P_{F,s}}{1 - P_{F,c}}.$$

Note that $P_{F,s}$ is always less than $P_{F,c}$ for a constant detection probability. The reason for this is that, for a fixed threshold λ, we have $\frac{\Gamma(M_{opt}, \frac{\lambda}{2})}{\Gamma(M_{opt})} > \frac{\Gamma(\frac{M_{opt}}{H_{Q_T}}, \frac{\lambda}{2})}{\Gamma\left(\frac{M_{opt}}{H_{Q_T}}\right)}$ as the false alarm probability is a monotoically decreasing function of the number of sensing samples. In addition to that, the state selection scheme offers a diversity gain, which means that even when the sensing samples are only $\frac{M_{opt}}{H_{QT}}$, the state selection scheme still outperforms the conventional scheme with M_{opt} samples at any SNR. Thus, for a fixed detection probability, the optimal threshold in the state selection scheme is greater than that used in the conventional scheme. Therefore, the false alarm probability is reduced by state selection even if the detection probability is kept constant. This means that by using reconfigurable antennas, a multifaceted throughput gain is achieved. For a fixed detection probability, the optimal sensing time is reduced allowing for longer transmission period, and the false alarm probability is reduced, which in turn, means a better utilization of the channel when the PU is absent.

Fig. 9 depicts the normalized throughput gain obtained by deploying state selection with Q = 2. Assuming that the detection probability is set to 0.9 at an average SNR of 0 dB, the normalized throughput curves for conventional and state selection schemes are plotted versus the number of samples M. It is shown that the optimal sensing time for the conven-

tional scheme is M= 6, which is reduced to 4 in the state selection scheme as the required number of samples to attain the same detection probability becomes $\frac{6}{H_2} = 4$. Besides, the false alarm improvement in the state selection scheme contributes to the total throughput gain. It can be deduced from the peak values that the maximum normalized throughput is boosted from 0.2 to 0.325 when state selection is applied. This gain degrades when switching delay is considered, which is depicted in Fig. 9 for D = 2 and 4. For D = 2, the maximum normalized throughput drops from 0.325 to 0.3, while a delay of D = 4 results in a maximum normalized throughput of 0.25 only.

C. Transmission Channel Capacity

In the previous subsection, we demonstrated the normalized throughput gain achieved by using a reconfigurable antenna in the sensing phase. It is worth mentioning that the SU transmitter can select different antenna states for sensing and transmission to achieve diversity in PU signal detection and SU-to-SU signal transmission. The maximum achievable average throughput is approximated as

$$R = \sup_{1\le i\le Q_T, 1\le j\le Q_R} \left(1-\frac{M}{K}\right) P_F P(\mathcal{H}_o) E\{\log(1+\gamma_{i,j}^S)\},$$

where $\gamma_{i,j}^S$ is the SU transmitter and receiver channel that corresponds to transmitter and receiver antenna states i and j, where $1 \le i \le Q_T$ and $1 \le j \le Q_R$. We drop the term $(1-\frac{M}{K})P_F P(\mathcal{H}_o)$ as it depends on the selected antenna state in the sensing phase. We assume that all possible $Q_T Q_R$ channel realizations are independent and identically distributed (which matches with the conceptual model in Section II), and that the average SNR of the SU link is $\overline{\gamma}_S$. The average (ergodic) transmission channel capacity $E\{\log(1+\gamma_{i,j}^S)\}$ depends on the pdf of the selected antenna state. By selecting the maximum channel out of $Q_T Q_R$ channel realizations, the pdf of $\gamma = \max_{1\le i\le Q_T, 1\le j\le Q_R}\{\gamma_{1,1}^S, \gamma_{1,2}^S, \ldots, \gamma_{1,Q_R}^S, \gamma_{2,1}^S, \ldots, \gamma_{Q_T,Q_R}^S\}$ is given by [34]

$$f_\gamma(\gamma) = \frac{Q_T Q_R}{\overline{\gamma}_S} e^{\frac{-\gamma}{\overline{\gamma}_S}} (1-e^{\frac{-\gamma}{\overline{\gamma}_S}})^{Q_T Q_R - 1},$$

which can be rewritten using the binomial theorem as

$$f_\gamma(\gamma) = Q_T Q_R \sum_{i=0}^{Q_T Q_R - 1} \binom{Q_T Q_R}{i} \frac{(-1)^i}{\overline{\gamma}_S} e^{-\frac{\gamma(i+1)}{\overline{\gamma}_S}}. \quad (28)$$

Thus, the ergodic capacity C_s of the state selection transmission is given by averaging shannon capacity over the pdf in (28)

$$C_s = Q_T Q_R \sum_{i=0}^{Q_T Q_R - 1} \binom{Q_T Q_R}{i} \frac{(-1)^i}{i+1} \int_{\gamma=0}^{\infty} \log(1+\gamma) \frac{e^{-\frac{\gamma(i+1)}{\overline{\gamma}_S}}}{\overline{\gamma}_S/(i+1)} d\gamma. \quad (29)$$

The ergodic capacity of the conventional single antenna scheme is given by $C = e^{\frac{1}{\overline{\gamma}_S}} \mathrm{Ei}\left(\frac{1}{\overline{\gamma}_S}\right)$ [35], where $\mathrm{Ei}(x) = -\int_{-x}^{\infty} \frac{e^{-t}}{t} dt$ is the exponential integral function. Thus, the ergodic capacity of the state selection scheme is given by

$$C_s = Q_T Q_R \sum_{i=0}^{Q_T Q_R - 1} \binom{Q_T Q_R}{i} \frac{(-1)^i}{i+1} e^{\frac{i+1}{\overline{\gamma}_S}} \mathrm{Ei}\left(\frac{i+1}{\overline{\gamma}_S}\right). \quad (30)$$

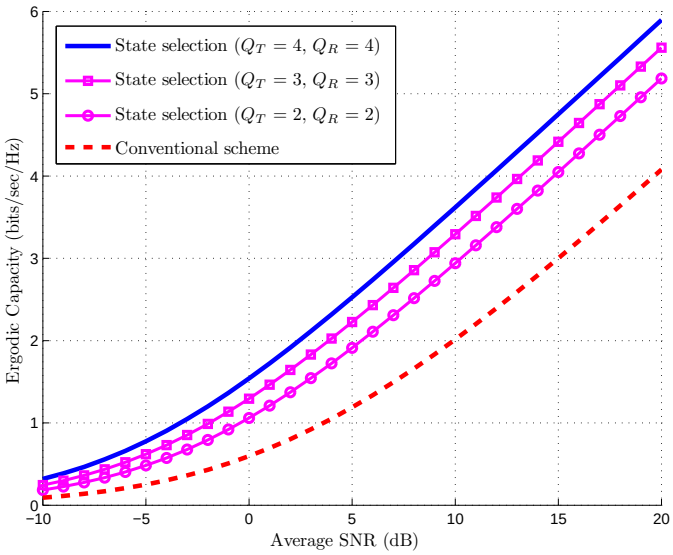


Fig. 9. Capacity gains for various numbers of antenna states.

Assuming that the SU transmitter applies equal power allocation for simplicity, Fig. 10 shows the ergodic capacity gain achieved by state selection for various number of combinations of antenna states. The capacity gain becomes more significant at high SNR. For instance, at an SNR of 10 dB, the capacity of state selection with 4 antenna states is 1.75 times the conventional scheme capacity. This gain can be transformed into an SNR gain of 7.5 dB. In other words, the transmission rate of the conventional scheme at an SNR of 10 dB can be achieved by state selection at an SNR of only 2.5 dB.

For a switching delay of D, the SU transmits on two parallel channels: the channel utilized for sensing is still effective for the first D samples of the transmission period, and the best transmission channel becomes effective for the remaining $K-M-D$ samples. The effective average capacity in this case is given by (30) on top of the next page. Note that when the SU transmits on the previously selected sensing channel for the first D samples, it attains the same capacity of the conventional scheme. Fig. 11 demonstrates the impact of switching delay on the average capacity of state selection with 4 antenna states. When the proportion of switching delay to the total transmission time is 0.2, the capacity gain at SNR = 10 dB reduces from 1.75 to 1.625. Moreover, if the switching delay reaches half of the transmission time, the capacity gain reduces to 1.375. We infer from Fig. 11 that as long as the proportion of the switching delay to the total transmission time is less than 0.2, the SNR loss is less than 1 dB. The effect of switching delay on the achieved capacity depends on the transmission period and the switching technology. An electronic switching device should be adopted if the transmission period is comparable to the switching delay

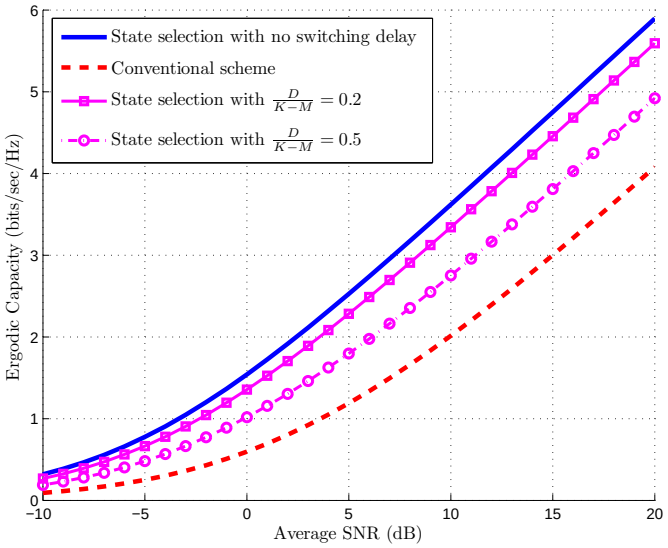


Fig. 10. Impact of switching delay on the average capacity ($Q_T = Q_R = 4$).

of MEMS switches.

VI. Conclusions

In this paper, we discussed a tradeoff between the diversity and coding gains achieved in various spectrum sensing schemes. By obtaining the diversity and coding gains in terms of the detection thresholds, we proved that cooperative schemes are not always beneficial as hard decisions taken at local SUs cause loss of coding gain, which can be significant at low SNR. Based on this analysis, we proposed a novel spectrum sensing scheme that utilizes a reconfigurable antenna at the SU to exploit the diversity of its radiation states, achieving full diversity and coding gains without SU cooperation. The proposed scheme can outperform cooperative sensing, which involves significant overhead, at all SNR ranges. Two schemes based on reconfigurable antennas were presented: state switching and state selection. Based on a conceptual model for the reconfigurable antenna, we obtained the fundamental limits on the achievable diversity order, throughput, and transmission capacity for the proposed schemes. Furthermore, the impact of the state switching delay on the detection performance and the achievable capacity was quantified. It was shown that even with significant switching delay, detection and throughput gains are still attainable.

Appendix A
Proof of Lemma 1

The NP optimization problem is formulated as

$$\max_{\lambda} \ \overline{P}_d(\lambda) \equiv \min_{\lambda} \ \overline{P}_{md}(\lambda)$$
$$\text{s.t. } P_F \leq \alpha,$$

where $\overline{P}_{md}(\lambda)$ is the missed detection probability as a function of the detection threshold. It follows from the definition of the diversity order in Section II that $d_{md} = -\lim_{\overline{\gamma}\to\infty} \frac{\log(\overline{P}_{md}(\lambda))}{\log \overline{\gamma}}$. Note that there is a one-to-one mapping between $f(x)$ and $\log(f(x))$, and that the $\log(\cdot)$ function preserves monotonicity. Thus, maximizing $\overline{P}_{md}(\lambda)$ is equivalent to maximizing $\log(\overline{P}_{md}(\lambda))$. Dividing the objective function by the constant $\log \overline{\gamma}$ yields the equivalent problem

$$\max_{\lambda} \ -\frac{\log(\overline{P}_{md}(\lambda))}{\log \overline{\gamma}}$$
$$\text{s.t. } P_F \leq \alpha. \tag{A.31}$$

It is clear that as $\overline{\gamma} \to \infty$, the optimization problem tends to maximizing the diversity order. This concludes the proof of the lemma.

Appendix B
Proof of Lemma 2

The Bayesian optimization problem is equivalent to minimizing the average probability of error, viz.,

$$\min_{\lambda} \ \overline{P}_e(\lambda) = P(\mathcal{H}_1)\,\overline{P}_{md} + P(\mathcal{H}_o)\,P_F.$$

Recall that the receiver operating characteristics (ROC) (the plot of P_F versus $\overline{P}_D$) is a strictly concave and monotonically increasing function [32], which implies the following

$$\frac{dP_F(\lambda)}{d\overline{P}_D(\lambda)} > 0, \text{ and } \frac{dP_F(\lambda)/d\lambda}{d\overline{P}_D(\lambda)/d\lambda} > 0. \tag{B.32}$$

Because $\frac{dP_F(\lambda)/d\lambda}{d\overline{P}_D(\lambda)/d\lambda}$ is always positive, we deduce that $\frac{dP_F(\lambda)/d\lambda}{d\overline{P}_{md}(\lambda)/d\lambda}$ is always negative. Thus, the derivatives $dP_F(\lambda)/d\lambda$ and $d\overline{P}_{md}(\lambda)/d\lambda$ have opposite signs, i.e., opposite monotonic behaviors. Therefore, we conclude that the average error probability $P_e(\lambda) = P(\mathcal{H}_1)\,\overline{P}_{md} + P(\mathcal{H}_o)\,P_F$ is a unimodal function and the optimal threshold can be obtained by solving the equation

$$\frac{dP_e(\lambda)}{d\lambda} = 0. \tag{B.33}$$

Considering the derivative of $\log(P_e)$ instead of P_e yields

$$\frac{d\log(P_e(\lambda))}{d\lambda} = \frac{1}{P_e(\lambda)}\frac{dP_e(\lambda)}{d\lambda} = 0,$$

which is equivalent to (B.33), thus the Bayesian optimization problem at high SNR reduces to trying to find the threshold λ^* such that

$$\lambda^* = \max_{\lambda} d_e, \tag{B.34}$$

which concludes the proof of the lemma.

Appendix C
Proof of Lemma 3

The average probability of error at high SNR is given by

$$P_e(\lambda) \asymp P(\mathcal{H}_o)\frac{\Gamma(M, \frac{\lambda}{2})}{\Gamma(M)} + P(\mathcal{H}_1)\frac{\lambda}{2\overline{\gamma}(M-1)}. \tag{C.35}$$

Through the second derivative test, it can be easily shown that $P_e(\lambda)$ is concave for $\lambda < 2M$ and convex elsewhere. Thus, $P_e(\lambda)$ has one maximum at λ_{max} and one minimum at λ_{min}. The optimum threshold is λ_{min} and is greater than λ_{max}. The

$$C_{s,D} = \frac{D}{K-M} e^{\frac{1}{\overline{\gamma}_S}} \operatorname{Ei}\left(\frac{1}{\overline{\gamma}_S}\right) + \frac{K-M-D}{K-M} Q_T Q_R \sum_{i=0}^{Q_T Q_R - 1} \binom{Q_T Q_R}{i} \frac{(-1)^i}{i+1} e^{\frac{i+1}{\overline{\gamma}_S}} \operatorname{Ei}\left(\frac{i+1}{\overline{\gamma}_S}\right). \quad (30)$$

maximum and minimum of P_e are obtained by equating $\frac{dP_e}{d\lambda}$ to zero

$$P(\mathcal{H}_o)\frac{-e^{\frac{-\lambda}{2}}\lambda^{M-1}}{2^{M-1}\Gamma(M)} + P(\mathcal{H}_1)\frac{1}{2\overline{\gamma}(M-1)} = 0. \quad (C.36)$$

The solutions of the transcendental Eq. in (C.36) are given by the principal and lower branches of the Lambert W function as [33]

$$\lambda_1 = \mu^{\frac{1}{M-1}} \exp\left(-\mathcal{W}_{-1}\left(\frac{-\mu^{\frac{1}{M-1}}}{2(M-1)}\right)\right),$$
$$\lambda_2 = \mu^{\frac{1}{M-1}} \exp\left(-\mathcal{W}_{o}\left(\frac{-\mu^{\frac{1}{M-1}}}{2(M-1)}\right)\right),$$

where $\mu = \frac{P(\mathcal{H}_1)}{P(\mathcal{H}_o)}\frac{2^{M-2}\Gamma(M-1)}{\overline{\gamma}}$. Given that $-\mathcal{W}_{-1}(x)$ is always greater than $-\mathcal{W}_o(x)$ for $x < 0$, the optimal threshold is simply $\lambda_{opt} = \lambda_1$, which concludes the proof.

Appendix D
Proof of Lemma 4

The series expansion of the Lambert W function is given by [33]

$$\mathcal{W}_{-1}(x) = L_1 - L_2 + \sum_{\ell=0}^{\infty}\sum_{m=1}^{\infty} \frac{(-1)^{\ell}\begin{bmatrix}\ell+m\\ \ell+1\end{bmatrix}}{m!} L_1^{-\ell-m} L_2^m,$$

where $L_1 = \log(-x)$ and $L_2 = \log(-\log(-x))$. As $x \to 0^-$, the first two terms dominate and $\mathcal{W}_{-1}(x) \approx \log(-x) - \log(-\log(-x))$. Thus, from Lemma 3, the optimal threshold can be written as

$$\lambda_{opt} = \mu^{\frac{1}{M-1}} \exp(-L_1 + L_2),$$

which can be expanded as

$$\lambda_{opt} \approx \mu^{\frac{1}{M-1}} \exp\Bigg(-\log\left(\frac{\mu^{\frac{1}{M-1}}}{2(M-1)}\right) + \log\left(-\log\left(\frac{\mu^{\frac{1}{M-1}}}{2(M-1)}\right)\right)\Bigg)$$
$$= 2(M-1)\log\left(\frac{2(M-1)}{\mu^{\frac{1}{M-1}}}\right). \quad (D.37)$$

Thus, as $\overline{\gamma} \to \infty$, and assuming that $P(\mathcal{H}_o) = P(\mathcal{H}_1)$, the optimal threshold can be approximated as

$$\lambda_{opt} \approx 2(M-1)\log\left(\frac{M-1}{\Gamma(M-1)^{\frac{1}{M-1}}}\overline{\gamma}^{\frac{1}{M-1}}\right).$$

The false alarm probability in (2) can be expressed in the series form as $P_F = \sum_{i=0}^{M-1}\frac{\lambda^i}{2^i\Gamma(i+1)}e^{\frac{-\lambda}{2}}$ [6]. At high SNR, the last term in the series representation dominates and $P_F \approx \frac{\lambda^{M-1}}{2^{M-1}\Gamma(M)}e^{\frac{-\lambda}{2}}$. By setting $\lambda = \theta\lambda_{opt}$, the asymptotic false alarm probability is given by

$$P_F \asymp \frac{1}{\Gamma(M)}\left(\theta(M-1)\log\left(\frac{M-1}{\Gamma(M-1)^{\frac{1}{M-1}}}\overline{\gamma}^{\frac{1}{M-1}}\right)\right)^{M-1} \times \left(\frac{\Gamma(M-1)^{\frac{1}{M-1}}}{(M-1)\overline{\gamma}^{\frac{1}{M-1}}}\right)^{\theta(M-1)}$$

and

$$P_{md} \asymp \frac{\theta}{\overline{\gamma}}\log\left(\frac{M-1}{\Gamma(M-1)^{\frac{1}{M-1}}}\overline{\gamma}^{\frac{1}{M-1}}\right). \quad (D.38)$$

Recalling the definitions in Section II, it is straightforward to see that $d_F = \theta$ and $d_{md} = 1$. Thus, the achieved diversity order is given by

$$d_e = \min\{\theta, 1\}.$$

Appendix E
Proof of Lemma 5

The likelihood function is given by

$$\Lambda(r_1, r_2, \ldots, r_M) = \frac{f(r_1, r_2, \ldots, r_M|\mathcal{H}_1)}{f(r_1, r_2, \ldots, r_M|\mathcal{H}_o)}.$$

Based on the signal model presented in Section II, the joint pdf of the sensed samples under hypotheses $\mathcal{H}_1$ and $\mathcal{H}_o$ are

$$f(r_1, r_2, \ldots, r_M|\mathcal{H}_1) = \prod_{i=1}^{M} f(r_i|\mathcal{H}_1)$$
$$= \prod_{i=1}^{M}\frac{1}{\sqrt{2\pi(1+\gamma_{i,j})}}e^{-\frac{r_i^2}{2(1+\gamma_{i,j})}}, \quad (E.39)$$

and

$$f(r_1, r_2, \ldots, r_M|\mathcal{H}_o) = \prod_{i=1}^{M}\frac{1}{\sqrt{2\pi}}e^{-\frac{r_i^2}{2}}. \quad (E.40)$$

By combining (E.39) and (E.40), the Log Likelihood Ratio (LLR) test reduces to

$$\sum_{i=1}^{M}\frac{\gamma_{i,j}}{1+\gamma_{i,j}}|r_i|^2 \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \eta. \quad (E.41)$$

Because the factor $\frac{\gamma_{i,j}}{1+\gamma_{i,j}}$ is constant over every l_j samples and j varies from 1 to Q, we can rewrite the LLR test as

$$\sum_{j=1}^{Q}\frac{\gamma_j}{1+\gamma_j}Z_i \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \eta, \quad (E.42)$$

where $Z_j = \sum_{i=l_{j-1}+1}^{l_{j-1}+l_j}|r_i|^2$ and $l_o = 0$. This concludes the proof of the lemma.

APPENDIX F
PROOF OF LEMMA 6

As stated in Proposition 1, the optimum threshold can be obtained by solving the equation $d_F(\lambda) = d_{md}(\lambda)$ for λ. Unlike the NP test, we do not know how λ_{opt} affects the diversity order as the functional form of λ_{opt} in terms of $\overline{\gamma}$ is unknown. Thus, applying the definition of diversity order in Section II to Eq. (22), we have $d_{md} = \frac{-M\log(\lambda)}{\log(\overline{\gamma})} + \min\{Q, M\}$, where the factor $\min\{Q, M\}$ results from the fact that if $Q > M$, we can switch the antenna modes M times only. The diversity order at large SNR is given by $\frac{-\log(P_F)}{\log(\overline{\gamma})}$. Hence, the error probability diversity order is

$$d_e = \min\left\{\frac{-\log(P_F)}{\log(\overline{\gamma})},\ -\frac{M\log(\lambda)}{\log(\overline{\gamma})} + \min\{Q, M\}\right\}. \tag{F.43}$$

From Proposition 1, we need to find λ_{opt} that satisfies $d_{md}(\lambda) = d_F(\lambda)$, which can be reduced to $\frac{\lambda^{M-1}}{2^{M-1}\Gamma(M)}e^{-\frac{\lambda}{2}} = \lambda^M\overline{\gamma}^{\min\{M,Q\}}$. Thus, similar to the solution of the transcendental equation in Appendix D, the optimum threshold is given by the Lambert W function as

$$\lambda_{opt} = 2\mathcal{W}_o\left(\frac{1}{2\zeta}\right),$$

where $\zeta = \overline{\gamma}^{-\min\{M,Q\}}2^{M-1}\Gamma(M)$. By replacing the Lambert W function with its asymptotic series expansion and considering the dominant terms as shown in Appendix E, the optimum threshold at large SNR is

$$\lambda_{opt} \approx 2\log\left(\frac{\overline{\gamma}^{\min\{M,Q\}}}{2^M\Gamma(M)\log\left(\frac{\overline{\gamma}^{\min\{M,Q\}}}{2^M\Gamma(M)}\right)}\right). \tag{F.44}$$

By substituting λ with $\theta\lambda_{opt}$ in the asymptotic expression of P_F, it is easy to show that $d_F = \theta\min\{Q, M\}$. Besides, it is obvious from (F.44) that $\lim_{\overline{\gamma}\to\infty}\frac{\log(\lambda_{opt})}{\log(\overline{\gamma})} = 0$. Combining this result with (F.43), we have $d_e = \min\{\theta\min\{M, Q\}, \min\{M, Q\}\}$, which concludes the proof.

REFERENCES

[1] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Jrl. Sel. Areas Commun.*, vol. 23, pp. 201–220, Feb. 2005.

[2] Y. Chen, "Analytical Performance of Collaborative Spectrum Sensing Using Censored Energy Detection," *IEEE Trans. Wireless Commun.*, vol. 9, pp. 3856–3865, Dec. 2010.

[3] S. Atapattu, C. Tellambura, and H. Jiang, "Energy Detection Based Cooperative Spectrum Sensing in Cognitive Radio Networks," *IEEE Trans. Wireless Commun.*, vol. 10, pp. 1232–1241, Apr. 2011.

[4] A. Ghasemi and E. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 131–136, Nov. 2005.

[5] Wei Zhang, R. K. Mallik and K. Ben Letaief, "Cooperative Spectrum Sensing Optimization in Cognitive Radio Networks," *Proceedings of IEEE International Conference on Communications (ICC'08)*, pp. 411–415, May 2008.

[6] D. Duan, L. Yang, and J. C. Principe,"Cooperative Diversity of Spectrum Sensing for Cognitive Radio Systems," *IEEE Trans. Sig. Process.*, vol. 58, pp. 3218-3227, Jun. 2010.

[7] Jun Ma, Guodong Zhao and Ye Li, "Soft Combination and Detection for Cooperative Spectrum Sensing in Cognitive Radio Networks," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 4502–4507, Nov. 2008.

[8] Z. Quan, S. Cui, and A. H. Sayed, "An Optimal Strategy for Cooperative Spectrum Sensing in Cognitive Radio Networks," *Proceedings of IEEE Global Telecommunications Conference, (GLOBECOM '07), Washington, DC*, pp. 2947 - 2951, Nov. 2007 .

[9] Z. Quan, S. Cui, and A. H. Sayed, "Optimal Linear Cooperation for Spectrum Sensing in Cognitive Radio Networks," *IEEE J. Sel. Topics Signal Process*, vol. 2, pp. 28–40, Feb. 2008.

[10] S.-J. Kim, E. DallAnese, and G. B. Giannakis, "Cooperative Spectrum Sensing for Cognitive Radios Using Kriged Kalman Filtering," *IEEE Jrl. Sel. Areas Commun.*, vol. 5, pp. 24 – 36, Feb. 2011.

[11] B. Wang, K. J. Ray Liu, and T. C. Clancy, "Evolutionary Cooperative Spectrum Sensing Game: How to Collaborate?," *IEEE Trans. Commun.*, vol. 58, pp. 890–900, Mar. 2010.

[12] S. Maleki and G. Leus, "Censored Truncated Sequential Spectrum Sensing for Cognitive Radio Networks," *IEEE Jrl. Sel. Areas Commun.*, vol. 31, pp. 364–378, Mar. 2013.

[13] T. Cui, F. Gao, and A. Nallanathan, "Optimization of Cooperative Spectrum Sensing in Cognitive Radio," *IEEE Trans. Veh. Technol*, vol. 60, pp. 1578–1589, May 2011.

[14] E. C. Y. Peh, Y. Liang, Y. L. Guan, and Y. Zeng, "Cooperative Spectrum Sensing in Cognitive Radio Networks with Weighted Decision Fusion Schemes," *IEEE Trans. Wireless Commun.*, vol. 9, pp. 3838–3847, Dec. 2010.

[15] E. R.Brown,"RF-MEMS Switches for Reconfigurable Integrated Circuits," *IEEE Trans. Microwave Theory Tech*, vol. 46, pp. 1868 – 1880, Nov. 1998.

[16] T. Gou, C. Wang, and S. A. Jafar, "Aiming Perfectly in the Dark-Blind Interference Alignment Through Staggered Antenna Switching," *IEEE Trans. Sig. Process.*, vol. 59, pp. 2734–2744, Jun. 2011.

[17] Yaxing Cai and Zhengwei Du,"A Novel Pattern Reconfigurable Antenna Array for Diversity Systems," *IEEE Antennas Wireless Propag. Lett.*, vol. 8, pp. 1227–1230, Nov. 2009.

[18] L. Petit, L. Dussopt, and J.-M. Laheurte, "MEMS-Switched Parasitic-Antenna Array for Radiation Pattern Diversity," *IEEE Trans. Antennas Propag.*, vol. 54, pp. 2624 – 2631, Sept. 2006.

[19] D. Piazza, N. J. Kirsch, A. Forenza, R. W. Heath, and K. R. Dandekar, "Design and Evaluation of a Reconfigurable Antenna Array for MIMO Systems," *IEEE Trans. Antennas Propag.*, vol. 54, pp. 869 – 881, Mar. 2008.

[20] A. Forenza and R. W. Heath, "Benefit of Pattern Diversity via Two-Element Array of Circular Patch Antennas in Indoor Clustered MIMO Channels," *IEEE Trans. Commun.*, vol. 54, pp. 943 – 954, May. 2006.

[21] R. Vaughan, "Switched parasitic elements for antenna diversity," *IEEE Trans. Antennas Propagat.*, vol. 47, no. 2, pp. 399-405, Feb. 1999.

[22] P. A. Martin, P. J. Smith, and R. Murch, "Improving Space-Time Code Performance in Slow Fading Channels using Reconfigurable Antennas," *IEEE Commun. Lett.*, vol. 16, pp. 494–497, Apr. 2012.

[23] Y.-C. Liang, Y. Zeng, E. C.Y. Peh, and A. T. Hoang, "Sensing-Throughput Tradeoff for Cognitive Radio Networks," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 1326–1337, Apr. 2008.

[24] S. Stotas and A. Nallanathan, "Overcoming the Sensing-Throughput Tradeoff in Cognitive Radio Networks," *Proceedings of IEEE International Conference on Communications, (ICC '10), Cape Town*, vol. 20, pp. 1 – 5, May 2010.

[25] E. Pei, J. Li, and F. Cheng, "Sensing-throughput Tradeoff for Cognitive Radio Networks with Additional Primary Transmission Protection," *Journal of Computational Information Systems*, vol. 9, pp. 3768 – 3773, May. 2013.

[26] F. F. Digham, M.-S. Alouini, and M. K. Simon, "On the Energy Detection of Unknown Signals Over Fading Channels," *IEEE Trans. Commun.*, vol. 55, pp. 21–24, Jan. 2007.

[27] Alan Jeffrey and Daniel Zwillinger, "Table of Integrals, Series, and Products" *Academic Press*, 2000.

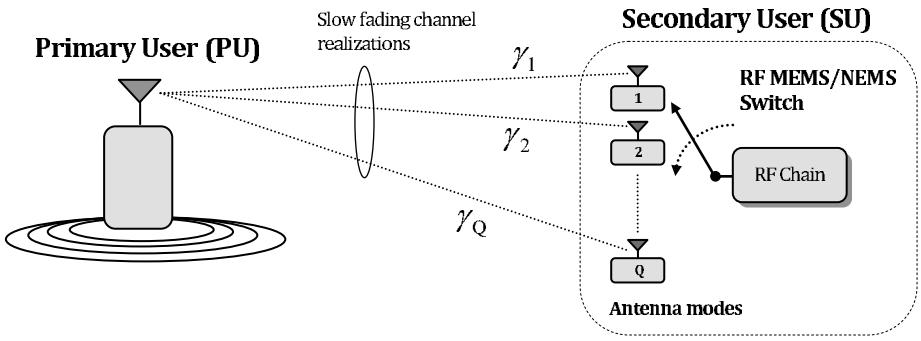
[28] W. Jiang, Z. He, K. Niu, Li Guo, and W. Wu, "Opportunistic Scheduling With BIA Under Block Fading Broadcast Channels," *IEEE Jrl. Sel. Areas Commun.*, vol. 20, pp. 1014 – 1017, Nov. 2013.

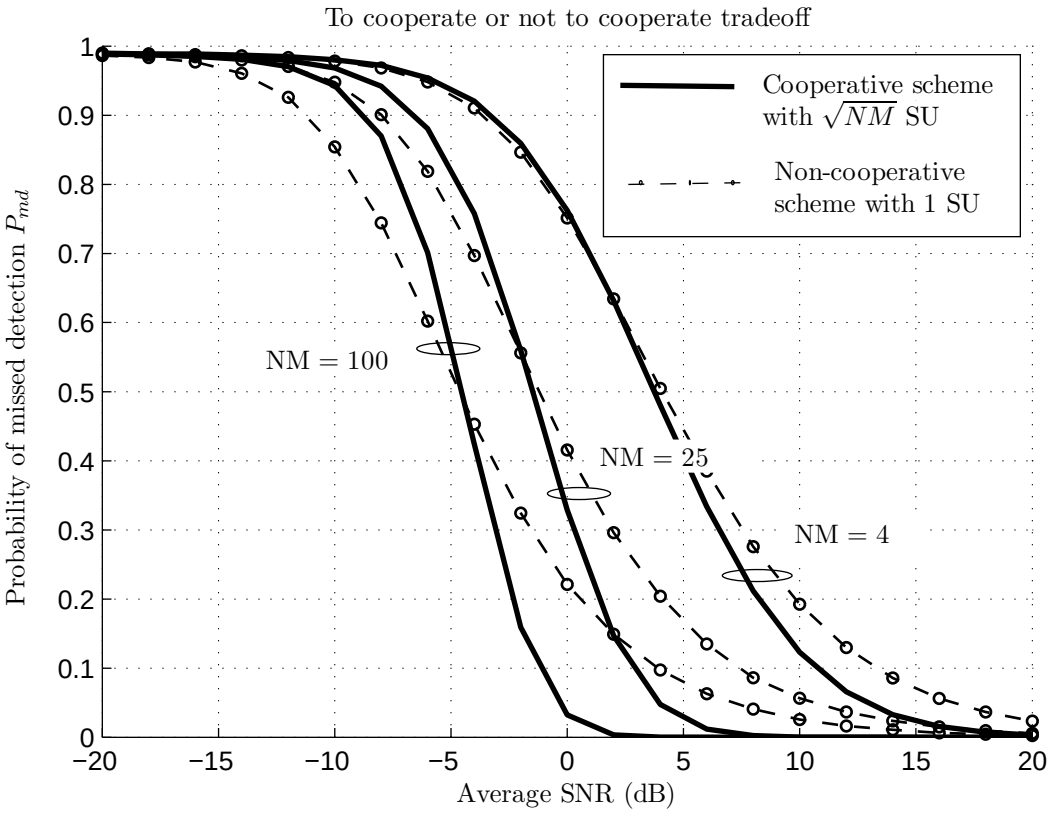
[29] L. Ke and Z. Wang, "Degrees of Freedom Regions of Two-User MIMO Z and Full Interference Channels: The Benefit of Reconfigurable Antennas," *IEEE Trans. Inform. Theory*, vol. 58, pp. 3766 – 3779, Jun. 2012.

[30] P. J. Smith, A. Firag, P. A. Martin, and R. Murch, "SNR Performance Analysis of Reconfigurable Antennas," *IEEE Commun. Lett.*, vol. 16, pp. 498 – 501, Apr. 2012.

[31] J. Hillenbrand, T. A. Weiss, and F. K. Jondral,"Calculation of Detection and False Alarm Probabilities in Spectrum Pooling Systems," *IEEE Commun. Lett.*, vol. 9, pp. 349–351, Apr. 2005.

[32] Harry L.Van Trees,"Detection, Estimation, and Modulation Theory", 2nd Edition, Wiley, 1968.
[33] R. M. Corless, G. H. Gonnet, D. E. Hare , D. J. Jeffrey, "On the Lambert W Function," *Advances In Computational Mathematics*, 1996.
[34] Ning Kong, "Performance Comparison among Conventional Selection Combining, Optimum Selection Combining and Maximal Ratio Combining," *Proceedings of IEEE International Conference on Communications (ICC'09), Dresden, Germany*, pp. 1–6, June 2009.
[35] P. K. Gopala, L. Lai and H. El Gamal, "On the Secrecy Capacity of Fading Channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 10, pp. 4687–4697, Oct. 2008.


Primary User (PU)
Slow fading channel realizations
Secondary User (SU)
γ_1
γ_2
γ_Q
RF MEMS/NEMS Switch
1
2
Q
RF Chain
Antenna modes


To cooperate or not to cooperate tradeoff
Probability of missed detection P_{md}
Average SNR (dB)
Cooperative scheme with $\sqrt{NM}$ SU
Non-cooperative scheme with 1 SU
NM = 100
NM = 25
NM = 4
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
−20
−15
−10
−5
0
5
10
15
20

PLOS ONE

RESEARCH ARTICLE

Interactions of Cultures and Top People of Wikipedia from Ranking of 24 Language Editions

Young-Ho Eom[1], Pablo Aragón[2], David Laniado[2], Andreas Kaltenbrunner[2], Sebastiano Vigna[3], Dima L. Shepelyansky[1]*

1 Laboratoire de Physique Théorique du CNRS, IRSAMC, Université de Toulouse, UPS, F-31062 Toulouse, France, **2** Barcelona Media Foundation, Barcelona, Spain, **3** Dipartimento di Informatica, Università degli Studi di Milano, Milano, Italy

* dima@irsamc.ups-tlse.fr

Citation: Eom Y-H, Aragón P, Laniado D, Kaltenbrunner A, Vigna S, Shepelyansky DL (2015) Interactions of Cultures and Top People of Wikipedia from Ranking of 24 Language Editions. PLoS ONE 10(3): e0114825. doi:10.1371/journal.pone.0114825

Academic Editor: Zhong-Ke Gao, Tianjin University, CHINA

Received: May 30, 2014

Accepted: November 14, 2014

Published: March 4, 2015

Data Availability Statement: All used computational data are publicly available at http://dumps.wikimedia.org/. All the raw data necessary to replicate the findings and conclusion of this study are within the paper, supporting information files and this Wikimedia web site.

Funding: This research is supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE number 288956). No additional external or internal funding was received for this study. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

Wikipedia is a huge global repository of human knowledge that can be leveraged to investigate interwinements between cultures. With this aim, we apply methods of Markov chains and Google matrix for the analysis of the hyperlink networks of 24 Wikipedia language editions, and rank all their articles by PageRank, 2DRank and CheiRank algorithms. Using automatic extraction of people names, we obtain the top 100 historical figures, for each edition and for each algorithm. We investigate their spatial, temporal, and gender distributions in dependence of their cultural origins. Our study demonstrates not only the existence of skewness with local figures, mainly recognized only in their own cultures, but also the existence of global historical figures appearing in a large number of editions. By determining the birth time and place of these persons, we perform an analysis of the evolution of such figures through 35 centuries of human history for each language, thus recovering interactions and entanglement of cultures over time. We also obtain the distributions of historical figures over world countries, highlighting geographical aspects of cross-cultural links. Considering historical figures who appear in multiple editions as interactions between cultures, we construct a network of cultures and identify the most influential cultures according to this network.

Introduction

The influence of digital media on collective opinions, social relationships, and information dynamics is growing significantly with the advances of information technology. On the other hand, understanding how collective opinions are reflected in digital media has crucial importance. Among such a medium, Wikipedia, the open, free, and online encyclopedia, has crucial importance since it is not only the largest global knowledge repository but also the biggest collaborative knowledge platform on the Web. Thanks to its huge size, broad coverage and ease of use, Wikipedia is currently one of the most widely used knowledge references. However, since its beginning, there have been constant concerns about the reliability of Wikipedia because of

Competing Interests: The authors have declared that no competing interests exist.

its openness. Although professional scholars may not be affected by a possible skewness or bias of Wikipedia, students and the public can be affected significantly [1, 2]. Extensive studies have examined the reliability of contents [1–3], topic coverage [4], vandalism [5], and conflict [6–8] in Wikipedia.

Wikipedia is available in different language editions; 287 language editions are currently active. This indicates that the same topic can be described in hundreds of articles written by different language user groups. Since language is one of the primary elements of culture [9], collective cultural biases may be reflected on the contents and organization of each Wikipedia edition. Although Wikipedia adopts a "neutral point of view" policy for the description of contents, aiming to provide unbiased information to the public [10], it is natural that each language edition presents reality from a different angle. To investigate differences and relationships among different language editions, we develop mathematical and statistical methods which treat the huge amount of information in Wikipedia, excluding cultural preferences of the investigators.

Cultural bias or differences across Wikipedia editions have been investigated in previous research [11–17]. A special emphasis was devoted to persons described in Wikipedia articles [12] and their ranking [18, 19]. Indeed, human knowledge, as well as Wikipedia itself, was created by people who are the main actors of its development. Thus it is rather natural to analyze a ranking of people according to the Wikipedia hyper-link network of citations between articles (see network data description below). A cross-cultural study of biographical articles was presented in [20], by building a network of interlinked biographies. Another approach was proposed recently in [21]: the difference in importance of historical figures across Wikipedia language editions is assessed on the basis of the global ranking of Wikipedia articles about persons. This study, motivated by the question "Is an important person in a given culture also important in other cultures?", showed that there are strong entanglements and local biases of historical figures in Wikipedia. Indeed, the results of the study show that each Wikipedia edition favors persons belonging to the same culture (language), but also that there are cross-Wikipedia top ranked persons, who can be signs of entanglement between cultures. These cross-language historical figures can be used to generate inter-culture networks demonstrating interactions between cultures [21]. Such an approach provides us novel insights on cross-cultural differences across Wikipedia editions. However, in [21] only 9 Wikipedia editions, mainly languages spoken in European, have been considered. Thus a broader set of language editions is needed to offer a more complete view on a global scale.

We note that the analysis of persons' importance via Wikipedia becomes more and more popular. This is well visible from the appearance of new recent studies for the English Wikipedia [22] and for multiple languages [23]. The analysis of coverage of researchers and academics via Wikipedia is reported in [24].

Here we investigate interactions and skewness of cultures with a broader perspective, using global ranking of articles about persons in 24 Wikipedia language editions. According to Wikipedia [25] these 24 languages cover 59 percent of world population. Moreover, according to Wikipedia [26], our selection of 24 language editions covers the 68 percent of the total number of 30.9 millions of Wikipedia articles in all 287 languages. These 24 editions also cover languages which played an important role in human history including Western, Asian and Arabic cultures.

On the basis of this data set we analyze spatial, temporal, and gender skewness in Wikipedia by analyzing birth place, birth date, and gender of the top ranked historical figures in Wikipedia. We identified overall Western, modern, and male skewness of important historical figures across Wikipedia editions, a tendency towards local preference (i.e. each Wikipedia edition favors historical figures born in countries speaking that edition's language), and the existence of

global historical figures who are highly ranked in most of Wikipedia editions. We also constructed networks of cultures based on cross-cultural historical figures to represent interactions between cultures according to Wikipedia.

To obtain a unified ranking of historical figures for all 24 Wikipedia editions, we introduce an average ranking which gives us the top 100 persons of human history. To assess the alignment of our ranking with previous work by historians, we compare it with the Hart's list of the top 100 people who, according to him, most influenced human history [27]. We note that Hart "ranked these 100 persons in order of importance: that is, according to the total amount of influence that each of them had on human history and on the everyday lives of other human beings".

Methods

In this research, we consider each Wikipedia edition as a network of articles. Each article corresponds to a node of the network and hyperlinks between articles correspond to links of the network. For a given network, we can define an adjacency matrix A_{ij}. If there is a link (one or more) from node (article) j to node (article) i then $A_{ij} = 1$, otherwise, $A_{ij} = 0$. The out-degree $k_{out}(j)$ is the number of links from node j to other nodes and the in-degree $k_{in}(j)$ is the number of links to node j from other nodes. The links between articles are considered only inside a given Wikipedia edition, there are no links counted between editions. Thus each language edition is analyzed independently from others by the Google matrix methods described below. The transcriptions of names from English to the other 23 selected languages are harvested from WikiData (http://dumps.wikimedia.org/wikidatawiki) and not directly from the text of articles.

To rank the articles of a Wikipedia edition, we use two ranking algorithms based on the articles network structure. Detailed descriptions of these algorithms and their use for Wikipedia editions are given in [18, 19, 28, 29]. The methods used here are described in [21]; we keep the same notations.

Google matrix

First we construct the matrix S_{ij} of Markov transitions by normalizing the sum of the elements in each column of A to unity ($S_{ij} = A_{ij}/\sum_i A_{ij}$, $\sum_i S_{ij} = 1$) and replacing columns with zero elements by elements $1/N$ with N being the matrix size. Then the Google matrix is given by the relation $G_{ij} = \alpha S_{ij} + (1 - \alpha)/N$, where α is the damping factor [30]. As in [21] we use the conventional value $\alpha = 0.85$. It is known that the variation of α in a range $0.5 \leq \alpha < 0.95$ does not significantly affect the probability distribution of ranks discussed below (see e.g. [18, 19, 30]).

PageRank algorithm

PageRank is a widely used algorithm to rank nodes in a directed network. It was originally introduced for Google web search engine to rank web pages of the World Wide Web based on the idea of academic citations [31]. Currently PageRank is used to rank nodes of network systems from scientific papers [32] to social network services [33], world trade [34] and biological systems [35]. Here we briefly outline the iteration method of PageRank computation. The PageRank vector $P(i, t)$ of a node i at iteration t in a network with N nodes is given by

$$P(i,t) = \sum_j G_{ij} P(j, t-1) = (1-\alpha)/N + \alpha \sum_j A_{ij} P(j, t-1)/k_{out}(j). \quad (1)$$

The stationary state $P(i)$ of $P(i, t)$ is the PageRank of node i. More detailed information about the PageRank algorithm is described in [30]. Ordering all nodes by their decreasing probability $P(i)$, we obtain the PageRank ranking index $K(i)$. In qualitative terms, the PageRank probability of a node is proportional to the number of incoming links weighted according to their own probability. A random network surfer spends on a given node a time given on average by the PageRank probability.

CheiRank algorithm

In a directed network, outgoing links can be as important as ingoing links. In this sense, as a complementary to PageRank, the CheiRank algorithm is defined and used in [18, 28, 36]. The CheiRank vector $P*(i, t)$ of a node at iteration time t is given by

$$P^*(i) = (1-\alpha)/N + \alpha \sum_j A_{ji} P^*(j)/k_{in}(j) \quad (2)$$

Same as the case of PageRank, we consider the stationary state $P*(i)$ of $P*(i, t)$ as the CheiRank probability of node i with $\alpha = 0.85$. High CheiRank nodes in the network have large outdegree. Ordering all nodes by their decreasing probability $P*(i)$, we obtain the CheiRank ranking index $K*(i)$. The PageRank probability of an article is proportional to the number of incoming links, while the CheiRank probability of an article is proportional to the number of outgoing links. Thus a top PageRank article is important since other articles refer to it, while a top CheiRank article is highly connected because it refers to other articles.

2DRank algorithm

PageRank and CheiRank algorithms focus only on in-degree and out-degree of nodes, respectively. The 2DRank algorithm considers both types of information simultaneously to rank nodes with a balanced point of view in a directed network. Briefly speaking, nodes with both high PageRank and CheiRank get high 2DRank ranking. Consider a node i which is K_i-th ranked by PageRank and $K*_i$ ranked by CheiRank. Then we can assign a secondary ranking $K'_i = max\{K_i, K^*{}_i\}$ to the node. If $K'_i < K'_j$, then node j has lower 2DRank and vice versa. A detailed illustration and description of this algorithm is given in [18].

We note that the studies reported in [21] show that the overlap between top CheiRank persons of different editions is rather small and due to that the statistical accuracy of this data is not sufficient for determining interactions between different cultures for the CheiRank list. Moreover, CheiRank, based on outgoing links only, selects mainly persons from such activity fields like sports and arts where the historical trace is not so important. Due to these reasons we restrict our study to PageRank and 2DRank. It can be also interesting to use other algorithms of ranking, e.g. LeaderRank [37], but here we restrict ourselves to the methods which we already tested, leaving investigation of other raking methods for further studies.

Data preparation

We consider 24 different language editions of Wikipedia: English (EN), Dutch (NL), German (DE), French (FR), Spanish (ES), Italian (IT), Portuguese (PT), Greek (EL), Danish (DA), Swedish (SV), Polish (PL), Hungarian (HU), Russian (RU), Hebrew (HE), Turkish (TR), Arabic (AR), Persian (FA), Hindi (HI), Malaysian (MS), Thai (TH), Vietnamese (VI), Chinese (ZH), Korean (KO), and Japanese (JA). The Wikipedia data were collected in middle February 2013. The overview summary of each Wikipedia is represented in Table 1.

We understand that our selection of Wikipedia editions does not represent a complete view of all the 287 languages of Wikipedia editions. However, this selection covers most of the

Table 1. Wikipedia hyperlink networks from the 24 considered language editions. Here N_a is the number of articles. Wikipedia data were collected in middle February 2013.

Edition	Language	N_a	Edition	Language	N_a
EN	English	4212493	RU	Russian	966284
NL	Dutch	1144615	HE	Hebrew	144959
DE	German	1532978	TR	Turkish	206311
FR	French	1352825	AR	Arabic	203328
ES	Spanish	974025	FA	Persian	295696
IT	Italian	1017953	HI	Hindi	96869
PT	Portuguese	758227	MS	Malaysian	180886
EL	Greek	82563	TH	Thai	78953
DA	Danish	175228	VI	Vietnamese	594089
SV	Swedish	780872	ZH	Chinese	663485
PL	Polish	949153	KO	Korean	231959
HU	Hungarian	235212	JA	Japanese	852087

doi:10.1371/journal.pone.0114825.t001

largest language editions and allows us to perform quantitative and statistical analysis of important historical figures. Among the 20 largest editions (counted by their size, taken at the middle of 2014) we have not considered the following editions: Waray-Waray, Cebuano, Ukrainian, Catalan, Bokmal-Riksmal, and Finish.

First we ranked all the articles in a given Wikipedia edition by PageRank and 2DRank algorithms, and selected biographical articles about historical figures. To identify biographical articles, we considered all articles belonging to "Category:living people", or to "Category:Deaths by year" or "Category:Birth by year" or their subcategories in the English Wikipedia. In this way, we obtained a list of about 1.1 million biographical articles. We identified birth place, birth date, and gender of each selected historical figure based on DBpedia [38] or a manual inspection of the corresponding Wikipedia biographical article, when for the considered historical figure no DBpedia data were available. We then started from the list of persons with their biographical article's title on the English Wikipedia, and found the corresponding titles in other language editions using the inter-language links provided by WikiData. Using the corresponding articles, identified by the inter-languages links in different language editions, we extracted the top 100 persons from the rankings of all Wikipedia articles of each edition. At the end, for each Wikipedia edition and for each ranking algorithm, we have information about the top 100 historical figures with their corresponding name in the English Wikipedia, their birth place and date, and their gender. All 48 lists of the top 100 historical figures in PageRank and 2DRank for the 24 Wikipedia editions and for the two ranking algorithms are represented in [39] and Supporting Information (SI). The original network data for each edition are available at [39]. The automatic extraction of persons from PageRank and 2DRank listings of articles of each edition is performed by using the above whole list of person names in all 24 editions. This method implies a significantly higher recall compared to the manual selection of persons from the ranking list of articles for each edition used in [21].

We attribute each of the 100 historical figures to a birth place at the country level (actual country borders), to a birth date in year, to a gender, and to a cultural group. Historical figures are assigned to the countries currently at the locations where they were born. The cultural group of historical figures is assigned by the most spoken language of their birth place at the current country level. For example, if someone was born in "Constantinople" in the ancient Roman era, since the place is now Istanbul, Turkey, we assign her/his birth place as "Turkey" and since Turkish is the most spoken language in Turkey, we assign this person to the Turkish

Table 2. List of top persons by PageRank and 2DRank for the English Wikipedia. All names are represented by article titles in the English Wikipedia.

Rank	PageRank persons	2DRank persons
1st	Napoleon	Frank Sinatra
2nd	Barack Obama	Michael Jackson
3rd	Carl Linnaeus	Pope Pius XII
4th	Elizabeth II	Elton John
5th	George W. Bush	Elizabeth II
6th	Jesus	Pope John Paul II
7th	Aristotle	Beyoncé Knowles
8th	William Shakespeare	Jorge Luis Borges
9th	Adolf Hitler	Mariah Carey
10th	Franklin D. Roosevelt	Vladimir Putin

doi:10.1371/journal.pone.0114825.t002

cultural group. If the birth country does not belong to any of the 24 cultures (languages) which we consider, we assign WR (world) as the culture of this person. We would like to point out that although a culture can not be defined only by language, we think that language is a suitable first-approximation of culture. All lists of top 100 historical figures with their birth place, birth date, gender, and cultural group for each Wikipedia edition and for each ranking algorithm are represented in [39]. A part of this information is also reported in SI.

To apply PageRank and 2DRank methods, we consider each edition as the network of articles of the given edition connected by hyper-links among the articles (see the details of ranking algorithms in Section Methods). The full list of considered Wikipedia language editions is given in Table 1. Table 2 represents the top 10 historical figures by PageRank and 2DRank in the English Wikipedia. Roughly speaking, top PageRank articles imply highly cited articles in Wikipedia and top 2DRank articles imply articles which are both highly cited and highly citing in Wikipedia. In total, we identified 2400 top historical figures for each ranking algorithm. However, since some historical figures such as *Jesus*, *Aristotle*, or *Napoleon* appear in multiple Wikipedia editions, we have 1045 unique top PageRank historical figures and 1616 unique top 2Drank historical figures.

We should note that the extraction of persons and their information from a Wikipedia edition is not an easy task even for the English edition, and much more complicated for certain other language editions. Therefore, the above automatic method based on 1.1 million English names and their corresponding names seems to us to be the most adequate approach. Of course, it will miss people who do not have a biographical article on the English Wikipedia. Cross-checking investigation is done for Korean and Russian Wikipedia, which are native languages for two authors, by manually selecting top 100 persons from top lists of all articles ordered by PageRank and 2DRank in both Wikipedia editions. We find that our automatic search misses on average only 2 persons from 100 top persons for these two editions (the missed names are given in SI). The errors appear due to transcription changes of names or missing cases in our name-database based on English Wikipedia. For Western languages the number of errors is presumably reduced since transcription remains close to English. Based on the manual inspection for the Korean and the Russian Wikipedia, we expect that the errors of our automatic recovery of the top people from the whole articles ordered by PageRank and 2DRank are on a level of two percent.

We also note that our study is in compliance with Wikipedia's Terms and Conditions.

Results

Above we described the methods used for the extraction of the top 100 persons in the ranking list of each edition. Below we present the obtained results describing the spatial, temporal and gender distributions of top ranked historical figures. We also determine the global and local persons and obtain the network of cultures based on the ranking of persons from a given language by other language editions of Wikipedia.

Spatial distribution

The birth places of historical figures are attributed to the country containing their geographical location of birth according to the present geographical territories of all world countries. The list of countries appeared for the top 100 persons in all editions is given in Table 3. We also

Table 3. List of country code (CC), countries as birth places of historical figures, and language code (LC) for each country. LC is determined by the most spoken language in the given country. Country codes are based on country codes of Internet top-level domains and language codes are based on language edition codes of Wikipedia; WR represents all languages other than the considered 24 languages.

CC	Country	LC	CC	Country	LC	CC	Country	LC
AE	United Arab Emirates	AR	AF	Afghanistan	FA	AL	Albania	WR
AR	Argentina	ES	AT	Austria	DE	AU	Australia	EN
AZ	Azerbaijan	TR	BE	Belgium	NL	BG	Bulgaria	WR
BR	Brazil	PT	BS	Bahamas	EN	BY	Belarus	RU
CA	Canada	EN	CH	Switzerland	DE	CL	Chile	ES
CN	China	ZH	CO	Colombia	ES	CU	Cuba	ES
CY	Cyprus	EL	CZ	Czech Rep.	WR	DE	Germany	DE
DK	Denmark	DA	DZ	Algeria	AR	EG	Egypt	AR
ES	Spain	ES	FI	Finland	WR	FR	France	FR
GE	Georgia	WR	GR	Greece	EL	HK	Hong Kong	ZH
HR	Croatia	WR	HU	Hungary	HU	ID	Indonesia	WR
IE	Ireland	EN	IL	Israel	HE	IN	India	HI
IQ	Iraq	AR	IR	Iran	FA	IS	Iceland	WR
IT	Italy	IT	JP	Japan	JA	KE	Kenya	EN
KG	Kyrgyzstan	WR	KH	Cambodia	WR	KO	S. Korea	KO
KP	N. Korea	KO	KW	Kuwait	AR	KZ	Kazakhstan	WR
LB	Lebanon	AR	LT	Lithuania	WR	LV	Latvia	WR
LY	Libya	AR	MK	Macedonia	WR	MM	Myanmar	WR
MN	Mongolia	WR	MX	Mexico	ES	MY	Malaysia	MS
NL	Netherlands	NL	NO	Norway	WR	NP	Nepal	WR
NZ	New Zealand	EN	OM	Oman	AR	PA	Panama	ES
PE	Peru	ES	PK	Pakistan	HI	PL	Poland	PL
PS	State of Palestine	AR	PT	Portugal	PT	RO	Romania	WR
RS	Serbia	WR	RU	Russia	RU	SA	Saudi Arabia	AR
SD	Sudan	AR	SE	Sweden	SV	SG	Singapore	ZH
SI	Slovenia	WR	SK	Slovakia	WR	SR	Suriname	NL
SY	Syria	AR	TH	Thailand	TH	TJ	Tajikistan	WR
TN	Tunisia	AR	TR	Turkey	TR	TW	Taiwan	ZH
TZ	Tanzania	WR	UA	Ukraine	WR	UK	United Kingdom	EN
US	United States	EN	UZ	Uzbekistan	WR	VE	Venezuela	ES
VN	Vietnam	VI	XX	Unknown	WR	YE	Yemen	AR
ZA	South Africa	WR						

doi:10.1371/journal.pone.0114825.t003

attribute each country to one of the 24 languages of the considered editions. This attribution is done according to the language spoken by the largest part of population in the given country. Thus e.g. Belgium is attributed to Dutch (NL) since the majority of the population speaks Dutch. If the main language of a country is not among our 24 languages, then this country is attributed to an additional section WR corresponding to the remaining world (e.g. Ukraine, Norway are attributed to WR). If the birth place of a person is not known, then it is also attributed to WR. The choice of attribution of a person to a given country in its current geographic territory, and as a result to a certain language, may have some fluctuations due to historical variations of country borders (e.g. Immanuel Kant was born in the current territory of Russia and hence is attributed to Russian language). However, the number of such cases is small, being on a level of 3.5 percent (see Section "Network of cultures" below). We think that the way in which a link between person, language and country is fixed by the birth place avoids much larger ambiguity of attribution of a person according to the native language which is not so easy to fix in an automatic manner.

The obtained spatial distribution of historical figures of Wikipedia over countries is shown in Fig. 1. This averaged distribution gives the average number of top 100 persons born in a specific country as birth place, with averaging done over our 24 Wikipedia editions. Thus an average over the 24 editions gives for Germany (DE) approximately 9.7 persons in the top 100 of PageRank, being at the first position, followed by USA with approximately 9.5 persons. For

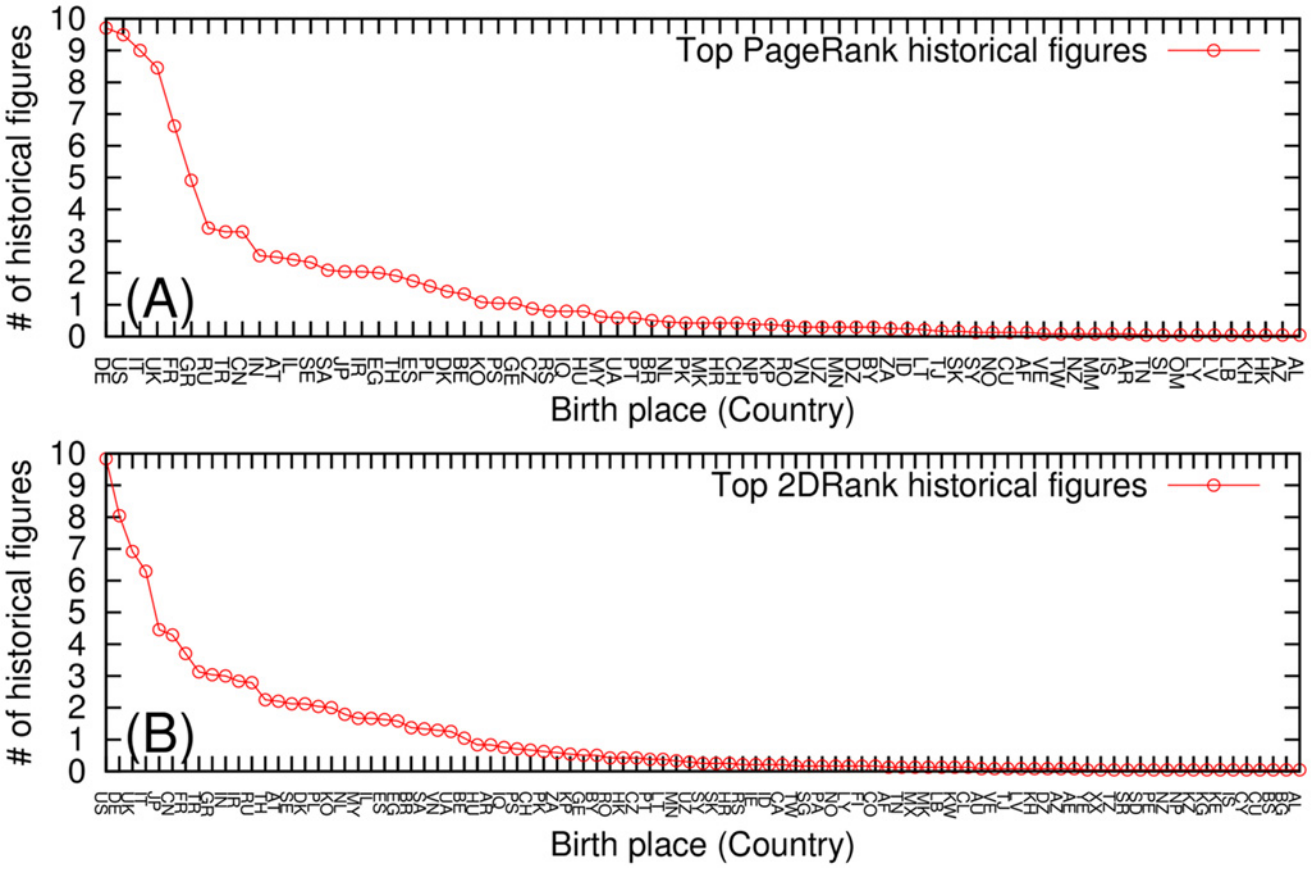


Fig 1. Birth place distribution of top historical figures averaged over 24 Wikipedia edition for (A) PageRank historical figures (71 countries) and (B) 2DRank historical figures (91 countries). Two letter country codes are represented in Table 3.

doi:10.1371/journal.pone.0114825.g001

2DRank we have USA at the first position with an average of 9.8 persons and Germany at the second with an average of 8.0 persons.

Western (Europe and USA) skewed patterns are observed in both top PageRank historical figures (Fig. 1. (A)) and top 2DRank historical figures (Fig. 1. (B)). This Western skewed pattern is remarkable since 11 Wikipedia editions of the 24 considered editions are not European language editions. Germany, USA, Italy, UK and France are the top five birth places of top PageRank historical figures among 71 countries. On the other hand, USA, Germany, UK, Italy and Japan are top five birth places of the top 2DRank historical figures among 91 countries.

In Fig. 2 we show the world map of countries, where color indicates the number of persons from a given country among the 24×100 top persons for PageRank and 2DRank. Additional figures showing these distributions for different centuries are available at [39].

We also observed local skewness in the spatial distribution of the top historical figures for the PageRank (2DRank) ranking algorithm as shown in Fig. 3A (in Fig. 3B). For example, 47 percent of the top PageRank historical figures in the English Wikipedia were born in USA (25 percent) and UK (22 percent) and 56 percent of the top historical figures in the Hindi Wikipedia were born in India. A similar strong locality pattern of the top historical figures was observed in our previous research [21]. However it should be noted that in the previous study we considered the native language of the top historical figure as a criterion of locality, while in the current study we considered 'birth place' as criterion of locality.

Regional skewness, the preferences of Wikipedia editions for historical figures who were born in geographically or culturally related countries, is also observed. For example, 18 (5) of the top 100 PageRank historical figures in the Korean (Japanese) Wikipedia were born in China. Also 9 of the top 100 PageRank historical figures in the Persian Wikipedia were born in Saudi Arabia. The distribution of top persons from each Wikipedia edition over world countries is shown in Fig. 3A and Fig. 3B. The countries on a horizontal axis are grouped by clusters of corresponding language so that the links inside a given culture (or language) become well visible.

To observe patterns in a better way at low numbers of historical figures, we normalized each column of Fig. 3A and Fig. 3B corresponding to a given country. In this way we obtain a rescaled distribution with better visibility for each birth country level as shown in Fig. 3C and Fig. 3D, respectively. We can observe a clear birth pattern of top PageRank historical figures born in Lebanon, Libya, Oman, and Tunisia in the case of the Arabic Wikipedia, and historical figures born in N. Korea appearing not only in the Korean but also in the Japanese Wikipedia.

In the case of the top 2DRank historical figures shown in Fig. 3B and Fig. 3D, we observe overall patterns of locality and regions being similar to the case of PageRank, but the locality is stronger.

In short, we observed that most of the top historical figures in Wikipedia were born in Western countries, but also that each edition shows its own preference to the historical figures born in countries which are closely related to the corresponding language edition.

Temporal distribution

The analysis of the temporal distribution of top historical figures is done based on their birth dates. As shown in Fig. 4A for PageRank, most of historical figures were born after the 17th century on average, which shows similar pattern with world population growth [40]. However, there are some distinctive peaks around BC 5th century and BC 1st century for the case of PageRank because of Greek scholars (*Socrates, Plato*, and *Herodotus*), Roman politicians (*Julius Caesar, Augustus*) and Christianity leaders (*Jesus, Paul the Apostle*, and *Mary (mother of Jesus)*). We also observe that the Arabic and the Persian Wikipedia have more historical figures

Fig 2. Sum of appearances of historical figures from a given country in the 24 lists of top 100 persons for PageRank (top panel) and 2DRank (bottom panel). Color changes from zero (white) to maximum (black). Maximal values are 233 appearances for Germany (top) and 236 for USA (bottom). Values are proportional to the averages per country shown in Fig. 1.

doi:10.1371/journal.pone.0114825.g002

than Western language Wikipedia editions from AD 6th century to AD 12th century. For the case of 2DRank in Fig. 4B, there is only one small peak around BC 1C, which is also smaller than the peak in the case of PageRank, and all the distribution is dominated by a strong growth on the 20th century.

The distributions of the top PageRank historical figures over the 24 Wikipedia editions for each century are shown in Fig. 4C. The same distribution, but normalized to unity over all

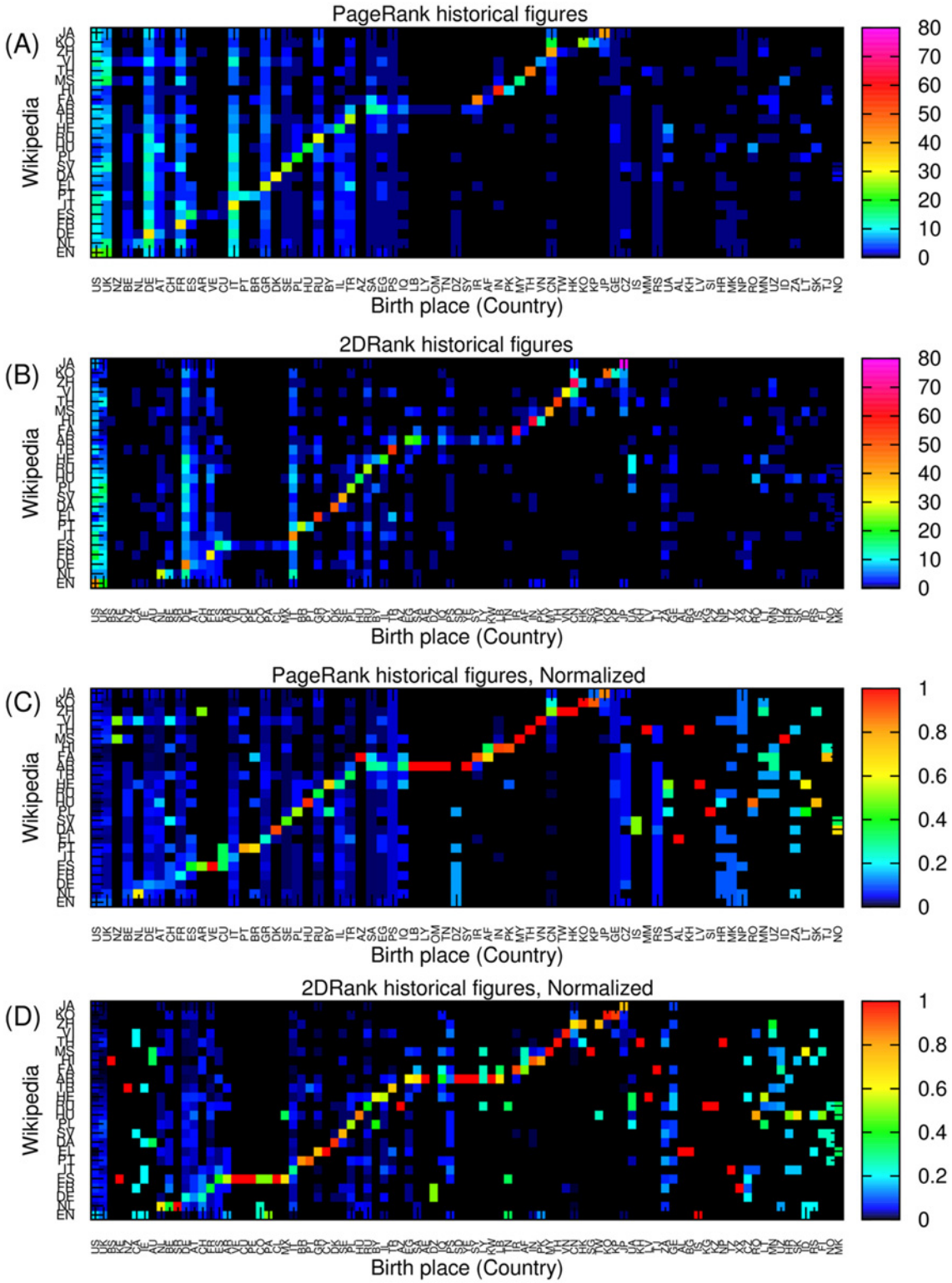


Fig 3. Birth place distributions over countries of top historical figures from each Wikipedia edition; two letter country codes are represented in Table 3. Panels: (A) distributions of PageRank historical figures over 71 countries for each Wikipedia edition; (B) distributions of 2DRank historical figures over 91 countries for each Wikipedia edition; (C) column normalized birth place distributions of PageRank historical figures of panel (A); (D) column normalized birth place distributions of 2DRank historical figures of panel (B).

doi:10.1371/journal.pone.0114825.g003

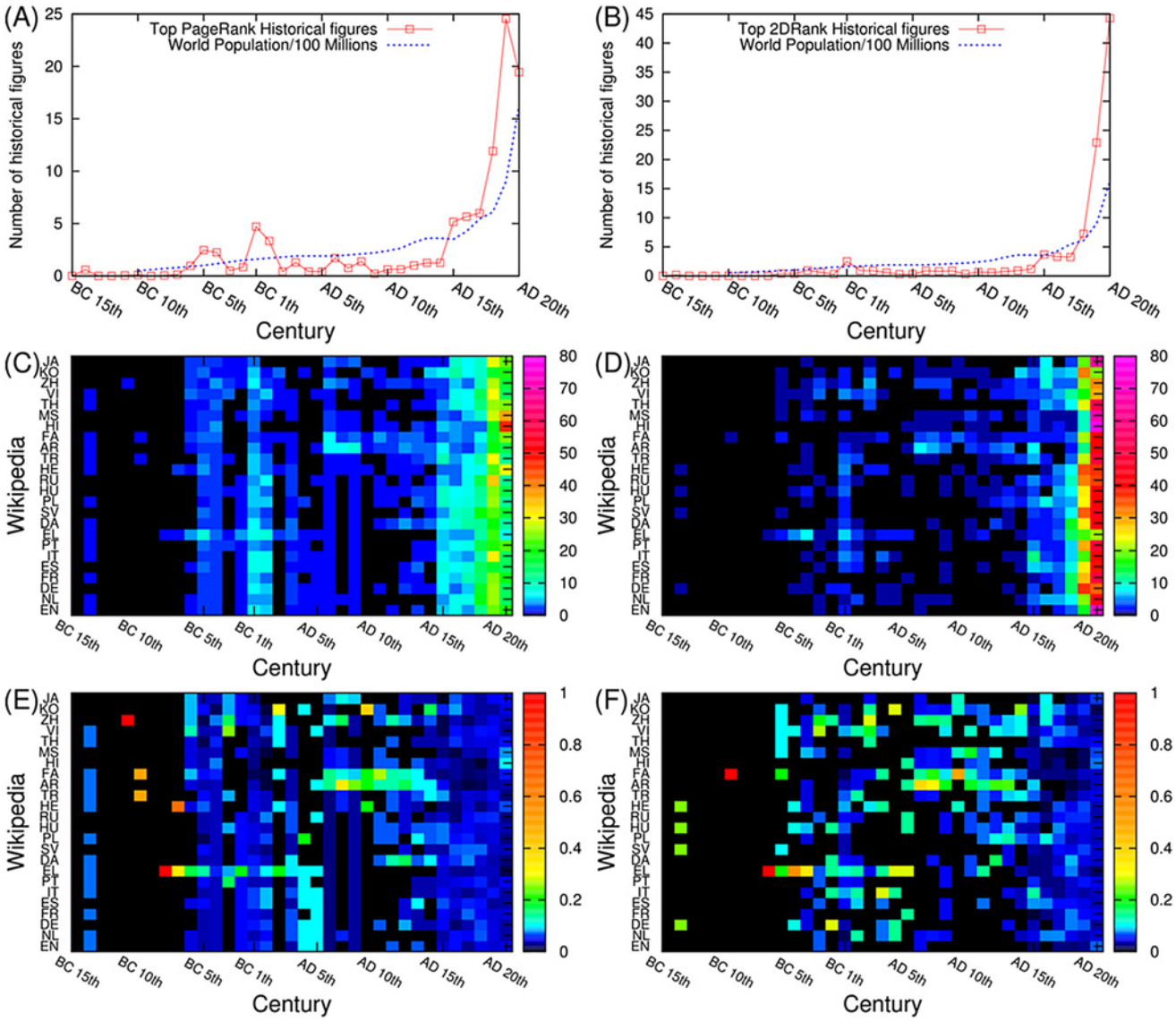


Fig 4. Birth date distributions of top historical figures. (A) Birth date distribution of PageRank historical figures averaged over 24 Wikipedia editions (B) Birth date distribution of 2DRank historical figures averaged over 24 Wikipedia editions (C) Birth date distributions of PageRank historical figures for each Wikipedia edition. (D) Birth date distributions of 2DRank historical figures for each Wikipedia edition. (E) Column normalized birth date distributions of PageRank historical figures for each Wikipedia edition. (F) Column normalized birth date distributions of 2DRank historical figures for each Wikipedia edition.

doi:10.1371/journal.pone.0114825.g004

editions for each century, is shown in Fig. 4E. The Persian (FA) and the Arabic (AR) Wikipedia have more historical figures than other language editions (in particular European language editions) from the 6th to the 12th century due to Islamic leaders and scholars. On the other hand, the Greek Wikipedia has more historical figures in BC 5th century because of Greek philosophers. Also most of western-southern European language editions, including English, Dutch, German, French, Spanish, Italian, Portuguese, and Greek, have more top historical figures because they have *Augustine the Hippo* and *Justinian I* in common. Similar distributions obtained from 2DRank are shown in Fig. 4D and Fig. 4F respectively.

The data of Figs. 4E, F clearly show well pronounced patterns, corresponding to strong interactions between cultures: from BC 5th century to AD 15th century for JA, KO, ZH, VI; from

AD 6th century to AD 12th century for FA, AR; and a common birth pattern in EN, EL, PT, IT, ES, DE, NL (Western European languages) from BC 5th century to AD 6th century. In supporting Figure S1 we show distributions of historical figures over languages according to their birth place. In this case the above patterns become even more pronounced.

At a first glance from Figs. 4E, F we observe for persons born in AD 20th century a significantly more homogeneous distribution over cultures compared to early centuries. However, as noted in [21], each Wikipedia edition favors historical figures speaking the corresponding language. We investigate how this preference to same-language historical figures changes in time. For this analysis, we define two variables $M_{L,C}$ and $N_{L,C}$ for a given language edition L and a given century C. Here $M_{L,C}$ is the number of historical figures born in all countries being attributed to a given language L, and $N_{L,C}$ is the total number of historical figures for a given century C and a given language edition L. For example, among the 21 top PageRank historical figures from the English Wikipedia, who were born in AD 20th century, two historical figures (Pope John Paul II and Pope Benedict XVI) were not born in English speaking countries. Thus in this case $N_{EN,20} = 21$ and $M_{EN,20} = 19$. Fig. 5 represents the ratio $r_{L,C} = M_{L,C}/N_{L,C}$ for each edition and each century. In ancient times (i.e. before AD 5th century), most historical figures for each Wikipedia edition are not born in the same language region except for the Greek, Italian, Hebrew, and Chinese Wikipedia. However, after AD 5th century, the ratio of same language historical figures is rising. Thus, in AD 20th century, most Wikipedia editions have significant numbers of historical figures born in countries speaking the corresponding language. For PageRank persons and AD 20th century, we find that the English edition has the largest fraction of its own language, followed by Arabic and Persian editions while other editions have significantly large connections with other cultures. For the English edition this is related to a significant number of USA presidents appearing in the top 100 list (see [18, 19]). For 2DRank persons the largest fractions were found for Greek, Arabic, Chinese and Japanese cultures. These data show that even in age of globalization there is a significant dominance of local historical figures for certain cultures.

Gender distribution

From the gender distributions of historical figures, we observe a strong male-skewed pattern across many Wikipedia editions regardless of the ranking algorithm. On average, 5.2(10.1) female historical figures are observed among the 100 top PageRank (2DRank) persons for each Wikipedia edition. Fig. 6 shows the number of top female historical figures for each Wikipedia edition. Thai, Hindi, Swedish, and Hebrew have more female historical figures than the average over our 24 editions in the case of PageRank. On the other hand, the Greek and the Korean versions have a lower number of females than the average. In the case of 2DRank, English, Hindi, Thai, and Hungarian Wikipedia have more females than the average while German, Chinese, Korean, and Persian Wikipedia have less females than the average. In short, the top historical figures in Wikipedia are quite male-skewed. This is not surprising since females had little chance to be historical figures for most of human history. We compare the gender skewness to other cases such as the number of female editors in Wikipedia (9 percent) in 2011 [41] and the share of women in parliaments, which was 18.7 percent in 2012 by UN Statistics and indicators on women and men [42], the male skewness for the PageRank list is stronger in the contents of Wikipedia [43]. However, the ratio of females among the top historical figures is growing by time as shown in Fig. 6 C. It is notable that the peak in Fig. 6C at BC 1st is due to "Mary (mother of Jesus)". In the 20th century 2DRank gives a larger percentage of women compared to PageRank. This is due to the fact that 2DRank has a larger fraction of singers and artists comparing to PageRank (see [18, 19]) and that the fraction of women in these fields of activity is larger.

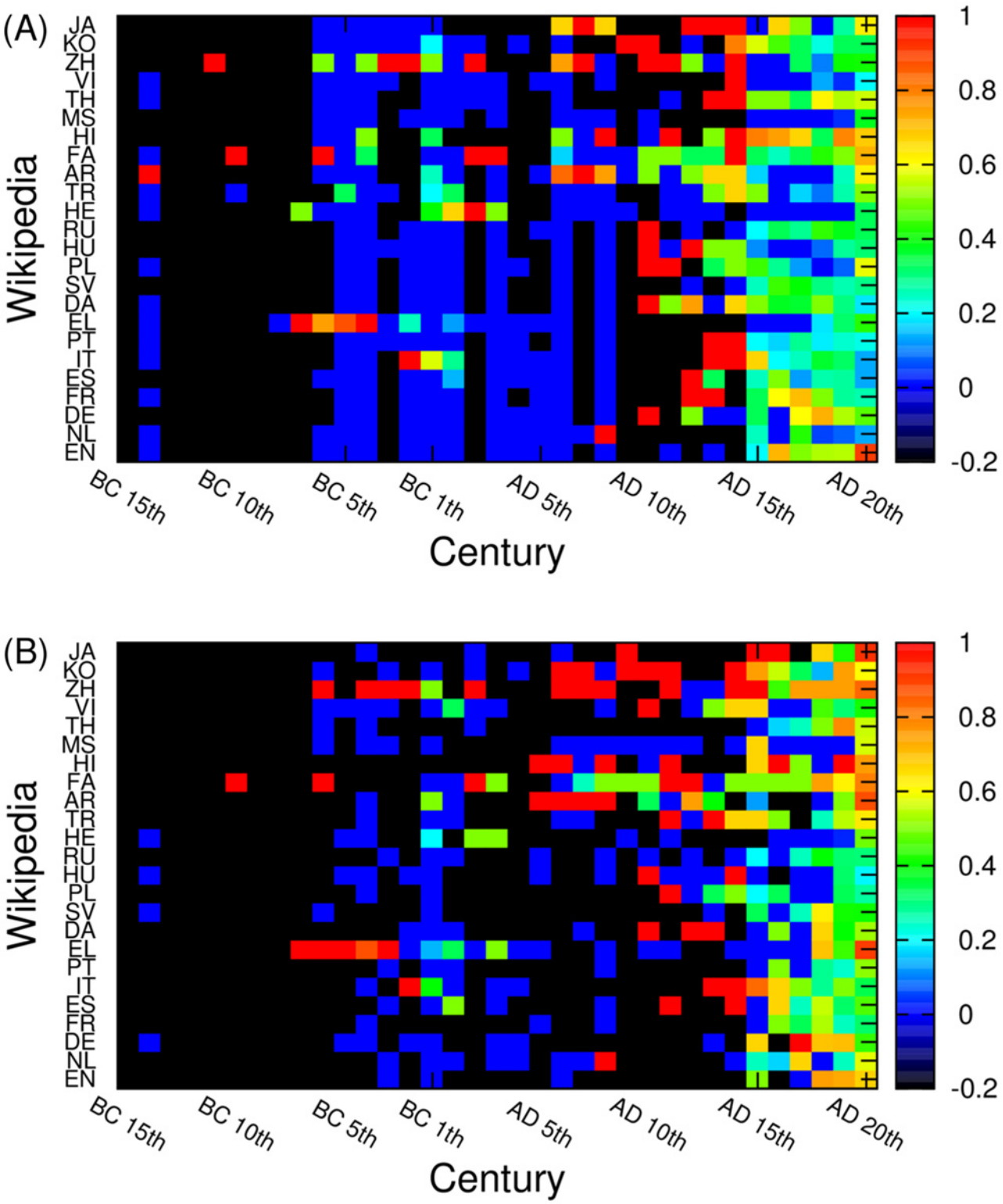


Fig 5. The locality property of cultures represented by the ratio $r_{L,C} = M_{L,C}/N_{L,C}$ for each edition *L* and each century *C*. Here $M_{L,C}$ is the number of historical figures born in countries attributed to a given language edition *L* at century *C* and $N_{L,C}$ is the total number of historical figures in a given edition at a given century, regardless of language of their birth countries. Black color (-0.2 in the color bars) shows that there is no historical figure at all for a given edition and century; blue (0 in the color bars) shows there there are some historical figures but no same language historical figures. Here (A) panel shows PageRank historical figures, and (B) panel shows 2DRank historical figures.

doi:10.1371/journal.pone.0114825.g005

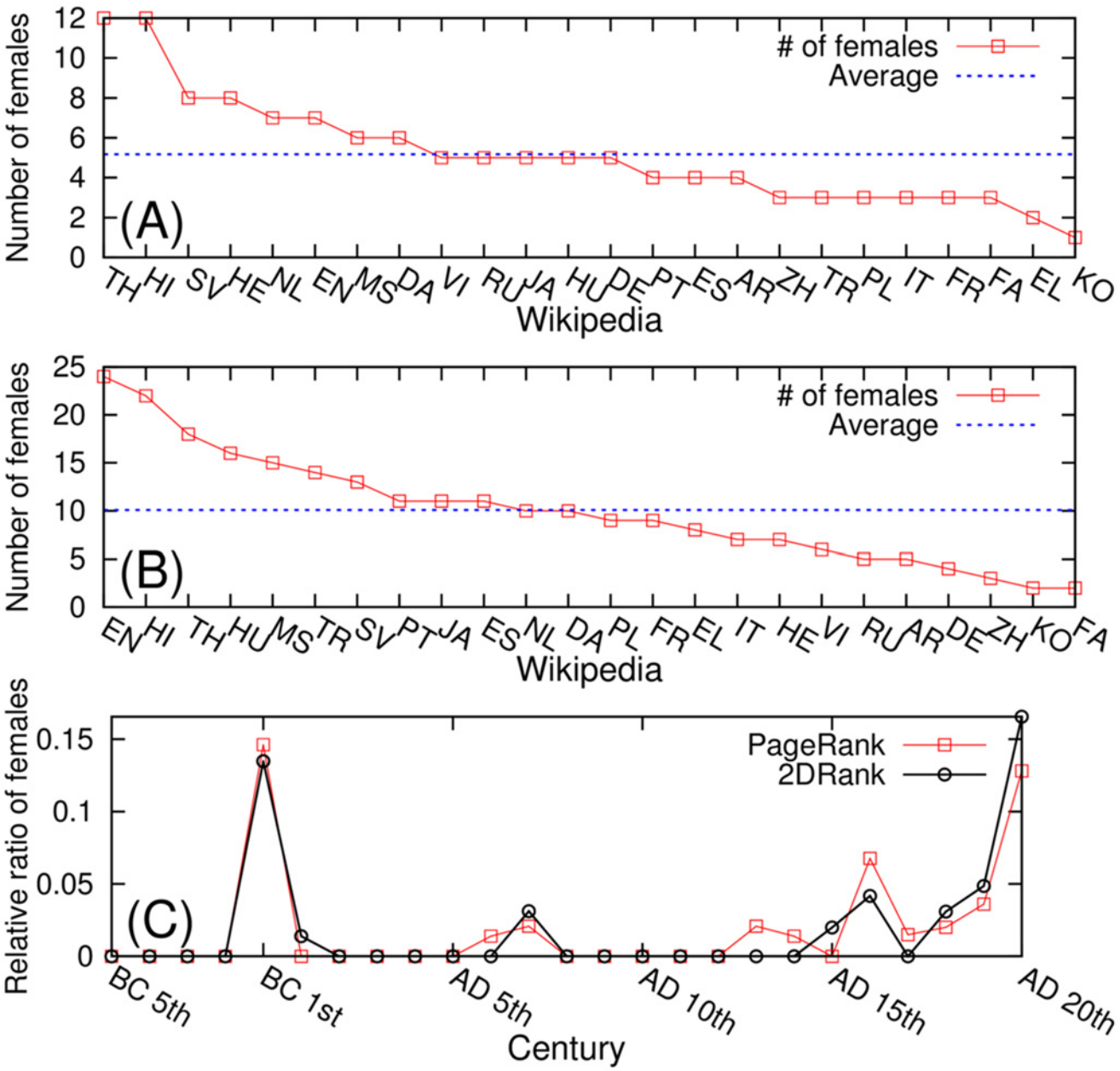


Fig 6. Number of females of top historical figures from each Wikipedia edition (A) Top PageRank historical figures (B) Top 2DRank historical figures. (C) The average female ratio of historical figures in given centuries across 24 Wikipedia editions.

doi:10.1371/journal.pone.0114825.g006

Global historical figures

Above we analyzed how top historical figures in Wikipedia are distributed in terms of space, time, and gender. Now we identify how these top historical figures are distributed in each Wikipedia edition and which are global historical figures. According to previous research [21], there are some global historical figures who are recognized as important historical figures across Wikipedia editions. We identify global historical figures based on the ranking score for a

Table 4. List of global historical figures by PageRank and 2DRank for all 24 Wikipedia editions. All names are represented by the corresponding article titles in the English Wikipedia. Here, Θ_A is the ranking score of algorithm A (3); N_A is the number of appearances of a given person in the top 100 rank for all editions.

Rank	PageRank global figures	Θ_{PR}	N_A	2DRank global figures	Θ_{2D}	N_A
1st	Carl Linnaeus	2284	24	Adolf Hitler	1557	20
2nd	Jesus	2282	24	Michael Jackson	1315	17
3rd	Aristotle	2237	24	Madonna (entertainer)	991	14
4th	Napoleon	2208	24	Jesus	943	14
5th	Adolf Hitler	2112	24	Ludwig van Beethoven	872	14
6th	Julius Caesar	1952	23	Wolfgang Amadeus Mozart	853	11
7th	Plato	1949	24	Pope Benedict XVI	840	12
8th	William Shakespeare	1861	24	Alexander the Great	789	11
9th	Albert Eistein	1847	24	Charles Darwin	773	12
10th	Elizabeth II	1789	24	Barack Obama	754	16

doi:10.1371/journal.pone.0114825.t004

given person determined by her number of appearances and ranking index over our 24 Wikipedia editions.

Following [21], the ranking score $\Theta_{P,A}$ of a historical figure P is given by

$$\Theta_{P,A} = \sum_{E} (101 - R_{P,E,A}) \quad (3)$$

Here $R_{P,E,A}$ is the ranking of a historical figure P in Wikipedia edition E by ranking algorithm A. According to this definition, a historical figure who appears more often in the lists of top historical figures for the given 24 Wikipedia editions or has higher ranking in the lists gets a higher ranking score. Table 4 represents the top 10 global historical figures for PageRank and 2DRank. *Carl Linnaeus* is the 1st global historical figure by PageRank followed by *Jesus, Aristotle. Adolf Hitler* is the 1st global historical figure by 2DRank followed by *Michael Jackson, Madonna (entertainer)*. On the other hand, the lists of the top 10 local historical figures ordered by our ranking score for each language are represented in supporting Tables S1–S25 and [39].

The reason for a somewhat unexpected PageRank leader *Carl Linnaeus* is related to the fact that he laid the foundations for the modern biological naming scheme so that plenty of articles about animals, insects and plants point to the Wikipedia article about him, which strongly increases the PageRank probability. This happens for all 24 languages where *Carl Linnaeus* always appears on high positions since articles about animals and plants are an important fraction of Wikipedia. Even if in a given language the top persons are often politicians (e.g. *Napoleon, Barak Obama* at $K = 1, 2$ in EN), these politicians have mainly local importance and are not highly ranked in other languages (e.g. in ZH *Carl Linnaeus* is at $K = 1$, *Napoleon* at $K = 3$ and *Barak Obama* is at $K = 24$). As a result when the global contribution is counted over all 24 languages *Carl Linnaeus* appears on the top PageRank position.

Our analysis suggests that there might be three groups of historical figures. Fig. 7 shows these three groups of top PageRank historical figures in Wikipedia: (i) global historical figures who appear in most of Wikipedia editions ($N_A \geq 18$) and are highly ranked ($\langle K \rangle \leq 50$) for each Wikipedia such as Carl Linnaeus, Plato, Jesus, and Napoleon (Right-Top of the Fig. 7A); (ii) local-highly ranked historical figures who appear in a few Wikipedia editions ($N_A < 18$) but are highly ranked ($\langle K \rangle \leq 50$) in the Wikipedia editions in which they appear, such as Tycho Brahe, Sejong the Great, and Sun Yat-sen (Left-Top of the Fig. 7A); (iii) locally-low

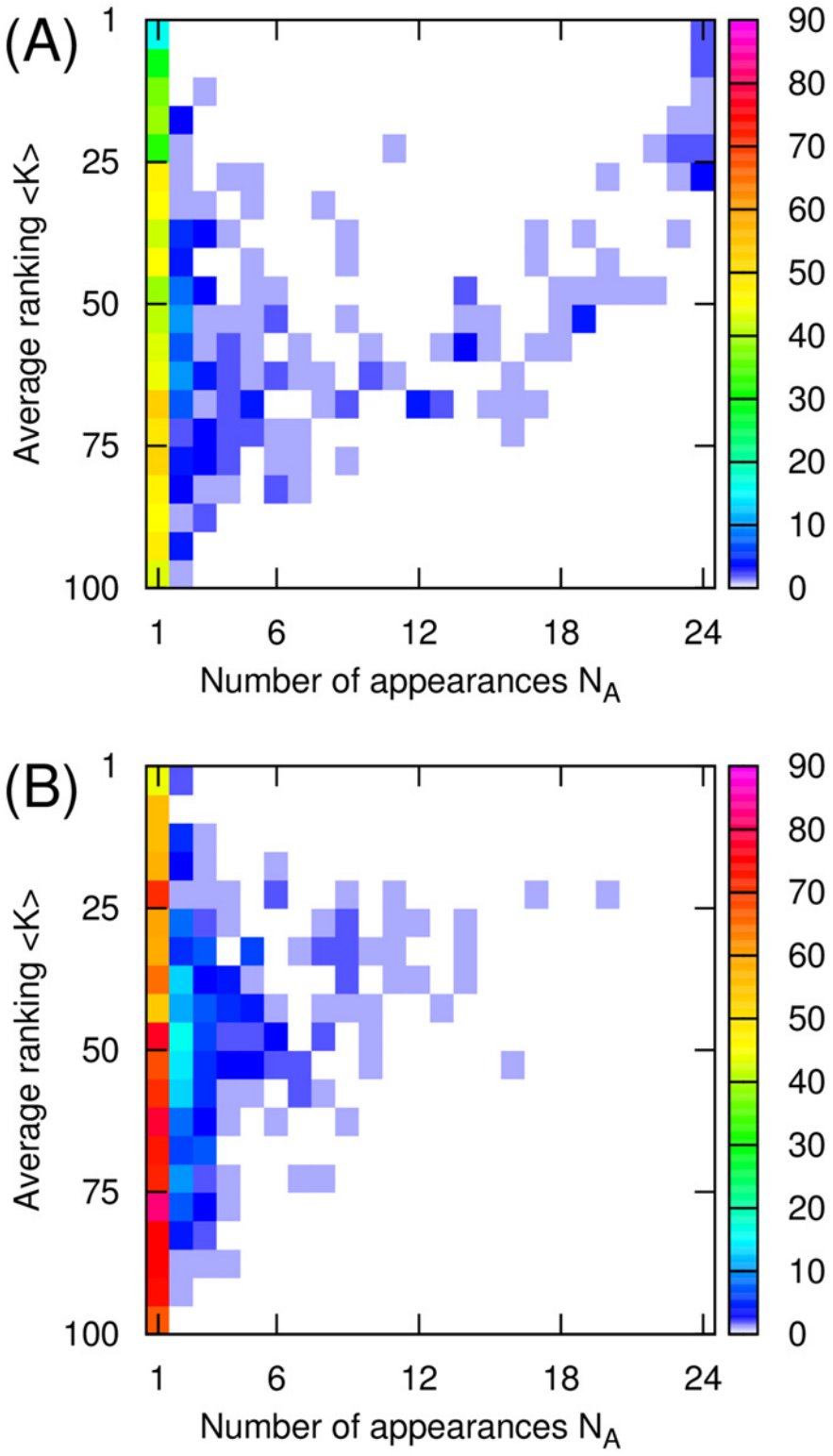


Fig 7. The distribution of 1045 top PageRank persons (A) and 1616 top 2DRank persons (B) as a function of number of appearances N_A of a given person and the rank $\langle K \rangle$ of this person averaged over Wikipedia editions where this person appeared.

doi:10.1371/journal.pone.0114825.g007

Table 5. List of the top 10 global female historical figures by PageRank and 2DRank for all the 24 Wikipedia editions. All names are represented by article titles in the English Wikipedia. Here, Θ_A is the ranking score of the algorithm A (Eq.3); N_A is the number of appearances of a given person in the top 100 rank for all editions. Here *CC* is the birth country code and *LC* is the language code of the given historical figure.

Rank	Θ_{PR}	N_A	PageRank female figures	*CC*	Century	*LC*
1	1789	24	Elizabeth II	UK	20	EN
2	1094	17	Mary (mother of Jesus)	IL	-1	HE
3	404	12	Queen Victoria	UK	19	EN
4	234	6	Elizabeth I of England	UK	16	EN
5	128	2	Maria Theresa	AT	18	DE
6	100	1	Benazir Bhutto	PK	20	HI
7	94	1	Catherine the Great	PL	18	PL
8	91	1	Anne Frank	DE	20	DE
9	87	1	Indira Gandhi	IN	20	HI
10	86	1	Margrethe II of Denmark	DK	20	DA
Rank	Θ_{2D}	N_A	**2DRank female figures**	*CC*	Century	*LC*
1	991	14	Madonna (entertainer)	US	20	EN
2	664	9	Elizabeth II	UK	20	EN
3	580	8	Mary (mother of Jesus)	IL	-1	HE
4	550	9	Queen Victoria	UK	19	EN
5	225	5	Agatha Christie	UK	19	EN
6	211	4	Mariah Carey	US	20	EN
7	206	7	Britney Spears	US	20	EN
8	200	3	Margaret Thatcher	UK	20	EN
9	191	2	Martina Navratilova	CZ	20	WR
10	175	2	Elizabeth I of England	UK	16	EN

doi:10.1371/journal.pone.0114825.t005

ranked historical figures who appear in a few Wikipedia editions ($N_A < 18$) and who are not highly ranked ($\langle K \rangle > 50$). Here N_A is the number of appearances in different Wikipedia editions for a given person and $\langle K \rangle$ is the average ranking of the given persons across Wikipedia editions for each ranking algorithm. In the case of 2DRank historical figures, due to the absence of global historical figures, most of them belong to two types of local historical figures (i.e. local-highly ranked or local-lowly ranked).

Following ranking of persons via $\Theta_{P,A}$ we determine also the top global female historical figures, presented in Table 5 for PageRank and 2DRank persons. The full lists of global female figures are available at [39] (63 and 165 names for PageRank and 2DRank).

The comparison of our 100 global historical figures with the top 100 from Hart's list [27] gives an overlap of 43 persons for PageRank and 26 persons for 2DRank. We note that for the top 100 from the English Wikipedia we obtain a lower overlap of 37 (PageRank) and 4 (2DRank) persons. Among all editions the highest overlaps with the Hart list are 42 (VI), 37 (EN, ES, PT, TR) and 33 (IT), 32 (DE), 31 (FR) for PageRank; while for 2DRank we find 18 (EL) and 17 (VI). We give the overlap numbers for all editions at [39]. This shows that the consideration of 24 editions provides us the global list of the top 100 persons with a more balanced selection of top historical figures. Our overlap of the top 100 global historical figures by PageRank with the top 100 people from Pantheon MIT ranking list [23] is 44 percent, while the overlap of this Pantheon list with Hart's list is 43 percent. We note that the Pantheon method is significantly based on a number of page views while our approach is based on the network structure of the whole Wikipedia network. The top 100 persons from [22] are not publicly available but nevertheless we present the overlaps between the top 100 persons from the lists of Hart, Pantheon,

Stony-Brook and our global PageRank and 2DRank lists in Figures S2, S3 (we received the Stony-Brook list as a private message from the authors of [22]). We have an average overlap between the 4 methods on a level of 40 percent (2DRank is on average lower by a few percent), we find a larger overlap between our PageRank list and the Stony-Brook list since the Stony-Brook method, applied only for the English Wikipedia, is significantly based on PageRank.

We also compared the distributions of our global top 100 persons of PageRank and 2DRank with the distribution of Hart's top 100 over centuries and over 24 languages with the additional WR category (see Figure S4). We find that these 3 distributions have very similar shapes. Thus the largest number of persons appears in centuries AD 18th, 19th, 20th for the 3 distributions. Among languages, the main peaks for the 3 distributions appear for EN, DE, IT, EL, AR, ZH. The deviations from Hart's distribution are larger for the 2DRank list. Thus the comparison of distributions over centuries and languages shows that the PageRank list has not only a strong overlap with the Hart list in the number of persons but that they also have very similar statistical distributions of the top 100 persons over centuries and languages.

The overlap of the top 100 global persons found here with the previous study [21] gives 54 and 47 percent for PageRank and 2DRank lists, respectively. However, we note that the global list in [21] was obtained from the top 30 persons in each edition while here we use the top 100 persons.

It is interesting to note that for the top 100 PageRank universities from the English Wikipedia edition the overlap with Shanghai top 100 list of universities is on a even higher level of 75 percent [18].

Finally, we note that the ranking of historical figures using the whole PageRank (or 2DRank) list of all Wikipedia articles of a given edition provides a more stable approach compared to the network of biographical articles used in [20]. Indeed, the number of nodes and links in such a biographical network is significantly smaller compared to the whole network of Wikipedia articles and thus the fluctuations become rather large. For example, from the biographical network of the Russian edition one finds as the top person *Napoleon III* (and even not *Napoleon I*) [20], who has a rather low importance for Russia. In contrast to that the present study gives us the top PageRank historical figure of the Russian edition to be *Peter the Great*, that has much more historical grounds. In a similar way for FR the results of [20] give at the first position *Adolf Hitler*, that is rather strange for the French culture, while we find a natural result *Napoleon*.

Network of cultures

We consider the selected top persons from each Wikipedia edition as important historical figures recognized by people who speak the language of that Wikipedia edition. Therefore, if a top person from a language edition *A* appears in another edition *B*, then we can consider this as a 'cultural' influence from culture *A* to *B*. Here we consider each language as a proxy for a cultural group and assign each historical figure to one of these cultural groups based on the most spoken language of her/his birth place at the country level. For example, *Adolf Hitler* was born in modern Austria and since German language is the most spoken language in Austria, he is considered as a German historical figure in our analysis. This method may lead to some misguiding results due to discrepancy between territories of country and cultures, e.g. *Jesus* was born in the modern State of Palestine (Bethlehem), which is an Arabic speaking country. Thus *Jesus* is from the Arabic culture in our analysis while usually one would say that he belongs to the Hebrew culture. Other similar examples we find are: *Charlemagne* (Belgium—Dutch), *Immanuel Kant* (Russia—Russian, while usually he is attributed to DE), *Moses* (Egypt—Arabic), *Catherine the Great* (Poland—Polish, while usually she would be attributed to DE or RU).

Table 6. Numbers of certain historical figures for top 100 list of each language: N_1 is the number of historical figures of a given language among the top 100 PageRank global historical figures; N_2 is the number of historical figures of a given language among the top 100 PageRank historical figures for the given language edition; N_3 is the number of historical figures of a given language among the top 100 2DRank global historical figures; N_4 is the number of historical figures of a given language among the top 100 2DRank historical figures for the given language edition.

Language	N_1	N_2	N_3	N_4	Language	N_1	N_2	N_3	N_4
EN	22	47	27	64	RU	2	29	3	27
NL	2	10	4	38	HE	2	17	2	22
DE	20	41	16	55	TR	2	27	2	54
FR	8	33	3	32	AR	8	42	5	69
ES	2	20	5	39	FA	0	46	1	64
IT	11	31	9	43	HI	1	65	0	76
PT	0	19	0	35	MS	0	15	0	40
EL	5	28	2	55	TH	0	46	0	53
DA	0	31	1	48	VI	0	7	0	30
SV	1	26	1	39	ZH	5	43	6	79
PL	1	20	2	26	KO	0	34	0	59
HU	0	18	0	18	JA	0	41	4	80
WR	8	-	7	-					

doi:10.1371/journal.pone.0114825.t006

In total there are such 36 cases from the global PageRank list of 1045 names (these 36 names are given in SI). However, in our knowledge, the birth place is the best way to assign a given historical figure to a certain cultural background computationally and systematically and with the data we have available. In total we have only about 3.4 percent of cases which can be discussed and where a native speaking language can be a better indicator of belonging to a given culture. For the global 2DRank list of 1616 names we identified 53 similar cases where an attribution to a culture via a native language or a birth place could be discussed (about 3.3 percent). These 53 names are given in SI. About half of such cases are linked with birth places in ancient Russian Empire where people from Belarus, Litvania and Ukraine moved to RU, IL, PL, WR. However, the percentage of such cases is small and the corresponding errors also remain small.

Based on the above assumption and following the approach developed in [21], we construct two weighted networks of cultures (or language groups) based on the top PageRank historical figures and top 2DRank historical figures respectively. Each culture (i.e. language) is represented as a node of the network, and the weight of a directed link from culture A to culture B is given by the number of historical figures belonging to culture B (e.g. French) appearing in the list of top 100 historical figures for a given culture A (e.g. English). The persons in a given edition, belonging to the language of the edition, are not taken into account since they do not create links between cultures. In Table 6 we give the number of such persons for each language. This table also gives the number of persons of a given language among the top 100 persons of the global PageRank and 2DRank listings.

For example, there are 5 French historical figures among the top 100 PageRank historical figures of the English Wikipedia, so we can assign weight 5 to the link from English to French. Fig. 8A and Fig. 8B represent the constructed networks of cultures defined by appearances of the top PageRank historical figures and top 2DRank historical figures, respectively. In total we have two networks with 25 nodes which include our 24 editions and an additional node WR for all the other world cultures.

The Google matrix G_{ij} for each network is constructed following the standard rules described in [21] and in the Methods Section. In a standard way we determine the PageRank index K and the CheiRank index $K*$ that order all cultures according to decreasing PageRank

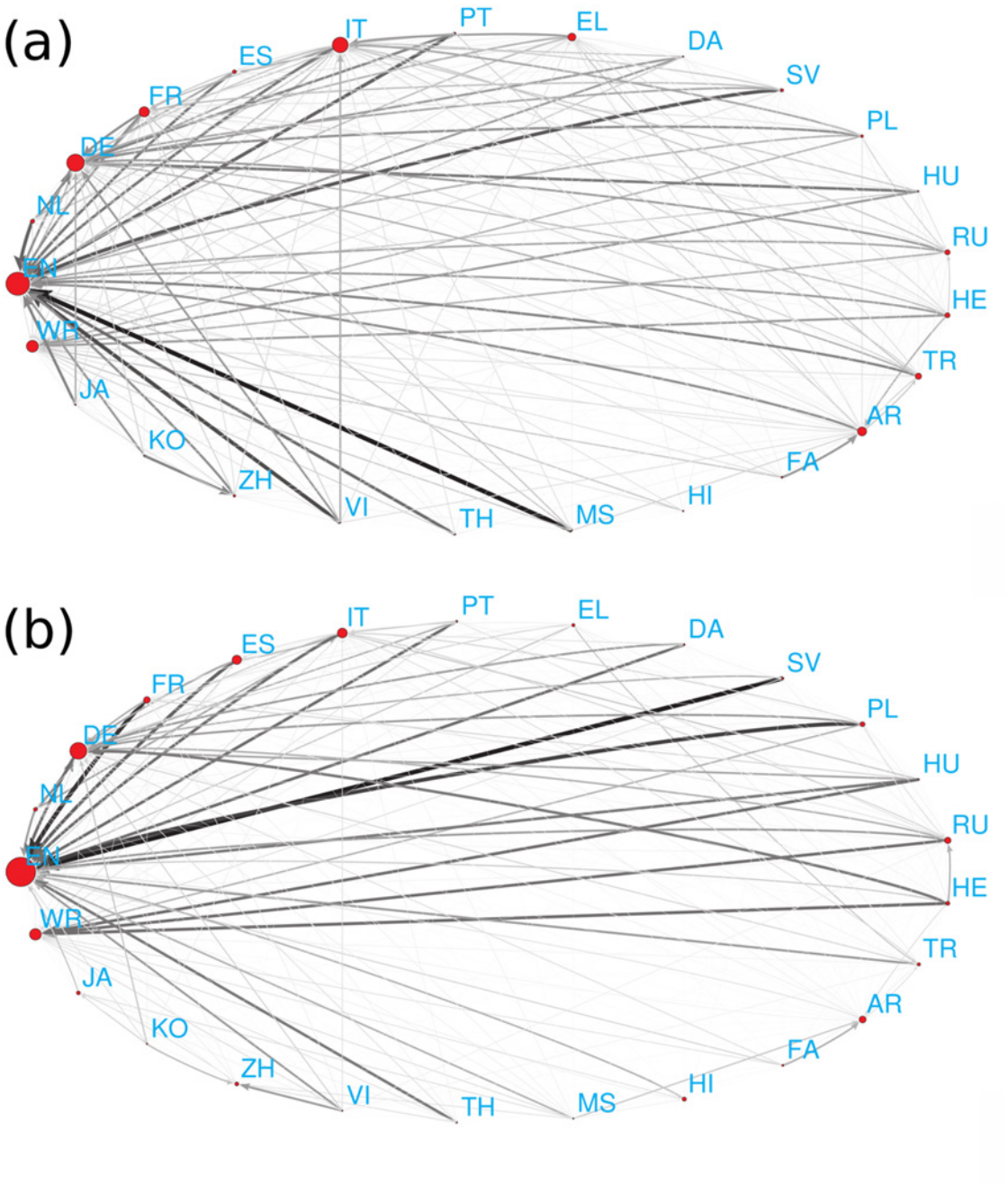


Fig 8. Network of cultures obtained from 24 Wikipedia languages and the remaining world (WR) consider (A) top PageRank historical figures and (B) 2DRank historical figures. The link width and darkness are proportional to a number of foreign historical figures quoted in top 100 of a given culture, the link direction goes from a given culture to cultures of quoted foreign historical figures, links inside cultures are not considered. The size of nodes is proportional to their PageRank.

doi:10.1371/journal.pone.0114825.g008

and CheiRank probabilities (see Methods and Figure S5). The structure of matrix elements $G_{KK'}$ is shown in Fig. 9.

To identify which cultures (or language groups) are more influential than others, we calculated PageRank and CheiRank of the constructed networks of cultures by considering link weights. Briefly speaking, a culture has high PageRank (CheiRank) if it has many ingoing

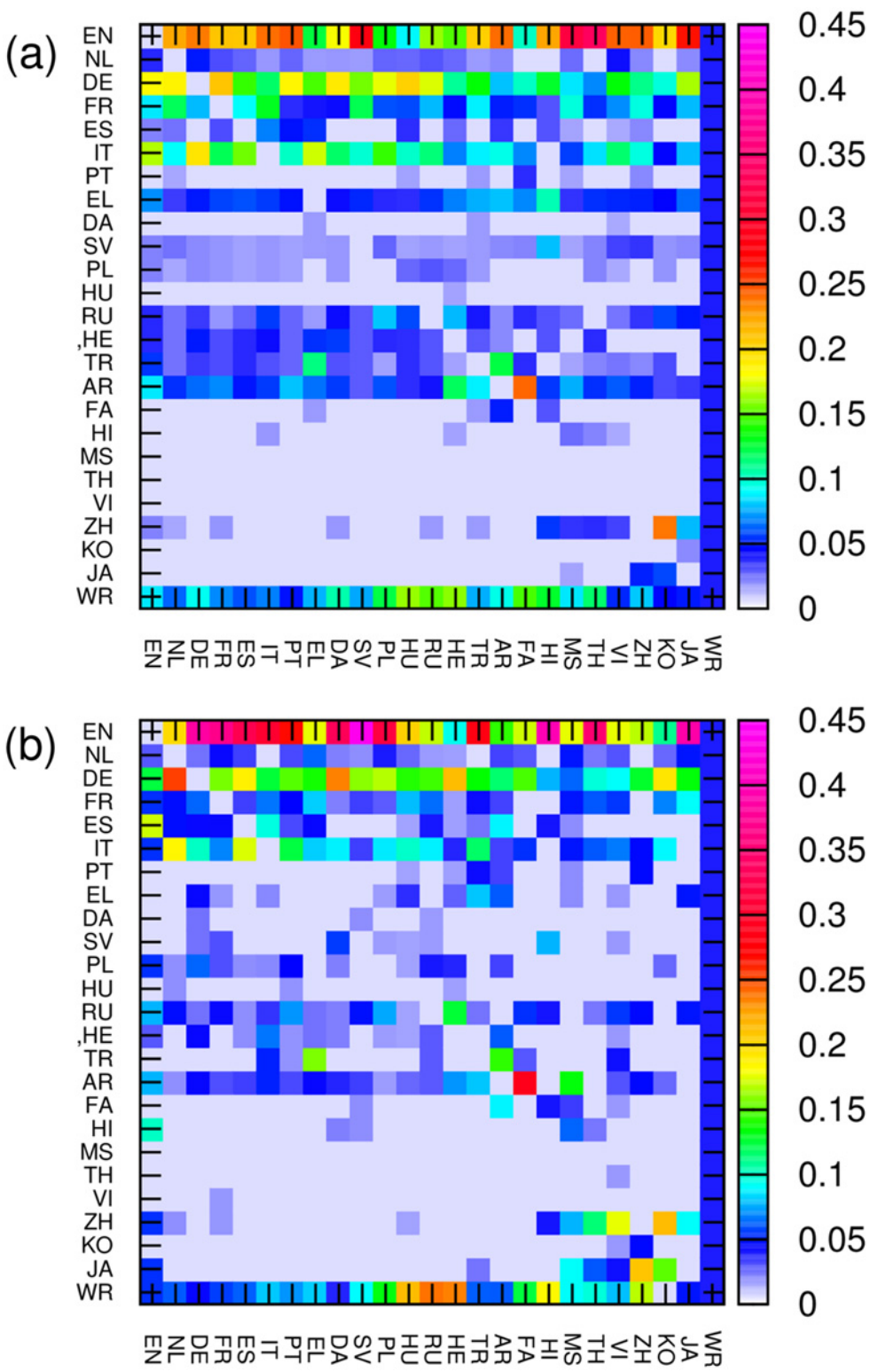


Fig 9. Google matrix of network of cultures shown in Fig. 8 respectively. The matrix elements G_{ij} are shown by color with damping factor $\alpha = 0.85$.

doi:10.1371/journal.pone.0114825.g009

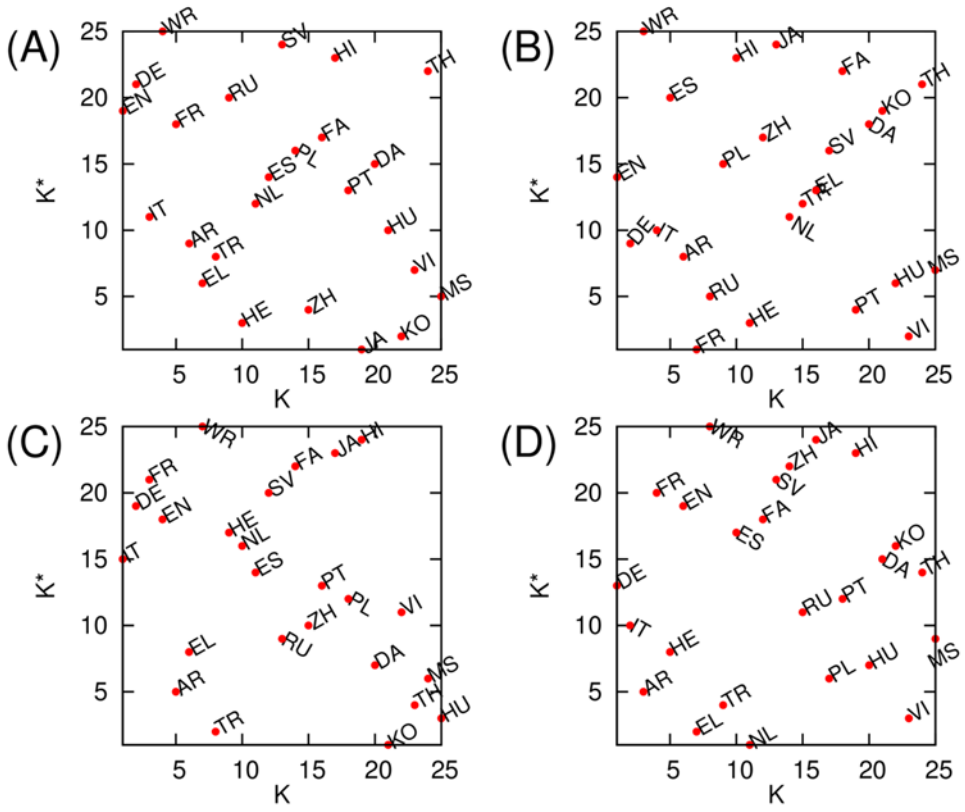


Fig 10. PageRank ranking versus CheiRank ranking plane of cultures with corresponding indexes *K* and *K obtained from the network of cultures based on (A) all PageRank historical figures, (B) all 2DRank historical figures, (C) PageRank historical figure born before AD 19th century, and (D) 2DRank historical figure born before AD 19th century, respectively.**

doi:10.1371/journal.pone.0114825.g010

(outgoing) links from (to) other cultures (see Methods). The distribution of cultures on a PageRank-CheiRank plane is shown in Fig. 10. In both cases of PageRank and 2DRank historical figures, historical figures of English culture (i.e. born in English language spoken countries) are the most influential (highest PageRank) and German culture is the second one (Fig. 10A, B). Here we consider the historical figures for the whole range of centuries. Fig. 10 represents the detailed features of how each culture is located on the plane of PageRank ranking K and CheiRank ranking $K*$ based on the top PageRank historical figures (Fig. 10A) and top 2DRank historical figures (Fig. 10B). Here K indicates the ranking of a given culture ordered by how many of its own top historical figures appear in other Wikipedia editions, and $K*$ indicates the ranking of a given culture according to how many of the top historical figures in the considered culture are from other cultures. As described above, English is on ($K = 1$, $K* = 19$) and German is on ($K = 2$, $K* = 21$) in the case of PageRank historical figures (Fig. 10A). In the case of 2DRank historical figures, English is on ($K = 1$, $K* = 14$) and German is on ($K = 2$, $K* = 9$).

It is important to note that there is a significant difference compared to the previous study [21]: there, only 9 editions had been considered and the top positions were attributed to the world node WR which captured a significant fraction of the top persons. This indicated that 9 editions are not sufficient to cover the whole world. Now for 24 editions we see that the importance of the world node WR is much lower (it moves from $K = 1$ for 9 editions [21] to $K = 4$ and 3 in Fig. 10A and Fig. 10B). Thus our 24 editions cover the majority the world. Still it

would be desirable to add a few additional editions (e.g. Ukraine, Baltic Republics, Serbia etc.) to fill certain gaps.

It is interesting to note that the ranking plane of cultures (K, $K*$) changes significantly in time. Indeed, if we take into account only persons born before the 19th century then the ranking is modified with EN going to 4th (Fig. 10C for PageRank figures) and 6th position (Fig. 10C for 2DRank figures) while the top positions are taken by IT, DE, FR and DE, IT, AR, respectively.

At the same time, we may also argue that for cultures it is important not only to be cited but also to be communicative with other cultures. To characterize communicative properties of nodes on the network of cultures shown in Fig. 8 we use again the concepts of PageRank, CheiRank and 2DRank for these networks as described in Methods and [21]. Thus, for the network of cultures of Fig. 8, the 2DRank index of cultures highlights their influence in a more balanced way taking into account their importance (incoming links) and communicative (outgoing links) properties in a balanced manner.

Thus we find for all centuries at the top positions Greek, Turkish and Arabic (for PageRank persons) and French, Russian and Arabic (for 2DRank persons). For historical figures before the 19th century, we find respectively Arabic, Turkish and Greek (for PageRank) and Arabic, Greek and Hebrew (for 2DRank). The high position of Turkish is due to its close links both with Greek culture in ancient times and with Arabic culture in more recent times. We see also that with time the positions of Greek in 2DRank improves due to a global improved ranking of Western cultures closely connected with Greece.

Discussion

By investigating birth place, birth date, and gender of important historical figures determined by the network structure of Wikipedia, we identified spatial, temporal, and gender skewness in Wikipedia. Our analysis shows that the most important historical figures across Wikipedia language editions were born in Western countries after the 17th century, and are male. Also, each Wikipedia edition highlights local figures so that most of its own historical figures are born in the countries which use the language of the edition. The emergence of such pronounced accent to local figures seems to be natural since there are more links and interactions within one culture. This is also visible from to the fact that in many editions the main country for the given language is at the first PageRank position among all articles (e.g. Russia in RU edition) [21]. Despite such a locality feature, there are also global historical figures who appear in most of the considered Wikipedia editions with very high rankings. Based on the cross-cultural historical figures, who appear in multiple editions, we can construct a network of cultures which describes interactions and entanglement between cultures.

It is very difficult to describe history in an objective way and due to that it was argued that history is "an unending dialogue between the past and present" [44]. In a similar way we can say that history is an unending dialogue between different cultural groups.

We use a computational and data mining approach, based on rank vectors of the Google matrix of Wikipedia, to perform a statistical analysis of interactions and entanglement of cultures. We find that this approach can be used for selecting the most influential historical figures through an analysis of collectively generated links between articles on Wikipedia. Our results are coherent with studies conducted by historians [27], with an overlap of 43% of important historical figures. Thus, such a mathematical analysis of local and global historical figures can be a useful step towards the understanding of local and global history and interactions of world cultures. Our approach has some limitations, mainly caused by the data source and by the difficulty of defining culture boundaries across centuries. The ongoing improvement of structured

content in Wikipedia through the WikiData project, eventually in conjunction with additional manual annotation, should allow to deal with these limitations. Furthermore, it would be useful to perform comparisons with other approaches to measure the interactions of cultures, such as the analysis of language crossings of multilingual users [45].

Influence of digital media on information dissemination and social collective opinions among the public is growing fast. Our research across Wikipedia language editions suggests a rigorous mathematical way, based on Markov chains and Google matrix, for the identification of important historical figures and for the analysis of interactions of cultures at different historical periods and in different world regions. We think that a further extension of this approach to a larger number of Wikipedia editions will provide a more detailed and balanced analysis of interactions of world cultures.

Supporting Information

S1 File. Supporting Information file S1 presents Figures S1–S5 with additional information discussed above in the main part of the paper, lists of top 100 global PageRank and 2DRank names; Tables S1–S25 of top 10 names of given language and remained world from the global PageRank and 2DRank ranking lists of persons ordered by the score $\Theta_{P,A}$ of Eq.(3). For a reader convenience the lists of all 100 ranked names for all 24 Wikipedia editions and corresponding network link data for each edition are also given at [39] in addition to Supporting Information file. All used computational data are publicly available at http://dumps.wikimedia.org/. All the raw data necessary to replicate the findings and conclusion of this study are within the paper, supporting information files and this Wikimedia web site.
(PDF)

Acknowledgments

This research is supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956)

Author Contributions

Conceived and designed the experiments: YHE DLS. Performed the experiments: YHE PA DL AK SV. Analyzed the data: YHE PA DLS. Contributed reagents/materials/analysis tools: YHE PA DL AK SV. Wrote the paper: YHE DLS.Conceived and designed the experiments: YHE DLS. Performed the experiments: YHE PA DL AK SV. Analyzed the data: YHE PA DLS. Contributed reagents/materials/analysis tools: YHE PA DL AK SV. Wrote the paper: YHE DLS.

References

1. Rosenzweig R (2006) Can history be open source? Wikipedia and the future and the past, Journal of American History 93(1): 117 doi: 10.2307/4486062
2. Lavsa SM, Corman SL, Culley CM, Pummer TL (2011) Reliability of Wikipedia as a medication information source for pharmacy students, Currents in Pharmacy Teaching and Learning 3(2): 154–158 doi: 10.1016/j.cptl.2011.01.007
3. Giles J (2005) Internet encylopedia go head to head, Nature, 438: 900 doi: 10.1038/438900a PMID: 16355180
4. Kittur A, Chi EH, Suh B (2009) What's in Wikipedia?: mapping topics and conflict using socially annotated category structure, In Proc. of SIGCHI Conference on Human Factors in Computing Systems, CHI'09, ACM, New York
5. Priedhorsky R, Chen J, Lam STK, Panciera K, Terveen L et al. (2007). Creating, Destroying, and Restoring Value in Wikipedia, In Proceedings of the Intl. Conf. on Supporting Group Work, 295, ACM, New York

6. Yasseri T, Sumi R, Rung A, Kornai A, Kertész J (2012) Dynamics of Conflicts in Wikipedia, PLoS ONE 7(6): e38869 doi: 10.1371/journal.pone.0038869 PMID: 22745683
7. Yasseri T, Spoerri A, Graham M, Kertész J (2013) The most controversial topics in Wikipedia: a multilingual and geographical analysis arXiv:1305.5566 [physics.soc-ph]
8. Laniado D, Tasso R, Volkovich Y. Kaltenbrunner A (2011) When the wikipedians talk: Network and tree structure of Wikipedia discussion pages, Proc. ICWSM 2011: 177–184
9. UNESCO World Report (2009) Investing in cultural diversity and intercultural dialogue, Available: http://www.unesco.org/new/en/culture/resources/report/the-unesco-world-report-on-cultural-diversity
10. Wikipedia: Neutral point of view. Retrived May 12, 2014 from http://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view
11. Pfeil U, Zaphiris P, Ang C A, (2006) Cultural Differences in Collaborative Authoring of Wikipedia, J. Computer-Mediated Comm. 12(1): 88 doi: 10.1111/j.1083-6101.2006.00316.x
12. Callahan ES, Herriing SC (2011) Cultural bias in Wikipedia content on famous persons, Journal of the American society for information science and technology 62: 1899 doi: 10.1002/asi.21577
13. Hecht B, Gergle D (2009) Measuring self-focus bias in community-maintained knowledge repositories, Proc. of the Fourth Intl Conf. Communities and technologies, ACM, New York 2009: 11
14. Hecht B, Gergle D (2010) The Tower of Babel Meets Web 2.0: User-Generated Content and Its Applications in a Multilingual Context, Proc. of SIGCHI Conference on Human Factors in Computing Systems, CHI'10, Atlanta, ACM, New York 291–300p
15. Nemoto K, Gloor PA (2011) Analyzing cultural differences in collaborative innovation networks by analyzing editing behavior in different-language Wikipedias, Procedia—Social and Behavioral Sciences 26: 180 doi: 10.1016/j.sbspro.2011.10.574
16. Warncke-Wang M, Uduwage A, Dong Z, Riedl J (2012) In search of the ur-Wikipedia: universality, similarity, and translation in the Wikipedia inter-language link network, Proceedings of the 8th Intl. Symposium on Wikis and Open Collaboration (WikiSym 2012), ACM, New York
17. Massa P, Scrinzi F (2012) Manypedia: Comparing language points of view of Wikipedia communities, Proceedings of the 8th Intl. Symposium on Wikis and Open Collaboration (WikiSym 2012), ACM, New York
18. Zhirov AO, Zhirov OV, Shepelyansky DL (2010) Two-dimensional ranking of Wikipedia articles, Eur. Phys. J. B 77: 523 doi: 10.1140/epjb/e2010-10500-7
19. Eom YH, Frahm KM, Benc ur A, Shepelyansky DL (2013) Time evolution of Wikipedia network ranking, Eur. Phys. J. B, 86:482 doi: 10.1140/epjb/e2013-40432-5
20. Aragón P, Laniado D, Kaltenbrunner A, Volkovich Y (2012) Biographical social networks on Wikipedia: a cross-cultural study of links that made history, Proc. of the 8th Intl. Symposium on Wikis and Open Collaboration (WikiSym 2012), ACM, New York No 19
21. Eom YH, Shepelyansky DL (2013) Highlighting Entanglement of Cultures via Ranking of Multilingual Wikipedia Articles, PLoS ONE, 8(10): e74554 doi: 10.1371/journal.pone.0074554 PMID: 24098338
22. Skiena S, Ward CB (2013) Who is Bigger?: Where Historical Figures Really Rank, Cambridge University Press, Cambridge UK
23. MIT Pantheon project. Available: http://pantheon.media.mit.edu. Accessed 2014 May 12.
24. Samoilenko A, Yasseri T (2014) The distorted mirror of Wikipedia: a quantitative analysis of Wikipedia coverage of academics, EPJ Data Sci. 3: 1 doi: 10.1140/epjds20
25. Wikipedia: List of languages by number of native speakers. Retrived May 12, 2014 from http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers
26. Wikipedia: Wikipedia. Retrived May 12, 2014 from http://en.wikipedia.org/wiki/Wikipedia
27. Hart MH (1992) The 100: ranking of the most influential persons in history, Citadel Press, N.Y.
28. Ermann L, Chepelianskii AD, Shepelyansky DL (2012) Toward two-dimensional search engines, J. Phys. A: Math. Theor. 45: 275101 doi: 10.1088/1751-8113/45/27/275101
29. Ermann L, Frahm KM, Shepelyansky DL (2013) Spectral properties of Google matrix of Wikipedia and other networks, Eur. Phys. J. D 86: 193 doi: 10.1140/epjb/e2013-31090-8
30. Langville AM, Meyer CD (2006) Google's PageRank and Beyond: The Science of Search Engine Rankings, Princeton University Press, Princeton
31. Brin S, Page L (1998) The anatomy of a large-scale hypertextual Web search engine, Computer Networks and ISDN Systems 30: 107 doi: 10.1016/S0169-7552(98)00110-X
32. Chen P, Xie H, Maslov S, Redner S (2007) Finding scientific gems with Google PageRank algorithm, Jour. Informetrics, 1: 8 doi: 10.1016/j.joi.2006.06.001

33. Kwak H, Lee C, Park H, Moon S (2010) What is Twitter, a social network or a news media?, Proc. 19th Int. Conf. WWW2010, ACM, New York 591p
34. Ermann L, Shepelyansky DL (2011) Google matrix of the world trade network, Acta Physica Polonica A 120(6A), A158
35. Kandiah V, Shepelyansky DL (2013) Google matrix analysis of DNA sequences, PLoS ONE 8(5): e61519 doi: 10.1371/journal.pone.0061519 PMID: 23671568
36. Chepelianskii AD (2010) Towards physical laws for software architecture, arXiv:1003.5455 [cs.SE]
37. Lü L, Zhang Y-C, Yeung CH, Zhou T (2011) Leaders in social networks, the delicious case, PLoS ONE 6(6): e21202 doi: 10.1371/journal.pone.0021202 PMID: 21738620
38. http://dbpedia.org. Accessed 2014 May 12
39. Top wikipeople. Available: http://www.quantware.ups-tlse.fr/QWLIB/topwikipeople/. Accessed 2014 May 12.
40. United States Census Bureau. Retrieved May 12, 2014 from http://www.census.gov/population/international/data/worldpop/table_history.php
41. Wikipedia: Wikipedians. Retrived May 12, 2014 from http://en.wikipedia.org/wiki/Wikipedia:Wikipedians
42. Statistics and indicators on women and men by United Nation. http://unstats.un.org/unsd/Demographic/products/indwm/ (accesible May 12, 2014)
43. Lam STK, Uduwage A, Dong Z, Sen S (2011) WP:clubhouse?: an exploration of Wikipedia's gender imbalance, Proc. of the 7th Intl. Symposium on Wikis and Open Collaboration, WikiSym'11, Moutain View 1–10p
44. Carr EH (1961) What is History?, Vintage Books, New York
45. Hale SA (2014) Multilinguals and Wikipedia editing, Proc. 6th Annual ACM Web Science Conf. ACM New York 1, 99

SUPPORTING INFORMATION FOR:
Interactions of cultures and top people of Wikipedia from ranking of 24 language editions

Young-Ho Eom[1], Pablo Aragón[2], David Laniado[2], Andreas Kaltenbrunner[2], Sebastiano Vigna[3], Dima L. Shepelyansky[1,*]

1 *Laboratoire de Physique Théorique du CNRS, IRSAMC, Université de Toulouse, UPS, F-31062 Toulouse, France*
2 *Barcelona Media Foundation, Barcelona, Spain*
3 *Dipartimento di Informatica, Università degli Studi di Milano, Milano, Italy*

∗ Corresponding Author E-mail: dima@irsamc.ups-tlse.fr

1 Additional data

Here we present additional figures and tables for the main part of the paper.

Figure S1 is analogous to Figures 4(C,D,E,F), however, now on the vertical axis we plot not the edition to which a given historical figure is attributed from top 100 figures of a given edition but the language, to which this historical figure from the global PageRank (1045 persons) or 2DRank (1616 persons) lists is attributed according to our procedure according to her/his country of birth and then to the major language of this country, if a person does not belong to any of 24 languages then he/she is attributed to the remaining world (WR). The data show that the separation between language (or culture) groups becomes now more distinct. Indeed, attribution to a language related to a birth place is more definite compared to the option where a person appears in one of 24 editions since some global historical figures appear in a few editions while each person is attributed only one language according to our procedure.

Figure S2 shows overlap between the global list of top 100 global PageRank persons and list of Hart [23], PageRank list of English Wikipedia from [15], list of Stony-Brook [19], list of Pantheon MIT project [20].

Figure S3 shows the overlap matrix (in percent) between 5 methods of ranking of top 100 historical figures including Hart, Pantheon, Stony-Brook results and our global PageRank and 2DRank lists. We see that our PageRank has most high correlation with Stony-Brook since the method of Stony-Brook uses significantly the PageRank method.

Figure S4 shows the number of persons from top 100 lists of Hart and our global PageRank and 2DRank lists. The panel (A) shows the number of persons at a given century corresponding to the time dependence and the panel (B) shows distribution of such persons over the language they are attributed according to our method based on the birth place and dominant language of a country of birth. We see that the pattern of Hart ranking is well reproduced from our global ranking, especially for the case of PageRank list.

Figure S5 shows PageRank and CheiRank probabilities for the networks of cultures shown in Figure 8.

The names of persons from top 100 missed by automatic recovery of persons are: Homer, Charles Darwin (RU PageRank); Philipp Kirkorov (RU 2DRank); Alexander the Great, Emperor Gaozu of Han, Homer (KO PageRank); Jinpyeong of Silla, Hyeonjong of Goryeo (KO 2DRank).

Unfortunately, the name of Homer has been missed in the 1.1 million list of English names, other names are missed due to incompleteness and modifications of inter-language translations.

Below we give the list of global top 100 PageRank names from 24 Wikipedia editions. The names are ordered by the ranking score $\Theta_{P,A}$ of Eq.(1). In brackets we give country of birth, century of birth, gender, and language of birth. In the same manner we also give th list of top 100 2DRank names from 24 Wikipedia editions.

We also give 24 names from global 1045 PageRank names and 40 names from 1616 global 2DRank names where a birth place language attribution differs from native language.

We also give the tables of top 10 persons in each language and also world names (tables S1 - S25) extracted from the global PageRank and 2DRank ranking lists of persons ordered by the score $\Theta_{P,A}$ of Eq.(1).

Top 100 of global PageRank names: 1. Carl Linnaeus (SE, 18, M, SV) 2. Jesus (PS, -1, M, AR) 3. Aristotle (GR, -4, M, EL) 4. Napoleon (FR, 18, M, FR) 5. Adolf Hitler (AT, 19, M, DE) 6. Julius Caesar (IT, -1, M, IT) 7. Plato (GR, -5, M, EL) 8. William Shakespeare (UK, 16, M, EN) 9. Albert Einstein (DE, 19, M, DE) 10. Elizabeth II (UK, 20, F, EN) 11. Alexander the Great (GR, -4, M, EL) 12. Isaac Newton (UK, 17, M, EN) 13. Muhammad (SA, 6, M, AR) 14. Karl Marx (DE, 19, M, DE) 15. Joseph Stalin (GE, 19, M, WR) 16. Augustus (IT, -1, M, IT) 17. Christopher Columbus (IT, 15, M, IT) 18. Charlemagne (BE, 8, M, NL) 19. Louis XIV of France (FR, 17, M, FR) 20. George W. Bush (US, 20, M, EN) 21. Immanuel Kant (RU, 18, M, RU) 22. Barack Obama (US, 20, M, EN) 23. Mary (mother of Jesus) (IL, -1, F, HE) 24. Vladimir Lenin (RU, 19, M, RU) 25. Wolfgang Amadeus Mozart (AT, 18, M, DE) 26. Paul the Apostle (TR, 1, M, TR) 27. Charles Darwin (UK, 19, M, EN) 28. Martin Luther (DE, 15, M, DE) 29. Herodotus (TR, -5, M, TR) 30. Franklin D. Roosevelt (US, 19, M, EN) 31. Galileo Galilei (IT, 16, M, IT) 32. Pope John Paul II (PL, 20, M, PL) 33. Constantine the Great (RS, 3, M, WR) 34. Benito Mussolini (IT, 19, M, IT) 35. Cicero (IT, -2, M, IT) 36. Ren Descartes (FR, 16, M, FR) 37. Saint Peter (IL, 1, M, HE) 38. Ludwig van Beethoven (DE, 18, M, DE) 39. George Washington (US, 18, M, EN) 40. Moses (EG, -14, M, AR) 41. Johann Sebastian Bach (DE, 17, M, DE) 42. Bill Clinton (US, 20, M, EN) 43. Leonardo da Vinci (IT, 15, M, IT) 44. Johann Wolfgang von Goethe (DE, 18, M, DE) 45. Gautama Buddha (NP, -6, M, WR) 46. Winston Churchill (UK, 19, M, EN) 47. John F. Kennedy (US, 20, M, EN) 48. Charles V, Holy Roman Emperor (BE, 15, M, NL) 49. Pope Benedict XVI (DE, 20, M, DE) 50. Richard Nixon (US, 20, M, EN) 51. Sigmund Freud (CZ, 19, M, WR) 52. Ronald Reagan (US, 20, M, EN) 53. Abraham Lincoln (US, 19, M, EN) 54. Saddam Hussein (IQ, 20, M, AR) 55. Ptolemy (EG, 1, M, AR) 56. Richard Wagner (DE, 19, M, DE) 57. Diocletian (HR, 3, M, WR) 58. Queen Victoria (UK, 19, F, EN) 59. Napoleon III (FR, 19, M, FR) 60. Charles de Gaulle (FR, 19, M, FR) 61. Mao Zedong (CN, 19, M, ZH) 62. William Herschel (DE, 18, M, DE) 63. Michael Jackson (US, 20, M, EN) 64. Justinian I (MK, 5, M, WR) 65. Augustine of Hippo (DZ, 4, M, AR) 66. Ali (SA, 7, M, AR) 67. Jean-Jacques Rousseau (CH, 18, M, DE) 68. Ernst Haeckel (DE, 19, M, DE) 69. Pliny the Elder (IT, 1, M, IT) 70. Pope Gregory XIII (IT, 16, M, IT) 71. Confucius (CN, -6, M, ZH) 72. Henry VIII of England (UK, 15, M, EN) 73. Thomas Jefferson (US, 18, M, EN) 74. Francisco Franco (ES, 19, M, ES) 75. Georg Wilhelm Friedrich Hegel (DE, 18, M, DE) 76. Pierre Andr Latreille (FR, 18, M, FR) 77. Pope Paul VI (IT, 19, M, IT) 78. Gottfried Wilhelm Leibniz (DE, 17, M, DE) 79. Chiang Kai-shek (CN, 19, M, ZH) 80. John Herschel (UK, 18, M, EN) 81. Elizabeth I of England (UK, 16, F, EN) 82. J. R. R. Tolkien

(ZA, 19, M, WR) 83. Socrates (GR, -5, M, EL) 84. Genghis Khan (MN, 12, M, WR) 85. Qin Shi Huang (CN, -3, M, ZH) 86. Umar (SA, 6, M, AR) 87. Philip II of Spain (ES, 16, M, ES) 88. Frederick the Great (DE, 18, M, DE) 89. Johannes Kepler (DE, 16, M, DE) 90. Emperor Wu of Han (CN, -2, M, ZH) 91. Friedrich Nietzsche (DE, 19, M, DE) 92. Plutarch (GR, 1, M, EL) 93. Thomas Edison (US, 19, M, EN) 94. Max Weber (DE, 19, M, DE) 95. Dante Alighieri (IT, 13, M, IT) 96. Ashoka (IN, -4, M, HI) 97. Tacitus (FR, 1, M, FR) 98. Ernst Mayr (DE, 20, M, DE) 99. Jean-Baptiste Lamarck (FR, 18, M, FR) 100. Elvis Presley (US, 20, M, EN).

Top 100 of global 2DRank names: 1. Adolf Hitler (AT, 19, M, DE) 2. Michael Jackson (US, 20, M, EN) 3. Madonna (entertainer) (US, 20, F, EN) 4. Jesus (PS, -1, M, AR) 5. Ludwig van Beethoven (DE, 18, M, DE) 6. Wolfgang Amadeus Mozart (AT, 18, M, DE) 7. Pope Benedict XVI (DE, 20, M, DE) 8. Alexander the Great (GR, -4, M, EL) 9. Charles Darwin (UK, 19, M, EN) 10. Barack Obama (US, 20, M, EN) 11. Johann Sebastian Bach (DE, 17, M, DE) 12. Napoleon (FR, 18, M, FR) 13. Pope John Paul II (PL, 20, M, PL) 14. Julius Caesar (IT, -1, M, IT) 15. Elizabeth II (UK, 20, F, EN) 16. Albert Einstein (DE, 19, M, DE) 17. Augustus (IT, -1, M, IT) 18. Bob Dylan (US, 20, M, EN) 19. Leonardo da Vinci (IT, 15, M, IT) 20. Mary (mother of Jesus) (IL, -1, F, HE) 21. Charlemagne (BE, 8, M, NL) 22. William Shakespeare (UK, 16, M, EN) 23. Elvis Presley (US, 20, M, EN) 24. Queen Victoria (UK, 19, F, EN) 25. John Lennon (UK, 20, M, EN) 26. George Frideric Handel (DE, 17, M, DE) 27. J. R. R. Tolkien (ZA, 19, M, WR) 28. Muhammad (SA, 6, M, AR) 29. Joseph Stalin (GE, 19, M, WR) 30. Karl Marx (DE, 19, M, DE) 31. Benito Mussolini (IT, 19, M, IT) 32. Franklin D. Roosevelt (US, 19, M, EN) 33. Michael Schumacher (DE, 20, M, DE) 34. Paul McCartney (UK, 20, M, EN) 35. Stephen King (US, 20, M, EN) 36. Henry VIII of England (UK, 15, M, EN) 37. Tokugawa Ieyasu (JP, 16, M, JA) 38. Edgar Allan Poe (US, 19, M, EN) 39. Martin Luther (DE, 15, M, DE) 40. David Bowie (UK, 20, M, EN) 41. Pope Pius XII (IT, 19, M, IT) 42. Alfred Hitchcock (UK, 19, M, EN) 43. Friedrich Nietzsche (DE, 19, M, DE) 44. Vladimir Putin (RU, 20, M, RU) 45. Christopher Columbus (IT, 15, M, IT) 46. Elton John (UK, 20, M, EN) 47. Carl Linnaeus (SE, 18, M, SV) 48. Michelangelo (IT, 15, M, IT) 49. Raphael (IT, 15, M, IT) 50. Roger Federer (CH, 20, M, DE) 51. Cao Cao (CN, 2, M, ZH) 52. Vincent van Gogh (NL, 19, M, NL) 53. Frdric Chopin (PL, 19, M, PL) 54. Steven Spielberg (US, 20, M, EN) 55. Rembrandt (NL, 17, M, NL) 56. Ali (SA, 7, M, AR) 57. Richard Wagner (DE, 19, M, DE) 58. Che Guevara (AR, 20, M, ES) 59. Nelson Mandela (ZA, 20, M, WR) 60. Isaac Asimov (RU, 20, M, RU) 61. Jules Verne (FR, 19, M, FR) 62. Toyotomi Hideyoshi (JP, 16, M, JA) 63. Winston Churchill (UK, 19, M, EN) 64. Paul the Apostle (TR, 1, M, TR) 65. Hirohito (JP, 20, M, JA) 66. 14th Dalai Lama (CN, 20, M, ZH) 67. Franz Liszt (AT, 19, M, DE) 68. Genghis Khan (MN, 12, M, WR) 69. Otto von Bismarck (DE, 19, M, DE) 70. Saint Peter (IL, 1, M, HE) 71. Charlie Chaplin (UK, 19, M, EN) 72. Liu Bei (CN, 2, M, ZH) 73. Oda Nobunaga (JP, 16, M, JA) 74. Suleiman the Magnificent (TR, 15, M, TR) 75. Cyrus the Great (IR, -6, M, FA) 76. George W. Bush (US, 20, M, EN) 77. Agatha Christie (UK, 19, F, EN) 78. Carl Friedrich Gauss (DE, 18, M, DE) 79. Louis XIV of France (FR, 17, M, FR) 80. Saddam Hussein (IQ, 20, M, AR) 81. Pablo Picasso (ES, 19, M, ES) 82. Mariah Carey (US, 20, F, EN) 83. Hans Christian Andersen (DK, 19, M, DA) 84. Plato (GR, -5, M, EL) 85. Britney Spears (US, 20, F, EN) 86. Rafael Nadal (ES, 20, M, ES) 87. George Harrison (UK, 20, M, EN) 88. Margaret Thatcher (UK, 20, F, EN) 89. Jorge Luis Borges (AR, 19, M, ES) 90. Salvador Dal (ES, 20, M, ES) 91. Peter the Great (RU, 17, M, RU) 92. Giuseppe Verdi (IT, 19, M, IT) 93. Sigmund Freud (CZ, 19, M,

WR) 94. Qin Shi Huang (CN, -3, M, ZH) 95. Kangxi Emperor (CN, 17, M, ZH) 96. Martina Navratilova (CZ, 20, F, WR) 97. Charles V, Holy Roman Emperor (BE, 15, M, NL) 98. Zhuge Liang (CN, 2, M, ZH) 99. Constantine the Great (RS, 3, M, WR) 100. Muammar Gaddafi (LY, 20, M, AR)

List of 36 names from the global PageRank list of 1045 names where the birth place in modern geography of countries differs from native language: Jesus (PS AR), Charlemagne (Belgium NL), Immanuel Kant (Russia RU), Moses (Egypt AR), Catherine the Great (Poland PL), Mustafa Kemal Atatürk (Greece EL), Bhumibol Adulyadej (USA EN), Christian V of Denmark (Germany DE), Józef Pilsudski (Litvania WR), Christian IX of Denmark (Germany DE), Philip V of Spain (France FR), Giuseppe Garibaldi (France FR), Muhammad al-Idrisi (Spain ES), Charles XIV John of Sweden (France FR), Leonid Brezhnev (Ukraine WR), George I of Greece (Denmark DA), Juan Carlos I of Spain (Italy IT), Leon Trotsky (Ukraine WR), Golda Meir (Ukraine WR), Valéry Giscard d'Estaing (Germany DE), Magnus IV of Sweden (Noroway WR), Christian I of Denmark (Germany DE), Yitzhak Ben-Zvi (Ukraine WR), Mikhail Bulgakov (Ukraine WR); Kim Jong-il (Russia RU). Lee Myung-bak (Japan JA), Jangsu of Goguryeo (China ZH); Galyani Vadhana (UK EN), Abhisit Vejjajiva (UK EN); Matthias Corvinus (Romania WR), Ferenc Kazinczy (Romania WR), György Kulin (Romania WR), Gabriel Bethlen (Romania WR), Endre Ady (Romania WR), János Arany (Romania WR), Béla Bartók (Romania WR).

List of 53 names from the global 2DRank list of 1616 names where the birth place in modern geography of countries differs from native language: Jesus (PS AR), Charlemagne (BE NL), Isaac Asimov (RU RU), Paul the Apostle (TR TR), Peter Paul Rubens (DE DE), Catherine the Great (PL PL), Julian (emperor) (TR TR), Józef Pilsudski (LT WR), Muhammad Ali of Egypt (GR EL), Juan Carlos I of Spain (IT IT), Shmuel Yosef Agnon (UA WR), Saint Joseph (PS AR), Golda Meir (UA WR), Baibars (UA WR), Levi Eshkol (UA WR), Augustine of Hippo (DZ AR), Yitzhak Ben-Zvi (UA WR), Natan Yonatan (UA WR), Edward Rydz-migy (UA WR), Immanuel Kant (RU RU), Pyotr Stolypin (DE DE), Czeslaw Niemen (BY RU), Moses (EG AR), Albert Camus (DZ AR), Leonid Brezhnev (UA WR), Aharon Barak (LT WR), George Orwell (IN HI), Sergei Korolev (UA WR), Garry Kasparov (AZ TR), Ibn 'Abd al-Barr (ES ES), Georges Simenon (BE NL), Ryszard Kapuściński (BY RU), Mihly Munkácsy (UA WR), Juliusz Slowacki (UA WR), Tadeusz Kościuszko (BY RU), John McCain (PA ES), Maurice, Prince of Orange (DE DE), Zbigniew Herbert (UA WR), Leon Trotsky (UA WR), Charles XIV John of Sweden (FR FR). Lee Myung-bak (JA JA), Jangsu of Goguryeo (CN ZH), Gwanggaeto the Great (CN ZH); Galyani Vadhana (UK EN), Abhisit Vejjajiva (UK EN); Matthias Corvinus (RO WR), Károly Kós (RO WR), László Németh (RO WR), Sándor Körösi Csoma (RO WR), János Bolyai (RO WR), György Kulin (RO WR), Ferenc Kazinczy (RO WR), Béla Bartók (RO WR).

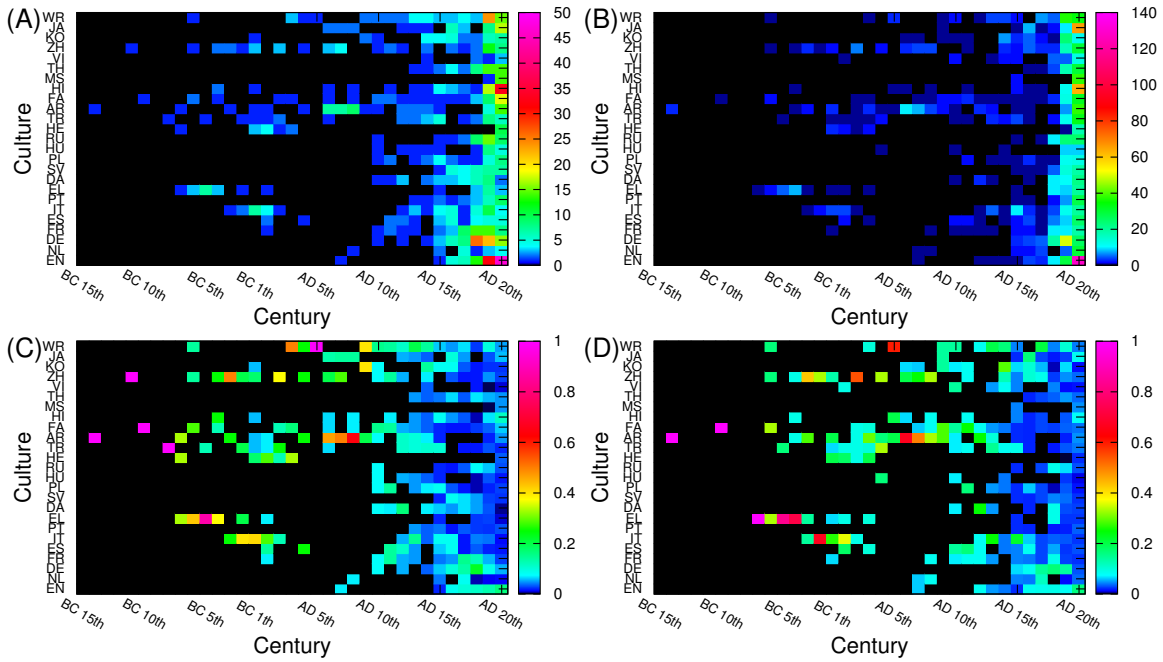


Figure S1. Birth date distribution of historical figures from the global PageRank list (A,C, 1045 persons) and 2DRank list (B,D, 1616 persons). Each historical figure is attributed to her/his own language according to her/his birth place as described in the paper (if the birth place is not among our 24 languages then a person is attributed to the remaining world (WR)). Color in panels (A,B) shows the total number of persons for a given century, while in panels (C,D) color shows a percent for a given century (normalized to unity in each column). This figure give a more distinct separation of cultures (languages) compared to a similar Fig.4 where the distribution over Wikipedia editions is shown on the vertical axis.

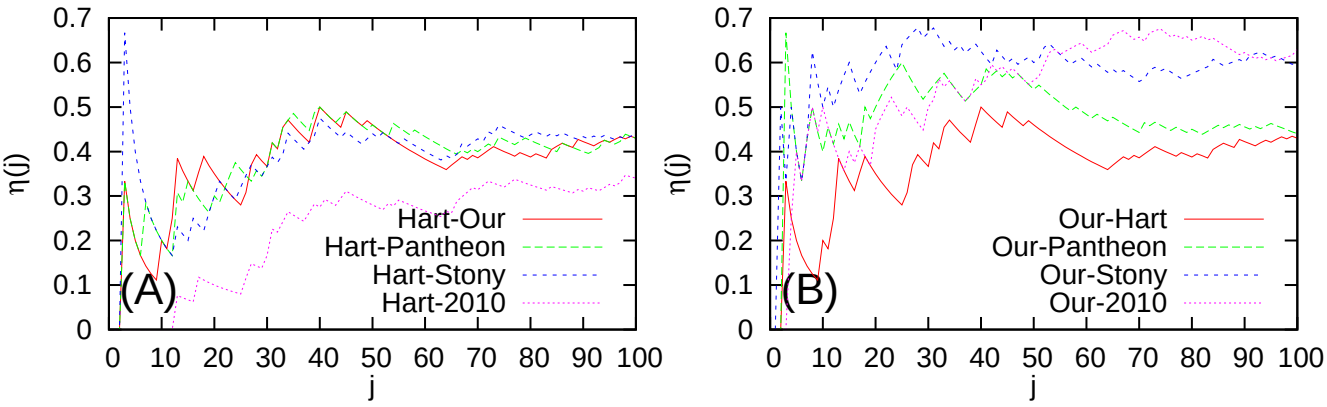


Figure S2. Dependence of fraction η of overlaped persons on rank index of person j. (A) Comparison is done of present study ("our"), PageRank list of English Wikipedia of [15] ("2010"), Stony-Brook list [19], Pantheon MIT project [20] in respect to Hart top 100 list. (B) Same as in (A) but comparison is done in respect to present study.

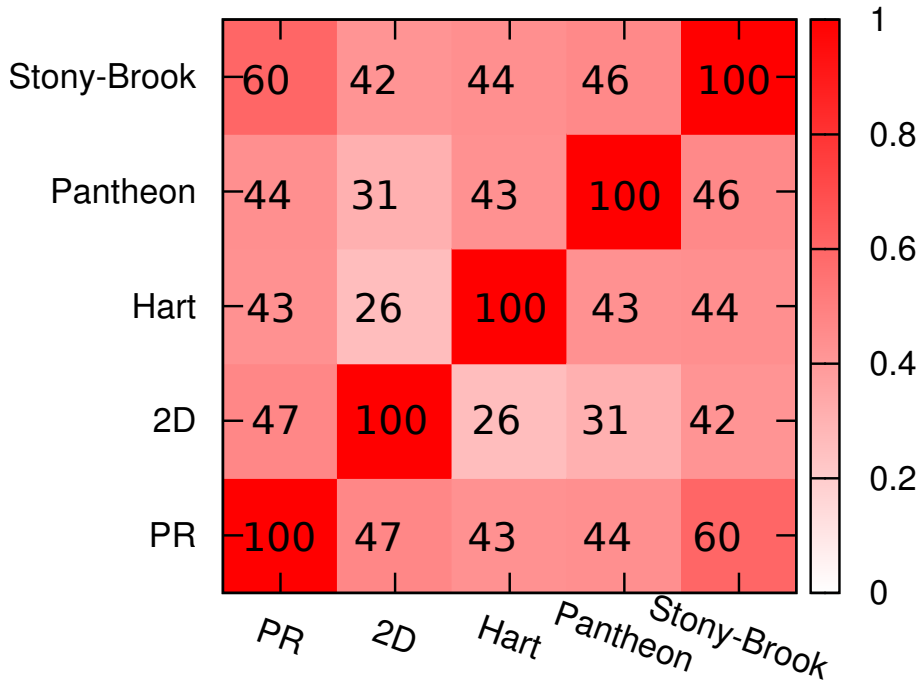


Figure S3. The overlap matrix (in percent) between 5 methods of ranking of top 100 historical figures from lists of Hart, Pantheon, Stony-Brook results and our global PageRank and 2DRank lists; percent or number of persons common for two lists is shown by color and numbers.

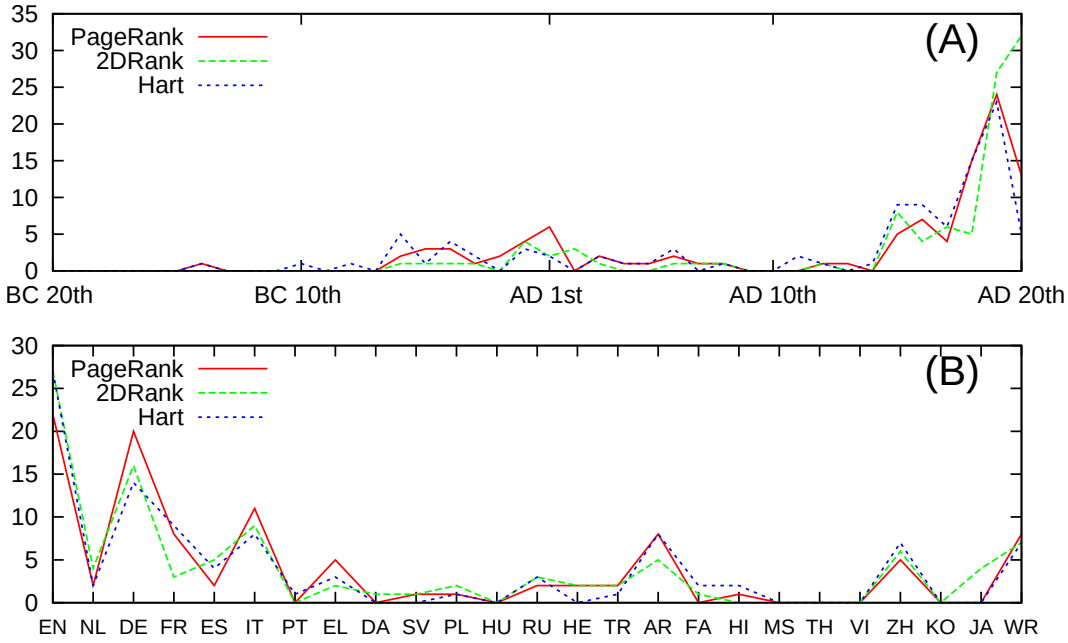


Figure S4. The number of top 100 historical figures, from the list of Hart and our global PageRank and 2DRank lists, are shown as a function of time (for a given century, panel A; one person from Hart's list *Menes*, born in Egypt at BC 32nd and thus attributed to AR, is outside of time range in this panel but he is counted in panel B) and for a given language to which a person is attributed according to her/his birth place (panel B).

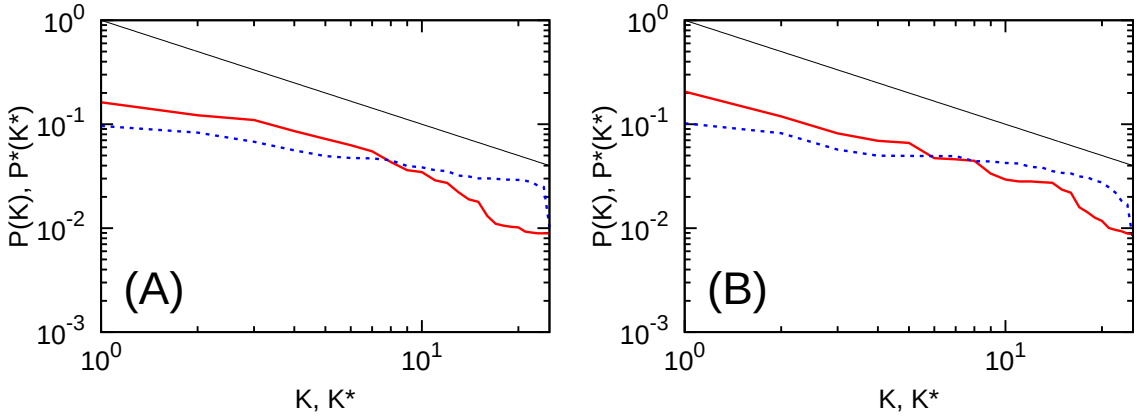


Figure S5. Dependence of probabilities of PageRank P (red) and CheiRank P^* (blue) on corresponding indexes K and K^*. The probabilities are obtained from the network shown in Fig.7 for corresponding panels $(A), (B)$. The straight lines indicate the Zipf's law $P \sim 1/K; P^* \sim 1/K^*$.

Table S1. List of local historical figures for EN category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	1861	William Shakespeare	1315	Michael Jackson
2	1789	Elizabeth II	991	Madonna (entertainer)
3	1756	Isaac Newton	773	Charles Darwin
4	1173	George W. Bush	754	Barack Obama
5	1101	Barack Obama	664	Elizabeth II
6	932	Charles Darwin	624	Bob Dylan
7	910	Franklin D. Roosevelt	556	William Shakespeare
8	656	George Washington	555	Elvis Presley
9	596	Bill Clinton	550	Queen Victoria
10	564	Winston Churchill	541	John Lennon

Table S2. List of local historical figures for NL category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	1476	Charlemagne	569	Charlemagne
2	556	Charles V, Holy Roman Emperor	297	Vincent van Gogh
3	83	Maurice Maeterlinck	294	Rembrandt
4	81	William I of the Netherlands	190	Charles V, Holy Roman Emperor
5	78	Beatrix of the Netherlands	138	Beatrix of the Netherlands
6	61	Baruch Spinoza	98	Baruch Spinoza
7	61	Rembrandt	94	Hugo Claus
8	51	Wilhelmina of the Netherlands	91	Johan Cruyff
9	47	Juliana of the Netherlands	76	Louis Couperus
10	39	Christiaan Huygens	75	Pierre Cuypers

Table S3. List of local historical figures for DE category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1)

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	2112	Adolf Hitler	1557	Adolf Hitler
2	1847	Albert Einstein	872	Ludwig van Beethoven
3	1730	Karl Marx	853	Wolfgang Amadeus Mozart
4	996	Wolfgang Amadeus Mozart	840	Pope Benedict XVI
5	925	Martin Luther	733	Johann Sebastian Bach
6	700	Ludwig van Beethoven	651	Albert Einstein
7	610	Johann Sebastian Bach	540	George Frideric Handel
8	570	Johann Wolfgang von Goethe	465	Karl Marx
9	528	Pope Benedict XVI	446	Michael Schumacher
10	417	Richard Wagner	344	Martin Luther

Table S4. List of local historical figures for FR category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	2208	Napoleon	720	Napoleon
2	1207	Louis XIV of France	268	Jules Verne
3	724	René Descartes	221	Louis XIV of France
4	397	Napoleon III	168	Giuseppe Garibaldi
5	385	Charles de Gaulle	146	Denis Diderot
6	260	Pierre André Latreille	144	Franois Mitterrand
7	167	Tacitus	127	Napoleon III
8	165	Jean-Baptiste Lamarck	121	Nicolas Sarkozy
9	157	Molière	113	Claudius
10	112	Francis I of France	112	Henry IV of France

Table S5. List of local historical figures for ES category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	276	Francisco Franco	285	Che Guevara
2	195	Philip II of Spain	216	Pablo Picasso
3	119	Pablo Picasso	206	Rafael Nadal
4	82	Lionel Messi	199	Jorge Luis Borges
5	74	Charles III of Spain	198	Salvador Dalí
6	72	Teresa of Ávila	178	Hadrian
7	71	Miguel de Cervantes	105	Shakira
8	70	Ferdinand VII of Spain	100	Francisco Goya
9	66	Alfonso X of Castile	95	Juan Perón
10	65	Ferdinand I, Holy Roman Emperor	94	Augusto Pinochet

Table S6. List of local historical figures for IT category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	1952	Julius Caesar	689	Julius Caesar
2	1662	Augustus	647	Augustus
3	1476	Christopher Columbus	616	Leonardo da Vinci
4	893	Galileo Galilei	464	Benito Mussolini
5	758	Benito Mussolini	339	Pope Pius XII
6	753	Cicero	330	Christopher Columbus
7	594	Leonardo da Vinci	326	Michelangelo
8	292	Pliny the Elder	322	Raphael
9	288	Pope Gregory XIII	197	Giuseppe Verdi
10	250	Pope Paul VI	172	Galileo Galilei

Table S7. List of local historical figures for PT category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	91	Getúlio Vargas	109	Ronaldo
2	83	Cristiano Ronaldo	100	Getúlio Vargas
3	74	John VI of Portugal	92	Juscelino Kubitschek
4	71	Luiz Inácio Lula da Silva	91	Rubens Barrichello
5	70	Pedro I of Brazil	90	Joaquim Maria Machado de Assis
6	67	Ferdinand Magellan	89	Fernando Henrique Cardoso
7	66	Maria I of Portugal	82	Luís de Camões
8	64	John I of Portugal	80	José Saramago
9	63	Pedro II of Brazil	79	John VI of Portugal
10	62	Juscelino Kubitschek	77	Oscar Niemeyer

Table S8. List of local historical figures for EL category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	2237	Aristotle	789	Alexander the Great
2	1949	Plato	207	Plato
3	1771	Alexander the Great	167	Aristotle
4	213	Socrates	108	Pericles
5	178	Plutarch	100	Mustafa Kemal Atatürk
6	153	Mustafa Kemal Atatürk	98	Eleftherios Venizelos
7	123	Sophocles	95	Andreas Papandreou
8	93	Aeschylus	94	Muhammad Ali of Egypt
9	86	Euripides	94	Ioannis Kapodistrias
10	84	Ioannis Kapodistrias	93	Plutarch

Table S9. List of local historical figures for DA category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	99	Tycho Brahe	210	Hans Christian Andersen
2	94	Ole Rømer	98	Margrethe II of Denmark
3	93	Christian IV of Denmark	95	N. F. S. Grundtvig
4	86	Margrethe II of Denmark	92	Sren Kierkegaard
5	85	Hans Christian Andersen	89	Christian IV of Denmark
6	84	Frederick IV of Denmark	88	Hans Christian Ørsted
7	80	Frederick II of Denmark	86	Anders Fogh Rasmussen
8	78	John Louis Emil Dreyer	84	Carl Nielsen
9	77	Christian VII of Denmark	83	Christian X of Denmark
10	76	Frederick III of Denmark	82	Niels Bohr

Table S10. List of local historical figures for SV category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	2284	Carl Linnaeus	326	Carl Linnaeus
2	125	August Strindberg	151	Ingmar Bergman
3	98	Alfred Nobel	146	Charles XII of Sweden
4	94	Gustav I of Sweden	116	Astrid Lindgren
5	93	Gustav III of Sweden	100	August Strindberg
6	86	Charles XII of Sweden	98	Carl XVI Gustaf of Sweden
7	82	Gustavus Adolphus of Sweden	92	Evert Taube
8	72	Carl XVI Gustaf of Sweden	89	Jan Myrdal
9	71	Charles XI of Sweden	88	Carl Jonas Love Almqvist
10	67	Charles IX of Sweden	83	Gustav I of Sweden

Table S11. List of local historical figures for PL category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	864	Pope John Paul II	693	Pope John Paul II
2	94	Catherine the Great	296	Frédéric Chopin
3	88	David Ben-Gurion	135	Catherine the Great
4	80	Casimir III the Great	98	David Ben-Gurion
5	72	Nathan Alterman	95	Bolesaw III Wrymouth
6	69	Lech Walesa	94	Andrzej Wajda
7	66	Lech Kaczyński	93	Nathan Alterman
8	63	Frédéric Chopin	91	Gerhart Hauptmann
9	60	Henryk Sienkiewicz	88	Anton Denikin
10	58	Sigismund I the Old	83	Lech Kaczyński

Table S12. List of local historical figures for HU category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	93	János Szentágothai	100	Stephen I of Hungary
2	91	Stephen I of Hungary	99	Sándor Petöfi
3	87	Lajos Kossuth	94	Kati Kovács
4	86	Miklós Réthelyi	93	Miklós Horthy
5	80	Béla IV of Hungary	92	Attila József
6	79	Louis I of Hungary	89	Sándor Weöres
7	75	Sándor Petöfi	86	Theodor Herzl
8	67	Miklós Horthy	83	Lajos Kossuth
9	56	Theodor Herzl	81	Miklós Radnóti
10	53	Andrew II of Hungary	77	János Kodolányi

Table S13. List of local historical figures for RU category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	1123	Immanuel Kant	334	Vladimir Putin
2	1022	Vladimir Lenin	274	Isaac Asimov
3	156	Peter the Great	198	Peter the Great
4	130	Mikhail Gorbachev	171	Vladimir Lenin
5	101	Pyotr Ilyich Tchaikovsky	127	Yuri Gagarin
6	97	Yuri Gagarin	109	Igor Stravinsky
7	97	Alexander Pushkin	100	Menachem Begin
8	91	Vladimir Putin	99	Dmitri Mendeleev
9	89	Nikita Khrushchev	96	Aleksander Griboyedov
10	88	Alexander II of Russia	95	Shimon Peres

Table S14. List of local historical figures for HE category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	1094	Mary (mother of Jesus)	580	Mary (mother of Jesus)
2	724	Saint Peter	240	Saint Peter
3	138	John the Baptist	171	John the Baptist
4	99	Yitzhak Rabin	99	Saint George
5	95	Yigal Amir	99	Yitzhak Rabin
6	84	Josephus	96	Ariel Sharon
7	81	Tom Segev	92	Benjamin Netanyahu
8	75	Ariel Sharon	85	Ehud Barak
9	65	Benjamin Netanyahu	82	Roni Dalumi
10	54	Herod the Great	79	Moshe Dayan

Table S15. List of local historical figures for TR category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	973	Paul the Apostle	252	Paul the Apostle
2	925	Herodotus	231	Suleiman the Magnificent
3	133	Strabo	172	Mehmed the Conqueror
4	117	Mehmed the Conqueror	169	Selim I
5	106	Suleiman the Magnificent	142	Abdul Hamid II
6	96	Abdul Hamid II	111	Julian (emperor)
7	93	Pausanias (geographer)	90	Recep Tayyip Erdoğan
8	83	İsmet İnönü	87	Adnan Menderes
9	79	Selim I	85	Lucian
10	79	Hesiod	84	Blent Ecevit

Table S16. List of local historical figures for AR category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	2282	Jesus	943	Jesus
2	1735	Muhammad	499	Muhammad
3	629	Moses	291	Ali
4	426	Saddam Hussein	219	Saddam Hussein
5	424	Ptolemy	181	Muammar Gaddafi
6	329	Augustine of Hippo	143	Hannibal
7	328	Ali	128	Saladin
8	196	Umar	128	Anwar Sadat
9	147	Anwar Sadat	117	Hosni Mubarak
10	134	Euclid	108	Yasser Arafat

Table S17. List of local historical figures for FA category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	110	Zoroaster	229	Cyrus the Great
2	101	Darius I	99	Zoroaster
3	100	Mahmoud Ahmadinejad	98	Mohammad Reza Pahlavi
4	97	Mohammad Reza Pahlavi	97	Mohammad Khatami
5	96	Rez Shh	96	Mir-Hossein Mousavi
6	94	Cyrus the Great	95	Ruhollah Khomeini
7	92	Ferdowsi	94	Naser al-Din Shah Qajar
8	90	Ruhollah Khomeini	93	Ali Khamenei
9	89	Naser al-Din Shah Qajar	92	Mohammad Mosaddegh
10	86	Mohammad Khatami	91	Ardashir I

Table S18. List of local historical figures for HI category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	168	Ashoka	126	Ashoka
2	106	Mahatma Gandhi	108	Akbar
3	100	Benazir Bhutto	99	Indira Gandhi
4	91	Vikramditya	98	Mahadevi Varma
5	90	Shivaji	96	Sanjeev Kumar
6	89	Jawaharlal Nehru	93	Amitabh Bachchan
7	88	Akbar	91	Premchand
8	87	Indira Gandhi	90	Dayananda Saraswati
9	86	Adi Shankara	89	Jaishankar Prasad
10	85	Vishnu Prabhakar	86	Adi Shankara

Table S19. List of local historical figures for MS category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	96	Mahathir Mohamad	100	Mahathir Mohamad
2	85	Najib Razak	99	Najib Razak
3	84	P. Ramlee	98	Anwar Ibrahim
4	81	Tunku Abdul Rahman	93	Mizan Zainal Abidin of Terengganu
5	79	Abdullah Ahmad Badawi	92	Sudirman Arshad
6	77	Muhyiddin Yassin	91	Tunku Abdul Rahman
7	74	Abdul Razak Hussein	90	Siti Nurhaliza
8	62	Anwar Ibrahim	89	Abdullah Ahmad Badawi
9	58	Hussein Onn	88	Abdul Taib Mahmud
10	37	Mizan Zainal Abidin of Terengganu	84	P. Ramlee

Table S20. List of local historical figures for TH category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	100	Chulalongkorn	100	Sirindhorn
2	97	Vajiravudh	98	Sirikit
3	96	Mongkut	97	Thaksin Shinawatra
4	94	Buddha Yodfa Chulaloke	94	Taksin
5	92	Nangklao	91	Pridi Banomyong
6	91	Thaksin Shinawatra	90	Yingluck Shinawatra
7	90	Damrong Rajanubhab	88	Srinagarindra
8	89	Taksin	86	Samak Sundaravej
9	88	Plaek Phibunsongkhram	82	Vajiralongkorn
10	87	Prajadhipok	80	Chao Keo Naovarat

Table S21. List of local historical figures for VI category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	91	Ho Chi Minh	98	Ho Chi Minh
2	71	Ngo Dinh Diem	97	Gia Long
3	62	Minh Mng	96	Minh Mng
4	46	Gia Long	94	Nguyen Hue
5	44	Bo i	86	Le Loi
6	22	Le Loi	84	Tran Hung Dao
7	15	Nhat Linh	83	Vo Nguyen Giap
8	N/A	N/A	82	Tu Duc
9	N/A	N/A	81	Le Thánh Tông
10	N/A	N/A	80	Trung Sisters

Table S22. List of local historical figures for ZH category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	375	Mao Zedong	306	Cao Cao
2	285	Confucius	243	14th Dalai Lama
3	244	Chiang Kai-shek	234	Liu Bei
4	197	Qin Shi Huang	192	Qin Shi Huang
5	186	Emperor Wu of Han	191	Kangxi Emperor
6	135	Cao Cao	188	Zhuge Liang
7	129	Hongwu Emperor	179	Qianlong Emperor
8	119	Qianlong Emperor	154	Mao Zedong
9	119	Kangxi Emperor	147	Hongwu Emperor
10	94	Sun Yat-sen	146	Sun Yat-sen

Table S23. List of local historical figures for KO category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	100	Gojong of the Korean Empire	114	Gojong of the Korean Empire
2	98	Kim Il-sung	106	Kim Il-sung
3	95	Sejong the Great	100	Park Chung-hee
4	94	Park Chung-hee	99	Kim Dae-jung
5	93	Taejong of Joseon	97	Roh Moo-hyun
6	92	Syngman Rhee	95	Sejong the Great
7	91	Yeongjo of Joseon	94	Taejo of Goryeo
8	90	Kim Dae-jung	93	Kim Young-sam
9	89	Seonjo of Joseon	92	Jeongjo of Joseon
10	86	Taejo of Joseon	90	Syngman Rhee

Table S24. List of local historical figures for JA category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	154	Toyotomi Hideyoshi	346	Tokugawa Ieyasu
2	153	Tokugawa Ieyasu	266	Toyotomi Hideyoshi
3	108	Hirohito	252	Hirohito
4	97	Oda Nobunaga	233	Oda Nobunaga
5	86	Emperor Meiji	140	Junichiro Koizumi
6	81	Minamoto no Yoritomo	131	Shinzō Abe
7	76	Junichiro Koizumi	112	Tsunku
8	73	Emperor Tenmu	106	Emperor Meiji
9	70	Natsume Sōseki	100	Koxinga
10	69	Akihito	97	Osamu Tezuka

Table S25. List of local historical figures for WR category. Here Θ_A is the ranking score of the algorithm A defined in Eq.(1).

	Θ_A	PageRank local figures	Θ_A	2DRank local figures
1	1686	Joseph Stalin	529	J. R. R. Tolkien
2	842	Constantine the Great	477	Joseph Stalin
3	564	Gautama Buddha	276	Nelson Mandela
4	506	Sigmund Freud	241	Genghis Khan
5	405	Diocletian	195	Sigmund Freud
6	351	Justinian I	191	Martina Navratilova
7	219	J. R. R. Tolkien	186	Constantine the Great
8	203	Genghis Khan	173	Justinian I
9	138	Avicenna	127	Nikola Tesla
10	129	Rumi	123	Kublai Khan

A Weighted Correlation Index for Rankings with Ties

Sebastiano Vigna*
Università degli Studi di Milano, Italy

April 28, 2014

Abstract

Understanding the correlation between two different scores for the same set of items is a common problem in information retrieval, and the most commonly used statistics that quantifies this correlation is Kendall's τ. However, the standard definition fails to capture that discordances between items with high rank are more important than those between items with low rank. Recently, a new measure of correlation based on *average precision* has been proposed to solve this problem, but like many alternative proposals in the literature it assumes that there are *no ties* in the scores. This is a major deficiency in a number of contexts, and in particular while comparing centrality scores on large graphs, as the obvious baseline, indegree, has a very large number of ties in web and social graphs. We propose to extend Kendall's definition in a natural way to take into account weights in the presence of ties. We prove a number of interesting mathematical properties of our generalization and describe an $O(n \log n)$ algorithm for its computation. We also validate the usefulness of our weighted measure of correlation using experimental data.

1 Introduction

In information retrieval, one is often faced with different scores[1] for the same set of items. This includes the lists of documents returned by different search engines and their associated relevance scores, the lists of query recommendation returned by different algorithms, and also the score associated to each node of a graph by different centrality measures (e.g., indegree and Bavelas's closeness [1]).

In most of the literature, the scores are assumed to be without ties, thus inducing a *ranking* of the elements. At that point, correlation statistics such as Spearman's rank correlation coefficient [24] and Kendall's τ [12] can be used to evaluate the similarity of the rankings. Spearman's correlation coefficient is equivalent to the traditional linear correlation coefficient computed on ranks of items. Kendall's τ, instead, is proportional to the number of pairwise adjacent swaps needed to convert one ranking into the other.

For a number of reasons, Kendall's τ has become a standard statistic to compare the correlation between two ranked lists. Such reasons include fast computation ($O(n \log n)$, where n is the length of the list, using Knight's algorithm [14]), and the existence of a variant that takes care of ties [13].

The explicit treatment of ties is of great importance when comparing global *exogenous* relevance scores in large collections of web documents. The baseline of such scores is indegree—the number of documents containing hypertextual link to a given document. More sophisticated approaches include Katz's index [10], PageRank [21], and countless variants. Due to the highly skewed indegree distribution, a very large number of documents share the same indegree, and the same happens of many other scores: it is thus of uttermost importance that the evaluation of correlation takes into account ties as first-class citizens.

On the other hand, Kendall's τ has some known problems that motivated the introduction of several weighted variants. In particular, a striking difference often emerges between the anecdotal evidence of the top elements by different scores being almost identical, and the τ value being quite low. This is due to a known phenomenon: the scores of important items tend to be highly correlated in all reasonable rankings, whereas most of the remaining items are ranked in slightly different ways, introducing a large amount of noise, yielding a low τ value.

*Sebastiano Vigna has been supported by the EU-FET grant NADINE (GA 288956).

[1] We purposely and consistently use "score" to denote real numbers associated to items, and "rank" to denote ordinal positions. The two terms are used somewhat interchangeably in the literature, but in this paper the distinction is important as we assume that scores of different items can be identical.

This problem motivates the definition of correlation statistics that consider more important correlation between highly ranked items. In particular, recently Yilmaz, Aslam and Robertson introduced a statistics, named *AP (average precision) correlation* [27], which aims at considering more important swaps between highly ranked items. The need for such a measure is very well motivated in the introduction of their paper, and we will not repeat here their detailed discussion.

In this paper, we aim at providing a measure of correlation in the same spirit of the definition oh Yilmaz, Aslam and Robertson, but taking smoothly ties into account. We will actually define a general notion of weighting for Kendall's τ, and develop its mathematical properties. Since it is important that such a statistics is computable on very large data sets, we will provide a generalization of Knight's algorithm that can be applied whenever the weighting depends additively or multiplicatively on a weight assigned to each item. The same algorithm can be used to compute AP correlation in time $O(n \log n)$.

All data and software used in this paper are available as part of the LAW software library under the GNU General Public License.[2]

2 Related work

Shieh [23] wrote the one of the first papers proposing a generic weighting of Kendall's τ. She assumes from the very start that there are no ties, and assign to the exchange between i and j a weight w_{ij}. Her motivation is the *fidelity evaluation of software packages for structural engineering*, in which a set of variables is ranked in two different ways, and one would like to emphasize agreement on the most important ones. In particular, she concentrates on weights given by the product of two weights associated with the elements participating in the exchange. Our work can be seen as a generalization of her approach, albeit we combine weights differently.

Kumar and Vassilvitskii [16] study a definition that extends Shieh's taking into account *position weights* and *similarity between elements*. Again, they assume that ties are broken arbitrarily, which is an unacceptable assumption if large sets of elements have the same score. Fagin, Kumar and Sivakumar [6] use instead *penalty weights* to apply Kendall's τ just to the top k elements of two ranked lists (with no ties). Exchanges partially or completely outside the top k elements obtain different weights.

Finally, the recent quoted work of Yilmaz, Aslam and Robertson [27] on AP correlation is the closest to ours in motivation and methodology, albeit targeted at ranked lists with no ties.

We remark that analogous research exists in association with Spearman's correlation: Iman and Conover [9], for example, study the usage of *Savage scores* [22] instead of ranks when comparing ranked lists. Savage scores for a ranked list of n elements are given by $\sum_{j=i}^{n} 1/j$, where i is the rank (starting at one) of an element. Spearman's correlation applied to Savage scores considers more important elements at the top of a ranked list.

Recently, Webber, Moffat and Zobel [26] have described a similarity measure for *indefinite rankings*—rankings that might have different lengths and contain different elements. Their work has some superficial resemblance with the approach of [16, 27] and our work, as it give preminence to differences at the top of ranked lists, but it is not technically a correlation index, as it is based on measuring overlaps of infinite lists, rather than on exchanges. Thus, the basic condition for a correlation index (i.e., that inverting the list one obtains the minimum possible correlation, usually standardized to -1), is not even expressible in their framework. Moreover, their measure, being defined on infinite lists, needs the fundamental assumption that the weight function applied to overlaps must be *summable*; in particular, they make importance decrease exponentially. As we will discuss in Section 4.2, and verify experimentally in Section 6, such a choice is a reasonable framework for very short lists, or when only very first elements are relevant (e.g., because one is modelling user behavior), but it would completely flatten the results of our correlation index on large examples, depriving it from its discriminatory power, even if the weight function would decrease just quadratically.

A fascinating proposal, entirely orthogonal to the ones we discussed, is the idea of weighting Kemeny's distance between permutations proposed by Farnoud and Milenkovic [7]. In this proposal, Kemeny's distance between two permutations π and σ is characterized as the minimum number of *adjacent transpositions* (i.e., transpositions of the form $(i\, i+1)$) that turn π into σ. At this point, one can define a *weight* associated to each adjacent transposition, and by assigning larger weights to adjacent transpositions with smaller indices one can make differences in the top part of the permutations more important than differences in the bottom part. The right notion of weighted distance turns out to be the minimum sum of weights of a sequence of adjacent transposition that turn π into σ. The interesting property of this approach is that avoids the need for a *ground truth* (an intrinsic notion of importance of an element), which is necessary,

[2] `http://law.di.unimi.it/`

implicitly or explicitly, to weigh an exchange in the approaches of [23, 27] and in the one discussed in this paper. The main drawback, presently, is that even in the presence of weighting functions that are monotonically decreasing in i the time necessary to compute the distance is $O(n^2)$ instead of $O(n \log n)$. It is also necessary more tuning to extend the distance to the case of ties, and to turn in this case the distance into a proper correlation index with range in $[-1 \mathinner{..} 1]$.

3 Motivation

The need for weighted correlation measures in the case of ranked list has been articulated in detail in previous work. Here we will focus on the case of centrality measures for graphs. Consider the graph of English Wikipedia[3], which has about four million nodes and one hundred million arcs. In this graph, 99.95% of the nodes have the same indegree of some other node—for example, more than a half million node has indegree one. It is clearly mandatory, when computing the correlation of other scores with indegree, that ties are taken into consideration in a systematic way (e.g., not broken arbitrarily).

We will consider four other commonly used scores based on the adjacency matrix A of the Wikipedia graph. One is PageRank [21], which is defined by

$$\mathbf{1}/n \sum_{k \geq 0} (\alpha \bar{A})^k,$$

where $\alpha \in [0 \mathinner{..} 1)$ is a *damping factor* and $\bar{A}$ is a stochasticization of A: every row not entirely made of zeroes is divided by its sum, so to have ℓ_1 norm one.

The other index we consider is Katz's [10], which is defined by

$$\mathbf{1} \sum_{k \geq 0} (\alpha A)^k,$$

where $\alpha \in [0 \mathinner{..} 1/\lambda)$ is an attenuation factor depending on λ, the dominant eigenvalue of A [19]. In both cases, we take α in the middle of the allowed interval (using different values does not change the essence of what follows, unless they are extreme).

A different kind of score is provided by Bavelas's *closeness*. The closeness of x is defined by

$$\frac{1}{\sum_{d(y,x)<\infty} d(y,x)},$$

where $d(-,-)$ denotes the usual graph distance. Note that we have to eliminate nodes at infinite distance to avoid zeroing all scores. By definition the closeness of a node with indegree zero is zero. Finally, we consider *harmonic centrality* [2], a modified version of Bavelas's closeness designed for directed graphs that are not strongly connected; the harmonic centrality of x is defined by

$$\sum_{y \neq x} \frac{1}{d(y,x)}.$$

These scores provide an interesting mix: indegree is an obvious baseline, and entirely local. PageRank and Katz are similar in their definition, but the normalization applied to A makes the scores quite different (at least in theory). Finally, closeness and harmonic centrality are of a completely different nature, having no connection with dominant eigenvectors or Markov chains.

Our first empirical observation is that, looking just at the very top pages of Wikipedia (Table 1; entries in boldface are unique to the list they belong to, here and in the following), we perceive these scores as almost identical, except for closeness, which displays almost random values. The latter behavior is a known phenomenon: nodes that are almost isolated obtain a very high closeness score (this is why harmonic centrality was devised). We note also that harmonic centrality has a slightly different slant, as it is the only ranking including Latin, Europe, Russia and the Catholic Church in the top 20.

The problem is that these facts are not reflected in any way in the values of Kendall's τ shown in Table 3. If we exclude closeness, with the exception of the correlation between indegree and Katz, all other correlation value fail to surpass the 0.9 threshold, usually considered the threshold for considering two rankings equivalent [25]. Actually, they

[3]More precisely, a specific snapshot of Wikipedia that will be made public by the author. The graph does not contain template pages.

Indegree	PageRank	Katz	Harmonic	Closeness
United States	United States	United States	United States	**Kharqan Rural District**
List of sovereign states	Animal	List of sovereign states	United Kingdom	**Talageh-ye Sofla**
Animal	List of sovereign states	United Kingdom	World War II	**Talageh-ye Olya**
England	France	France	France	**Greatest Remix Hits (Whigfield album)**
France	Germany	Animal	Germany	**Suzhou HSR New Town**
Association football	Association football	World War II	Association football	**Suzhou Lakeside New City**
United Kingdom	England	England	English language	**Mepirodipine**
Germany	India	Association football	China	**List of MPs ... M–N**
Canada	United Kingdom	Germany	Canada	**List of MPs ... O–R**
World War II	Canada	Canada	India	**List of MPs ... S–T**
India	Arthropod	India	**Latin**	**List of MPs ... U–Z**
Australia	Insect	Australia	World War I	**List of MPs ... J–L**
London	World War II	London	England	**List of MPs ... C**
Japan	Japan	Italy	Italy	**List of MPs ... F–I**
Italy	Australia	Japan	**Russia**	**List of MPs ... A–B**
Arthropod	Village	New York City	**Europe**	**List of MPs ... D–E**
Insect	Italy	English language	Australia	**Esmaili-ye Sofla**
New York City	Poland	China	**European Union**	**Esmaili-ye Olya**
English language	English language	Poland	**Catholic Church**	**Levels of organization (ecology)**
Village	**Nationa Reg. of Hist. Places**	World War I	London	**Jacques Moeschal (architect)**

Table 1: Top 20 pages of the English version of Wikipedia following five different centrality measures.

are below the threshold 0.8, under which we are supposed to see considerable changes. The correlation of closeness with harmonic centrality, moreover, is even more pathological: it is the *largest* correlation.

An obvious observation is that, maybe, the score is lowered by a large discordance in the rest of the rankings. Table 2 tries to verify this intuition by listing the top pages that are associated with the Wordnet category "scientist" in the Yago2 ontology data [8]. These pages have considerably lower score (their rank is below 300), yet the first three rankings are almost identical. Harmonic centrality is still slightly different (Linnaeus is absent, and actually ranks 21), which tells us that the Kendall's τ is not giving completely unreasonable data. Nonetheless, closeness continues to provide apparently random results.

We have actually to delve deep into Wikipedia, beyond rank 100 000 using the category "cocktail" to see that, finally, things settle down (Table 5). While closeness still displays a few quirks, the rankings start to stabilize.

To understand what happens in the very low-rank region, in Table 4 we provide Kendall's τ as in Table 3, but *restricting the computation to nodes of indegree 1 and 2*. As it is immediately evident, after stabilization the low-rank region is fraught with noise and all correlation values drop significantly.

The very high correlation between closeness and harmonic centrality is, actually, not strange: on the nodes reachable from giant connected component of our Wikipedia snapshot (89% of the nodes) they agree almost exactly, as closeness is the reciprocal of a denormalized *arithmetic* mean, whereas harmonic centrality is the reciprocal of a denormalized *harmonic* mean [2]. Even if the remaining 11% of the nodes is completely out of place, making closeness useless, Kendall's τ tells us that it should be interchangeable with harmonic centrality. At the same time, Kendall's τ tells us that indegree is very different from PageRank, which again goes completely against our empirical evidence.

In the rest of the paper, we will try to approach in a systematic manner these problems by defining a new weighted correlation index for scores with ties.

4 Definitions and Tools

In his 1945 paper about ranking with ties [13], Kendall, starting from an observation of Daniels [4], reformulates his correlation index using a definition similar in spirit to that of an inner product, which will be the starting point of our proposal: we consider two real-valued vectors $\boldsymbol{r}$ and $\boldsymbol{s}$ (to be thought as scores) with indices in $[n]$; then, let us define

$$\langle \boldsymbol{r}, \boldsymbol{s} \rangle := \sum_{i<j} \operatorname{sgn}(r_i - r_j) \operatorname{sgn}(s_i - s_j),$$

Indegree	PageRank	Katz	Harmonic	Closeness
Carl Linnaeus	Carl Linnaeus	Carl Linnaeus	Aristotle	**Noël Bernard (botanist)**
Aristotle	Aristotle	Aristotle	Albert Einstein	**Charles Coquelin**
Thomas Jefferson	Thomas Jefferson	Thomas Jefferson	Thomas Jefferson	**Markku Kivinen**
Margaret Thatcher	Charles Darwin	Albert Einstein	Charles Darwin	**Angiolo Maria Colomboni**
Plato	Plato	Charles Darwin	Thomas Edison	**Om Prakash (historian)**
Charles Darwin	Albert Einstein	Karl Marx	**Alexander Graham Bell**	**Michel Mandjes**
Karl Marx	Karl Marx	Plato	**Nikola Tesla**	**Kees Posthumus**
Albert Einstein	Pliny the Elder	Margaret Thatcher	**William James**	**F. Wolfgang Schnell**
Vladimir Lenin	Vladimir Lenin	Vladimir Lenin	Isaac Newton	**Christof Ebert**
Sigmund Freud	Johann Wolfgang von Goethe	Isaac Newton	Karl Marx	**Reese Prosser**
J. R. R. Tolkien	Margaret Thatcher	Ptolemy	**Charles Sanders Peirce**	**David Tulloch**
Johann Wolfgang von Goethe	Ptolemy	Johann Wolfgang von Goethe	Noam Chomsky	**Kim Hawtrey**
Spider-Man	Sigmund Freud	Pliny the Elder	**Enrico Fermi**	**Patrick J. Miller**
Pliny the Elder	Isaac Newton	Benjamin Franklin	Ptolemy	**Mikel King**
Benjamin Franklin	Benjamin Franklin	J. R. R. Tolkien	**John Dewey**	**Albert Perry Brigham**
Leonardo da Vinci	J. R. R. Tolkien	Thomas Edison	Johann Wolfgang von Goethe	**Gordon Wagner (economist)**
Isaac Newton	Immanuel Kant	Sigmund Freud	**Bertrand Russell**	**George Henry Chase**
Ptolemy	Leonardo da Vinci	Immanuel Kant	Plato	**Charles C. Horn**
Immanuel Kant	**Pierre André Latreille**	Leonardo da Vinci	**John von Neumann**	**Paul Goldstene**
George Bernard Shaw	Thomas Edison	Noam Chomsky	Vladimir Lenin	**Robert Stanton Avery**

Table 2: Top 20 pages of Wikipedia following five different centrality measures and restricting pages to Yago2 Wordnet category "scientist". The global rank of these items is beyond 300.

	Ind.	PR	Katz	Harm.	Cl.
Indegree	1	0.75	0.90	0.62	0.55
PageRank	0.75	1	0.75	0.61	0.56
Katz	0.90	0.75	1	0.70	0.62
Harmonic	0.62	0.61	0.70	1	0.92
Closeness	0.55	0.56	0.62	0.92	1

Table 3: Kendall's τ between Wikipedia centrality measures.

where

$$\operatorname{sgn}(x) := \begin{cases} 1 & \text{if } x > 0; \\ 0 & \text{if } x = 0; \\ -1 & \text{if } x < 0. \end{cases}$$

Indices of score vectors in summations belong to $[n]$ throughout the paper. Note that

$$\langle \boldsymbol{r}, \alpha \boldsymbol{s} \rangle = \langle \alpha \boldsymbol{r}, \boldsymbol{s} \rangle = \operatorname{sgn}(\alpha) \langle \boldsymbol{r}, \boldsymbol{s} \rangle,$$

which reminds of the analogous property for inner products, and that $\langle \boldsymbol{r}, - \rangle = \langle -, \boldsymbol{r} \rangle = 0$ if $\boldsymbol{r}$ is constant. Following the analogy, we can define

$$\|\boldsymbol{r}\| := \sqrt{\langle \boldsymbol{r}, \boldsymbol{r} \rangle},$$

so

$$\|\alpha \boldsymbol{r}\| = |\operatorname{sgn}(\alpha)| \cdot \|\boldsymbol{r}\|.$$

The norm thus defined measures the "untieness" of $\boldsymbol{r}$: it is zero if and only if $\boldsymbol{r}$ is a constant vector, and it has maximum value $\sqrt{n(n-1)/2}$ when all components of $\boldsymbol{r}$ are distinct.

We can now define Kendall's τ between two vectors $\boldsymbol{r}$ and $\boldsymbol{s}$ with nonnull norm as a normalized inner product, in a way formally identical to cosine similarity:

$$\tau(\boldsymbol{r}, \boldsymbol{s}) := \frac{\langle \boldsymbol{r}, \boldsymbol{s} \rangle}{\|\boldsymbol{r}\| \cdot \|\boldsymbol{s}\|}. \tag{1}$$

We recall that if $\boldsymbol{r}$ and $\boldsymbol{s}$ have no ties, the definition reduces to the classical "normalized difference of concordances and discordances", as the denominator is exactly $n(n-1)/2$. The definition above is exactly that proposed by Kendall [13], albeit we use a different formalism.

The form of (1) suggests that to obtain a weighted correlation index it would be natural to define a *weighted* inner product

$$\langle \boldsymbol{r}, \boldsymbol{s} \rangle_w := \sum_{i<j} \operatorname{sgn}(r_i - r_j) \operatorname{sgn}(s_i - s_j) w(i, j),$$

	Ind.	PR	Katz	Harm.	Cl.
Indegree	1	0.31	0.63	0.24	0.06
PageRank	0.31	1	0.27	0.10	0.10
Katz	0.63	0.27	1	0.50	0.20
Harmonic	0.24	0.10	0.50	1	0.65
Closeness	0.06	0.10	0.20	0.65	1

Table 4: Kendall's τ between Wikipedia centrality measures, restricted to nodes of indegree 1 and 2.

where $w(-,-) : [n] \times [n] \to \mathbf{R}_{\geq 0}$ is some nonnegative weight function. We would have then a new norm $\|\boldsymbol{r}\|_w = \sqrt{\langle \boldsymbol{r}, \boldsymbol{r} \rangle_w}$ and a new correlation index

$$\tau_w(\boldsymbol{r}, \boldsymbol{s}) := \frac{\langle \boldsymbol{r}, \boldsymbol{s} \rangle_w}{\|\boldsymbol{r}\|_w \cdot \|\boldsymbol{s}\|_w}.$$

Note that still $\langle \boldsymbol{r}, - \rangle_w = \langle -, \boldsymbol{r} \rangle_w = 0$ if $\boldsymbol{r}$ is constant.

We say that two score vectors $\boldsymbol{r}$ and $\boldsymbol{s}$ are *equivalent* if $\operatorname{sgn}(r_i - r_j) = \operatorname{sgn}(s_i - s_j)$, *opposite* if $\operatorname{sgn}(r_i - r_j) = -\operatorname{sgn}(s_i - s_j)$ for all i and j.

Lemma 1 *We have*

$$\sum_{i<j} |\operatorname{sgn}(r_i - r_j) \operatorname{sgn}(s_i - s_j)| w(i,j) \leq \|\boldsymbol{r}\|_w \|\boldsymbol{s}\|_w. \tag{2}$$

A sufficient condition for equality to hold is that the two vectors are equivalent or opposite.

Proof. Let $R_{ij} = |\operatorname{sgn}(r_i - r_j)|$ and $S_{ij} = |\operatorname{sgn}(s_i - s_j)|$. Then,

$$\begin{aligned}
&\Big(\sum_{i<j} R_{ij} S_{ij} w(i,j)\Big)^2 \\
&= \Big(\sum_{i<j} R_{ij}^2 S_{ij}^2 w(i,j)^2\Big) + \Big(\sum_{\substack{i<j,k<\ell \\ i\neq k \vee j \neq \ell}} R_{ij} S_{ij} R_{k\ell} S_{k\ell} w(i,j) w(k,\ell)\Big) \\
&\leq \Big(\sum_{i<j} R_{ij}^2 S_{ij}^2 w(i,j)^2\Big) + \Big(\sum_{\substack{i<j,k<\ell \\ i\neq k \vee j \neq \ell}} R_{ij}^2 S_{k\ell}^2 w(i,j) w(k,\ell)\Big) \\
&= \Big(\sum_{i<j} R_{ij}^2 w(i,j)\Big)\Big(\sum_{i<j} S_{ij}^2 w(i,j)\Big) = \|\boldsymbol{r}\|_w^2 \|\boldsymbol{s}\|_w^2.
\end{aligned}$$

Note that if the vectors are equivalent or opposite then

$$R_{ij} S_{ij} R_{k\ell} S_{k\ell} = R_{ij}^2 S_{k\ell}^2$$

for all i, j, k and ℓ, so we obtain equality. ∎

We now prove a fundamental Cauchy–Schwartz-like inequality:

Theorem 1 $|\langle \boldsymbol{r}, \boldsymbol{s} \rangle_w| \leq \|\boldsymbol{r}\|_w \|\boldsymbol{s}\|_w$. *A sufficient condition for equality to hold is that the two vectors are equivalent or opposite. The condition is necessary if w is strictly positive and $|\langle \boldsymbol{r}, \boldsymbol{s} \rangle_w| \neq 0$.*

Proof. The first two statements are immediate from Lemma 1, as

$$|\langle \boldsymbol{r}, \boldsymbol{s} \rangle_w| \leq \sum_{i<j} |\operatorname{sgn}(r_i - r_j) \operatorname{sgn}(s_i - s_j)| w(i,j)$$

and in the case of equivalent or opposite vectors we have equality. On the other hand, if we let $R_{ij} = \operatorname{sgn}(r_i - r_j)$ and $S_{ij} = \operatorname{sgn}(s_i - s_j)$ the chain of equalities and inequalities at the beginning of the proof of Lemma 1 continues to be true. To have equality, however, assuming that w is strictly positive we must have

$$R_{ij} S_{ij} R_{k\ell} S_{k\ell} w(i,j) w(k,\ell) = R_{ij}^2 S_{k\ell}^2 w(i,j) w(k,\ell)$$

for all i, j, k and ℓ, that is,

$$R_{ij}S_{ij}R_{k\ell}S_{k\ell} = R_{ij}^2 S_{k\ell}^2.$$

Now, since $|\langle \boldsymbol{r}, \boldsymbol{s}\rangle_w| \neq 0$ there must be a pair $\bar{\imath}, \bar{\jmath}$ such that $R_{\bar{\imath}\bar{\jmath}} \neq 0$ and $S_{\bar{\imath}\bar{\jmath}} \neq 0$. Letting $\sigma = R_{\bar{\imath}\bar{\jmath}}S_{\bar{\imath}\bar{\jmath}}$ we have

$$R_{k\ell}S_{k\ell} = \sigma S_{k\ell}^2$$

and

$$R_{ij}S_{ij} = \sigma R_{ij}^2$$

for all i, j, k and ℓ. In particular, if $R_{k\ell} = 0$ we have necessarily $S_{k\ell} = 0$, and *vice versa*. If $R_{k\ell} \neq 0$, then $S_{k\ell} = \sigma R_{k\ell}$, which completes the proof. ∎

Another application of Lemma 1 gives the triangular inequality:

Theorem 2 $\|\boldsymbol{r} + \boldsymbol{s}\|_w \leq \|\boldsymbol{r}\|_w + \|\boldsymbol{s}\|_w$.

Proof.

$$\begin{aligned}
&\|\boldsymbol{r} + \boldsymbol{s}\|_w^2 = \langle \boldsymbol{r} + \boldsymbol{s}, \boldsymbol{r} + \boldsymbol{s}\rangle_w \\
&= \sum_{i<j} \operatorname{sgn}(r_i + s_i - r_j - s_j)^2 w(i,j) \\
&= \sum_{i<j} |\operatorname{sgn}(r_i + s_i - r_j - s_j)|^2 w(i,j) \\
&\leq \sum_{i<j} (|\operatorname{sgn}(r_i - r_j)| + |\operatorname{sgn}(s_i - s_j)|)^2 w(i,j) \\
&= \langle \boldsymbol{r}, \boldsymbol{r}\rangle_w + \langle \boldsymbol{s}, \boldsymbol{s}\rangle_w + 2\sum_{i<j} |\operatorname{sgn}(r_i - r_j)\operatorname{sgn}(s_i - s_j)| w(i,j) \\
&\leq \|\boldsymbol{r}\|_w^2 + \|\boldsymbol{s}\|_w^2 + 2\|\boldsymbol{r}\|_w\|\boldsymbol{s}\|_w \\
&= (\|\boldsymbol{r}\|_w + \|\boldsymbol{s}\|_w)^2.
\end{aligned}$$

∎

The triangular inequality has a nice combinatorial interpretation: adding score vectors can only *decrease* the amount of "untieness". There is no way to induce in a sum vector more untieness than the amount present in the summands.

Finally, an easy application of Theorem 1 shows that τ_w is sensible and works as expected:

Theorem 3 *Let $w : [n] \times [n] \to \mathbf{R}$ be a nonnegative weight function. The following properties hold for every score vector $\boldsymbol{t}$ and for every $\boldsymbol{r}$, $\boldsymbol{s}$ with nonnull norm:*

- *if $\boldsymbol{t}$ is constant, $\|\boldsymbol{t}\|_w = 0$;*
- *$-1 \leq \tau_w(\boldsymbol{r}, \boldsymbol{s}) \leq 1$;*
- *if $\boldsymbol{r}$ and $\boldsymbol{s}$ are equivalent, $\tau_w(\boldsymbol{r}, \boldsymbol{s}) = 1$;*
- *if $\boldsymbol{r}$ and $\boldsymbol{s}$ are opposite, $\tau_w(\boldsymbol{r}, \boldsymbol{s}) = -1$;*

Moreover, if w is strictly positive:

- *if $\|\boldsymbol{t}\|_w = 0$, $\boldsymbol{t}$ is constant;*
- *if $\tau_w(\boldsymbol{r}, \boldsymbol{s}) = 1$, $\boldsymbol{r}$ and $\boldsymbol{s}$ are equivalent;*
- *if $\tau_w(\boldsymbol{r}, \boldsymbol{s}) = -1$, $\boldsymbol{r}$ and $\boldsymbol{s}$ are opposite.*

As a result, if w is strictly positive and we obtain correlation ± 1 the equivalence classes formed by tied scores are necessarily in a size-preserving bijection that is monotone decreasing on the scores.

Indegree	PageRank	Katz	Harmonic	Closeness
Martini (cocktail)	Martini (cocktail)	Irish coffee	Irish coffee	**Magie Noir**
Piña colada	Caipirinha	Caipirinha	Caipirinha	**Batini (drink)**
Mojito	Mojito	Martini (cocktail)	Kir (cocktail)	**Scorpion bowl**
Caipirinha	Piña colada	Piña colada	Martini (cocktail)	**Poinsettia (cocktail)**
Cuba Libre	Irish coffee	Kir (cocktail)	Piña colada	Irish coffee
Irish coffee	Kir (cocktail)	Mojito	Mojito	Caipirinha
Singapore Sling	Cosmopolitan (cocktail)	Mai Tai	Beer cocktail	Kir (cocktail)
Manhattan (cocktail)	Manhattan (cocktail)	Cuba Libre	Shaken, not stirred	Martini (cocktail)
Windle (sidecar)	IBA Official Cocktail	Singapore Sling	Pisco Sour	Piña colada
Cosmopolitan (cocktail)	Beer cocktail	Long Island Iced Tea	Mai Tai	Mojito
Mai Tai	Mai Tai	Shaken, not stirred	Spritz (alcoholic beverage)	Beer cocktail
IBA Official Cocktail	Singapore Sling	Beer cocktail	Long Island Iced Tea	Shaken, not stirred
Kir (cocktail)	Cuba Libre	Manhattan (cocktail)	Sazerac	Mai Tai
Shaken, not stirred	**Tom Collins**	Cosmopolitan (cocktail)	Fizz (cocktail)	Spritz (alcoholic beverage)
Beer cocktail	Long Island Iced Tea	Windle (sidecar)	Flaming beverage	Pisco Sour
Pisco Sour	Sour (cocktail)	Pisco Sour	Cuba Libre	Long Island Iced Tea
Long Island Iced Tea	Shaken, not stirred	White Russian (cocktail)	Wine cocktail	Sazerac
Sour (cocktail)	**Negroni**	IBA Official Cocktail	Singapore Sling	Flaming beverage
White Russian (cocktail)	Flaming beverage	Moscow mule	Moscow mule	Fizz (cocktail)
Vesper (cocktail)	**Lillet**	Vesper (cocktail)	White Russian (cocktail)	Wine cocktail

Table 5: Top 20 pages of Wikipedia following five different centrality measures and restricting pages to Yago2 Wordnet category "cocktail". The global rank of these items is beyond 100 000.

4.1 Decoupling rank and weight

The reader has probably already noticed that the dependence on the weight on the *indices* associated to the elements has no meaning: a trivial request (see, for instance [11]) on a correlation measure is that, like Kendall's τ, it is *invariant by isomorphism*, that is, it does not change if we permute the indices of the vector. This currently doesn't happen because we are using the numbering of the element as *ground truth* to weigh the correlation between $\boldsymbol{r}$ and $\boldsymbol{s}$. While there is nothing bad in principle (we can stipulate that elements are indexed in order of importance using some external source of information), we think that a more flexible approach decouples the problem of the ground truth from the problem of weighting. We thus define the *ranked-weight* product

$$\langle \boldsymbol{r}, \boldsymbol{s} \rangle_{\rho,w} := \sum_{i<j} \operatorname{sgn}(r_i - r_j) \operatorname{sgn}(s_i - s_j) w(\rho(i), \rho(j)),$$

where $\rho : [n] \to [n] \cup \{\infty\}$ is a ranking function associating with each index a *rank*, the highest rank being zero. We admit the possibility of rank ∞, given that the weight function provides a meaningful value in such a case, to include also the case of *partial ground truths*. The definition of the ranked-weighted product induces, as in (1), a correlation index $\tau_{\rho,w}$, and the machinery we developed applies immediately, as $w(\rho(-), \rho(-))$ is just a different weight function.

What if there is no ground truth to rely on? Our best bet is to use the rankings induced by the vectors $\boldsymbol{r}$ and $\boldsymbol{s}$. Let us denote by $\rho_{\boldsymbol{r},\boldsymbol{s}}$ the ranking defined by ordering elements lexicographically with respect to $\boldsymbol{r}$ and then $\boldsymbol{s}$ in case of a tie (in descending order), and analogously for $\rho_{\boldsymbol{s},\boldsymbol{r}}$ (if two elements are at a tie in both vectors, their can be placed in any order, as their rank does not influence the value of $\tau_{\rho,w}$). We define

$$\tau_{w,\bullet}(\boldsymbol{r}, \boldsymbol{s}) := \frac{\tau_{\rho_{\boldsymbol{r},\boldsymbol{s}},w}(\boldsymbol{r}, \boldsymbol{s}) + \tau_{\rho_{\boldsymbol{s},\boldsymbol{r}},w}(\boldsymbol{r}, \boldsymbol{s})}{2}. \tag{3}$$

The same approach has been used in [27] to make AP correlation symmetric. This is the definition used in the rest of the paper.

4.2 Choosing a weighting scheme

There are of course many ways to choose w. For computational reasons, we will see that it is a good idea to restrict to a class of weighting schemes in which w is obtained by combining additively or multiplicatively a one-argument weighting function $f : [n] \to \mathbf{R}_{\geq 0}$ applied to each element of a pair.

Shieh [23], for instance, combines weights multiplicatively, without giving a motivation. We have, however, two important motivations for *adding* weights. First and foremost, unless weights are scaled in some way that depends on n (which we would like to avoid), the largest weight will be some constant, and then weight will decrease monotonically with importance. As a result, an exchange between the first and the last element would be assigned an extremely low

weight. Second, adding weights paves the way to a natural measure for *top k correlation* [6] by assigning rank ∞ to elements after the first k. The definition of such a measure in the multiplicative case is quite contrived and ends up being case-by-case.

For what matters f, we are particularly interested in the *hyperbolic* weight function.

$$f(r) := \frac{1}{r+1}.$$

This function gives more importance to elements of high rank, and weights zero only pairs in which both index have infinite rank. Using a hyperbolic weight has a number of useful features. First, it reminds the well-motivated weight given to exchanges by AP correlation. Second, it guarantees that as n grows the mass of weight grows indefinitely. Using a function with quadratic decay, for instance, might end up in making the influence of low-rank element vanish too quickly, as it is summable. For the opposite reason, a *logarithmic* decay might fail to be enough discriminative to provide additional information with respect to the standard τ.

We try to make this intuition more concrete in Figure 1, where we display a number of scatter plots showing the correlation between Kendall's τ and the additive weighted τ defined by (3) under different weighting schemes. The left half of the plots correlates all permutations on 12 elements with the identity permutation. The right half correlates all score vectors made of 15 values with skewed distribution (there are $t+1$ elements with score $0 \leq t \leq 4$) with the same vector in descending order. A visual examination of the plots suggests, indeed, that logarithmic weighting restricts too much the possible divergence from Kendall's τ, whereas quadratic weighting ends up in providing answers that are too uncorrelated. We will return to these consideration in Section 6.

5 Computing $\tau_{\rho,w}$

Our motivations come from the study of web and social graphs. It is thus essential that our new correlation measure can be evaluated efficiently. We now describe a generalization of Knight's algorithm [14] that makes it is possible to compute $\tau_{\rho,w}$ in time $O(n \log n)$ under some assumptions on w. Our first observation is that, similarly to the unweighted case, each pair of indices i, j with $i < j$ belongs to one of five subsets; it can be

- a *joint tie*, if $r_i = r_j$ and $s_i = s_j$;
- a *left tie*, if $r_i = r_j$ and $s_i \neq s_j$;
- a *right tie*, if $r_i \neq r_j$ and $s_i = s_j$;
- a *concordance*, if $\operatorname{sgn}(r_i - r_j)\operatorname{sgn}(s_i - s_j) = 1$;
- a *discordance*, if $\operatorname{sgn}(r_i - r_j)\operatorname{sgn}(s_i - s_j) = -1$.

Let J, L, R, C and D be the overall weight of joint ties, left ties, right ties, concordances and discordances, respectively. Clearly,

$$J + L + R + C + D = \sum_{i<j} w(\rho(i), \rho(j)) = T.$$

The first requirement for our technique to work is that T can be computed easily. This is possible if weights are computed additively or multiplicatively from some single-argument function f. In the additive case,

$$T = \sum_{i<j} \big(f(\rho(i)) + f(\rho(j))\big) = (n-1) \sum_i f(\rho(i)). \tag{4}$$

Also the multiplicative case is easy, as

$$2T = 2 \sum_{i<j} f(\rho(i)) f(\rho(j)) = \left(\sum_i f(\rho(i)) \right)^2 - \sum_i f(\rho(i))^2. \tag{5}$$

The same observation leads to a simple $O(n \log n)$ algorithm to compute L: sort the indices in $[n]$ by $\boldsymbol{r}$, and for each block of consecutive $k > 1$ elements with the same score apply (4) or (5) restricting the indices to the subset. In the same way one can compute R and J.

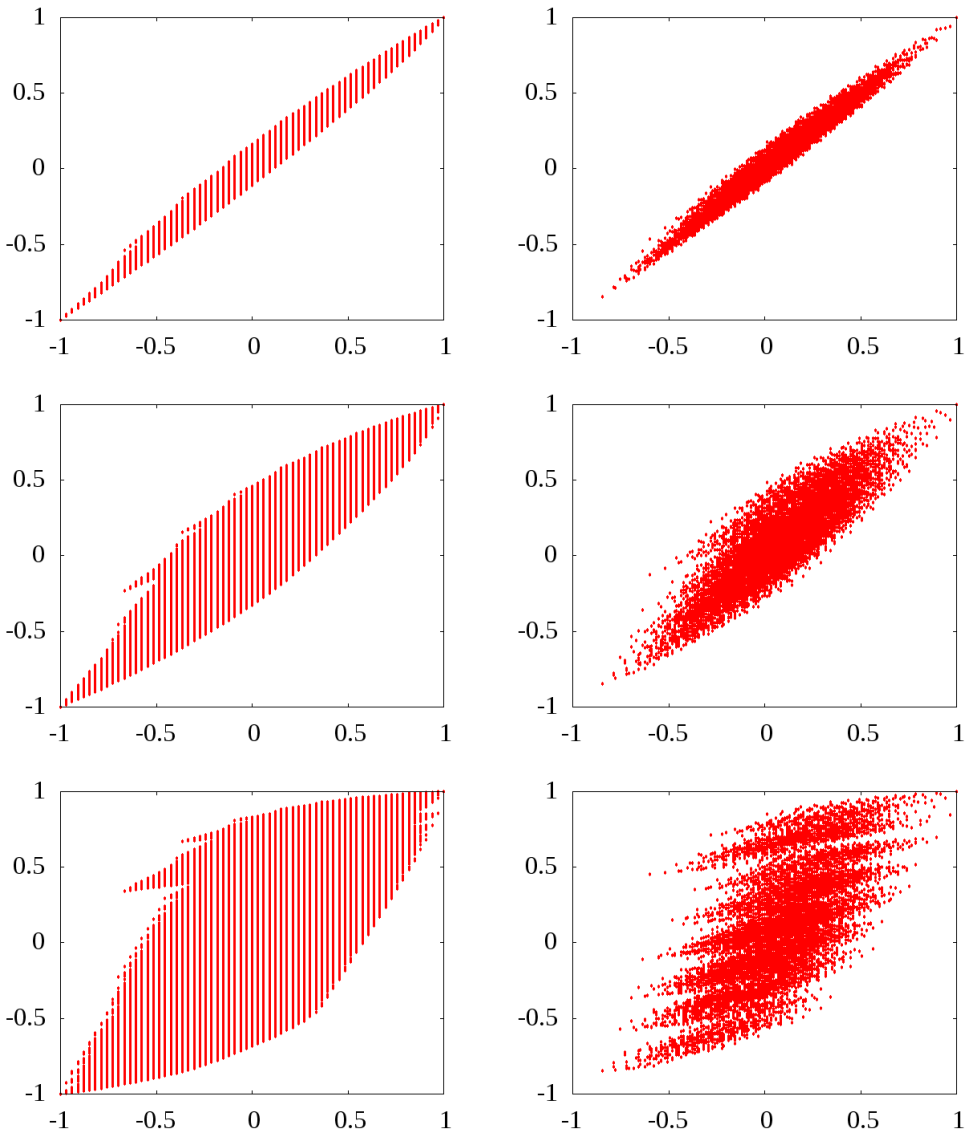


Figure 1: Scatter plots between Kendall's τ and the additive weighted τ. The rows, from top to bottom, represent logarithmic, hyperbolic and quadratic weighting. The plots are generated correlating a permutation of 12 elements versus the identity permutation (left), or a permuted set of scores with skewed distribution w.r.t. the same scores in descending order (right).

We now observe that, as in the unweighted case,

$$\langle \boldsymbol{r}, \boldsymbol{s} \rangle_{\rho,w} = C - D = T - (L + R - J) - 2D.$$

This can be easily seen from the fact that C is given by the total weight T, minus the weight of discordances D, minus the number of ties, joint or not, which is $L + R - J$ (we must avoid to count twice the weight of joint ties, hence the $-J$ term). In particular,

$$\langle \boldsymbol{r}, \boldsymbol{r} \rangle_{\rho,w} = T - L \qquad \langle \boldsymbol{s}, \boldsymbol{s} \rangle_{\rho,w} = T - R,$$

as in this case there are just concordances and all ties are joint.

We are left with the computation of D. The core of Knight's algorithm is an *exchange counter*: an $O(n \log n)$ algorithm that given a list of elements and an order $\preceq$ on the elements of the list computes the number of exchanges that are necessary to $\preceq$-sort the list. The algorithm is a modified MergeSort [15][4]: during the merging phase, whenever an element is moved from the second list to the temporary result list the current number of elements of the first list is added to the number of exchanges. The number of discordances is then equal to the number of exchanges (as we evaluate whether there is a discordance on i and j only for $i < j$).

Our goal is to make this computation weighted: for this to happen, it must be possible to keep track incrementally of a *residual weight* r associated with the first list, and obtain in constant time the weight of the exchanges generated by the movement of an element from the second list.

If weights are computed multiplicatively or additively starting from a single-argument function f this is not difficult: it is sufficient to let r be the sum of the values of f applied to the elements currently in the first list. In the additive case, moving an element i from the second list increases the weight of exchanges by the residual r plus the weight $f(\rho(i))$ multiplied by the length of the first list. In the multiplicative case, we must instead use the weight $f(\rho(i))$ multiplied by the residual r. When we move an element from the first list we update the residual by subtracting its weight.

The resulting recursive procedure (for the additive case) is Algorithm 1. The final layout of the computation of $\tau_{\rho,w}$ is thus as follows:

- Consider a list $\mathscr{L}$ initially filled with the integers $[0 \mathinner{..} n)$.
- Sort stably $\mathscr{L}$ using $\boldsymbol{r}$ as primary key and $\boldsymbol{s}$ as secondary key.
- Compute T and L using $\mathscr{L}$ to enumerate elements in the order defined by $\boldsymbol{r}$ and $\boldsymbol{s}$.
- Apply Algorithm 1 to $\mathscr{L}$ using $\boldsymbol{s}$ to define the order $\preceq$, thus computing D and sorting $\mathscr{L}$ by $\boldsymbol{s}$.
- Compute R using $\mathscr{L}$ to enumerate elements in the order defined by $\boldsymbol{s}$.
- Compute T and put everything together.

The running time of the computation is dominated by the sorting phases, and it is thus $O(n \log n)$.

5.1 The asymmetric case and AP Correlation

It is easy to adapt Algorithm 1 for the case in which $w(i, j)$ is given by a combination of *two* different one-argument functions, one, f, for the left index and one, g, for the right index. The only modification of Algorithm 1 is the replacement of f with g at line 14, so that we combine the residual computed with f with a weight computed with g.

The formulae for computing T can be updated easily for the additive case:

$$T = \sum_{i<j} \big(f(\rho(i)) + g(\rho(j))\big) = \sum_{i \neq 0} i(f(\rho(n-1-i)) + g(\rho(i)))$$

and for the multiplicative case:

$$T = \sum_{i<j} f(\rho(i))g(\rho(j)) = \sum_{i} f(\rho(i)) \sum_{i<j} g(\rho(j)).$$

[4] In principle, any stable algorithm that sorts by comparison could be used. This is particularly interesting as entirely on-disk algorithms, such as *polyphase merge* [15], could be used to count exchanges using constant core memory.

Algorithm 1 A generalization of Knight's algorithm for weighing exchanges.

Input: A list $\mathscr{L}$, a comparison function $\preceq$ for the elements of $\mathscr{L}$, a rank function ρ, and a single-argument weight function f that will be combined additively. e is a global variable initialized to 0 that will contain the weight of exchanges after the call weigh(0, $|\mathscr{L}|$). The procedure works on a sublist specified by its starting index $0 \leq s < |\mathscr{L}|$ and its length ℓ. $\mathscr{T}$ is a temporary list.
Output: the sum of $f(\rho(-))$ on the specified sublist.

```
0   function weigh(s : integer, ℓ : integer)
1     if ℓ = 1 then return f(ρ(ℒ[s])) fi
2     ℓ₀ ← ⌊ℓ/2⌋
3     ℓ₁ ← ℓ − ℓ₀
4     m ← s + ℓ₀
5     r ← weigh(s, ℓ₀)
6     w ← weigh(m, ℓ₁) + r
7     i, j, k ← 0
8     while j < ℓ₀ and k < ℓ₁ do
9       if ℒ[s + j] ⪯ ℒ[m + k] then
10        𝒯[i] = ℒ[s + j++]
11        r ← r − f(ρ(𝒯[i]))
12      else
13        𝒯[i] = ℒ[m + k++]
14        e ← e + f(ρ(𝒯[i])) · (ℓ₀ − j) + r
15      fi
16      i++
17    od
18    for k = ℓ₀ − j − 1, ..., 0 do
19      ℒ[s + i + k] ← ℒ[s + j + k]
20    od
21    for k = 0, ..., i − 1 do ℒ[s + k] ← 𝒯[k] od
22    return w
23  end
```

Both formulae can be computed in linear time using a suitable loop.

Given this setup, it is easy to compute AP correlation: as it can be easily checked from the very definition [27], the AP correlation of $\boldsymbol{r}$ w.r.t. $\boldsymbol{s}$, where both vectors have no ties, is simply $\tau_{w,\rho_{\boldsymbol{s}}}(\boldsymbol{r}, \boldsymbol{s})$, where $\rho_{\boldsymbol{s}}$ is the ranking induced by $\boldsymbol{s}$ and the weight function w is computed additively from two weight functions $f(r) = 0$, $g(r) = 1/r$. In this case, $T = n - 1$, $J = L = R = 0$ (we are under the assumption that there are no ties) and Algorithm 1 can be considerably simplified, as the residual r is always zero.[5]

Algorithm 2 makes explicit the change to the selection statement of Algorithm 1 that is sufficient to compute AP correlation. Since keeping track of the residual is no longer necessary, the recursive function can be further simplified to a recursive procedure that does not return a value. The value e computed by the modified algorithm is all we need to compute AP correlation using the formula $(T - 2e)/T$.

[5]Of course, it is possible to forget that we are computing AP correlation and use the weight matrix just described combined with the machinery of Section 4 to define an "AP correlation with ties". In this case, J, L and R should be computed using the formulae for the asymmetric case, and the probabilistic interpretation would be lost. Such an index would probably give a notion of correlation very similar to τ_{h}, but we find more natural and more in line with Kendall's original definition to introduce the weighted τ as a symmetric index in which both ends of an exchange are relevant in computing the exchange weight.

Algorithm 2 The replacement for lines 9–15 of Algorithm 1 to compute AP correlation.

```
9   if 𝒮[s + j] ⪯ 𝒮[m + k] then
10      𝒯[i] = 𝒮[s + j++]
11  else
12      𝒯[i] = 𝒮[m + k++]
13      e ← e + (ℓ_0 − j)/ρ(𝒯[i])
14  fi
```

	Ind.	PR	Katz	Harm.	Cl.
Indegree	1	0.95	0.98	0.90	0.27
PageRank	0.95	1	0.96	0.92	0.65
Katz	0.98	0.96	1	0.93	0.26
Harmonic	0.90	0.92	0.93	1	0.28
Closeness	0.27	0.65	0.26	0.28	1

Table 6: τ_{h} on Wikipedia.

6 Experiments

We now return to our main motivation—understanding the correlation between centralities on large graph. In this section, we gather the results of a number of computational experiment that help to corroborate our intuition that τ_{h}, the *additive hyperbolic weighted* τ, works as expected. We will find also an interesting surprise along the way.

Note that judging whether a new measure is useful for such a purpose is a difficult task: to be interesting, a new measure must highlight features that were previously undetectable or badly evaluated, but those are exactly those features on which a systematic assessment is problematic.

Table 6 reports the value of τ_{h} on the Wikipedia graph. We finally see data corresponding to the empirical evidence discussed in Section 3: indegree, Katz and PageRank are almost identical, harmonic centrality is highly correlated but definitely less than the previous triple, which matches our empirical observations. Closeness is not close to any ranking (and in particular, not to harmonic centrality) due to its pathological behavior.

There is of course a value that immediately stands out: the suspiciously high correlation (0.65) between closeness and PageRank. We reserve discussing this value for later.

In Table 7 we show the same data for logarithmic and quadratic weights. The intuition we gathered from Figure 1 is fully confirmed: logarithmic weights provides results almost indistinguishable from Kendall's τ (compare with Table 3), and quadratic weighs make the influence of the tail so low that all non-pathological scores collapse.

To gather a better understanding of the behavior of τ_{h} we extended our experiments to two very different datasets: the *Hollywood co-starship graph*, an undirected graph (2 million nodes, 229 million edges) with an edge between two persons appearing in the Internet Movie Data Base if they ever worked together, and a *host graph* (100 million nodes, 2 billion arcs) obtained from a large-scale crawl gathered by the Common Crawl Foundation[6] in the first half of 2012.[7] As (unavoidably anecdotal) empirical evidence we report the top 20 nodes for both graphs.

Table 8 should be compared with Table 10. PageRank and harmonic centrality turns to be less correlated to indegree than Katz in Table 8, and indeed we find many quirk choices in the very top PageRank actors (Ron Jeremy is a famous porn star; Lloyd Kaufman is an independent horror/splatter filmmaker and Debbie Rochon an actress working with him). Harmonic centrality provides unique names such as Malcolm McDowell, Robert De Niro, Anthony Hopkins and Sylvester Stallone, and drops all USA presidents altogether. Kendall's τ values, instead, suggest that PageRank and harmonic centrality are entirely uncorrelated (whereas we find several common items), and that harmonic and closeness centrality should be extremely similar.

We see analogous results comparing Table 9 with Table 11. Here τ_{h} separates in a very strong way harmonic centrality from the first three, and indeed we see a significant difference in the lists, with numerous sites that have a high indegree and appear in at least two of the three lists because of technical or political reasons (`gmpg.org`,

[6] `http://commoncrawl.org/`

[7] The crawl contains 3.53 billion web documents; we are using the associated host graph, which has a node for each host and an arc between two hosts x and y if some page in x points to some page in y. More information about the graph can be found in [18], and the complete host ranking can be accessed at `http://wwwranking.webdatacommons.org/`.

	Ind.	PR	Katz	Harm.	Cl.
Indegree	1	0.76	0.90	0.63	0.55
PageRank	0.76	1	0.76	0.62	0.56
Katz	0.90	0.76	1	0.70	0.62
Harmonic	0.63	0.62	0.70	1	0.91
Closeness	0.55	0.56	0.62	0.91	1

	Ind.	PR	Katz	Harm.	Cl.
Indegree	1	1.00	1.00	1.00	0.22
PageRank	1.00	1	1.00	1.00	0.85
Katz	1.00	1.00	1	1.00	0.18
Harmonic	1.00	1.00	1.00	1	0.07
Closeness	0.22	0.85	0.18	0.07	1

Table 7: The logarithmic (top) and quadratic (bottom) additive τ on Wikipedia.

	Ind.	PR	Katz	Harm.	Cl.
Indegree	1	0.42	0.93	0.55	0.43
PageRank	0.42	1	0.36	0.10	0.18
Katz	0.93	0.36	1	0.61	0.49
Harmonic	0.55	0.10	0.61	1	0.86
Closeness	0.43	0.18	0.49	0.86	1

	Ind.	PR	Katz	Harm.	Cl.
Indegree	1	0.90	0.98	0.91	0.10
PageRank	0.90	1	0.88	0.81	0.64
Katz	0.98	0.88	1	0.92	0.11
Harmonic	0.91	0.81	0.92	1	0.18
Closeness	0.10	0.64	0.11	0.18	1

Table 8: Kendall's τ (top) and τ_{h} (bottom) on the Hollywood co-starship graph.

`rtalabel.org`, `staff.tumblr.com`, `miibeian.gov.cn`, `phpbb.com`) disappearing altogether in favor of sites such as `apple.com`, `amazon.com`, `myspace.com`, `microsoft.com`, `bbc.co.uk`, `nytimes.com` and `guardian.co.uk`, which do not appear in any other list. If we look at Kendall's τ, we should expect PageRank and Katz to give very different rankings, whereas more than half of their top 20 elements are in common.

6.1 PageRank and closeness

It is now time to examine the mysteriously high τ_{h} between PageRank and closeness we found in all our graphs. When we first computed our correlation tables, we were puzzled by its value. The phenomenon is interesting for three reasons: first, it has never been reported—using standard, unweighted indices this correlation is simply undetectable; second, it was known for techniques based on singular vectors [17]; third, we know *exactly* the cause of this correlation, because the only real difference between harmonic and closeness centrality is the score assigned to nodes unreachable from the giant component. We thus expect to discover an unsuspected correlation between the way PageRank and closeness rank these nodes.

To have a visual understanding of what is happening, we created Figure 2, 3 and 4 in the following way: first, we isolated the nodes that are unreachable from the giant component (in the case of Hollywood, which is undirected, these nodes form separate components), omitting nodes which have indegree zero, modulo loops (as all measures give the lowest score to such nodes); then, we sorted the nodes in order of decreasing closeness rank, and plotted for each node

	Ind.	PR	Katz	Harm.	Cl.
Indegree	1	0.71	0.89	0.61	0.54
PageRank	0.71	1	0.66	0.50	0.50
Katz	0.89	0.66	1	0.69	0.59
Harmonic	0.61	0.50	0.69	1	0.86
Closeness	0.54	0.50	0.59	0.86	1

	Ind.	PR	Katz	Harm.	Cl.
Indegree	1	0.91	0.96	0.72	0.20
PageRank	0.91	1	0.90	0.81	0.69
Katz	0.96	0.90	1	0.78	0.15
Harmonic	0.72	0.81	0.78	1	0.35
Closeness	0.20	0.69	0.15	0.35	1

Table 9: Kendall's τ (top) and τ_{h} (bottom) on the on the Common Crawl host graph.

Indegree	PageRank	Katz	Harmonic	Closeness
Shatner, William	Jeremy, Ron	Shatner, William	Sheen, Martin	**Östlund, Claes Göran**
Flowers, Bess	Hitler, Adolf	Sheen, Martin	Clooney, George	**Östlund, Catarina**
Sheen, Martin	**Kaufman, Lloyd**	Hanks, Tom	Jackson, Samuel L.	**von Preußen, Oskar Prinz**
Reagan, Ronald (I)	Bush, George W.	Williams, Robin (I)	Hopper, Dennis	**von Preußen, Georg Friedrich**
Clooney, George	Reagan, Ronald (I)	Clooney, George	Hanks, Tom	**von Mannstein, Robert Grund**
Jackson, Samuel L.	Clinton, Bill (I)	Reagan, Ronald (I)	Stone, Sharon (I)	**von Mannstein, Concha**
Williams, Robin (I)	Sheen, Martin	Willis, Bruce	Brosnan, Pierce	**von der Busken, Mart**
Hanks, Tom	**Rochon, Debbie**	Jackson, Samuel L.	Hitler, Adolf	**van der Putten, Thea**
Jeremy, Ron	**Kennedy, John F.**	Stone, Sharon (I)	**McDowell, Malcolm**	**de la Bruheze, Joel Albert**
Hitler, Adolf	Hopper, Dennis	Freeman, Morgan (I)	Williams, Robin (I)	**de la Bruheze, Emile**
Willis, Bruce	**Nixon, Richard**	Flowers, Bess	**De Niro, Robert**	**te Riele, Marloes**
Clinton, Bill (I)	**Estevez, Joe**	Brosnan, Pierce	Willis, Bruce	**de Reijer, Eric**
Freeman, Morgan (I)	Shatner, William	Douglas, Michael (I)	**Hopkins, Anthony**	**des Bouvrie, Jan**
Hopper, Dennis	Jackson, Samuel L.	Madonna (I)	Madonna (I)	**de Klijn, Judith**
Stone, Sharon (I)	**Stewart, Jon (I)**	Travolta, John	**Lee, Christopher (I)**	**de Freitas, Luís (II)**
Madonna (I)	**Carradine, David (I)**	Hopper, Dennis	Douglas, Michael (I)	**de Freitas, Luís (I)**
Bush, George W.	Asner, Edward	Ford, Harrison (I)	**Sutherland, Donald (I)**	**Zuu, Winnie Otondi**
Harris, Sam (II)	**Zirnkilton, Steven**	Asner, Edward	Freeman, Morgan (I)	**Zuu, Emmanuel Dahngbay**
Brosnan, Pierce	**Colbert, Stephen**	**MacLaine, Shirley**	**Stallone, Sylvester**	**Zilbersmith, Carla**
Travolta, John	**Madsen, Michael (I)**	Clinton, Bill (I)	Ford, Harrison (I)	**Zilber, Mac**

Table 10: Top 20 pages of the Hollywood co-starship graph.

its rank following the other measures (we average ranks on block of nodes so to contain the number of points in the plots). A point of high abscissa in the figures implies a high rank.

All three pictures show clearly that *PageRank assigns a preposterously high rank to to nodes belonging to components that are unreachable from the giant component*. This behavior is actually related to PageRank's *insensitivity to size*: for instance, in a graph made of two components, one of which is a 3-clique and the other a k-clique, the PageRank score of all nodes is $1/(3+k)$, independently of k. This explains why small dense components end up being so highly ranked. The same phenomenon is at work when the community around Lloyd Kaufman's production company (very small and very dense) is attributed such a great importance that its elements make their way to the very top ranks (even if Kaufman himself has indegree rank 219 and Debbie Rochon 1790).

We remark that the gap in rank is lower in the case of Wikipedia, but this is fully in concordance with the higher baseline value of Kendall's τ.

7 Conclusions

In this paper, motivated by the need to understand similarity between score vectors, such as those generated by centrality measures on large graphs, we have defined a weighted version of Kendall's τ starting from its 1945 definition for scores with ties. We have developed the mathematical properties of our generalization following a mathematical similarity with internal products, and showing that for a wide range of weighting schemes our new measure behaves as expected, providing a correlation index between -1 and 1, and hitting boundaries only for opposite or equivalent scores.

Indegree	PageRank	Katz	Harmonic	Closeness
wordpress.org	gmpg.org	wordpress.org	youtube.com	0-p.com
youtube.com	wordpress.org	youtube.com	en.wikipedia.org	0-0-0-0-0-0-0.indahiphop.ru
gmpg.org	youtube.com	gmpg.org	twitter.com	0-0-1.i.tiexue.net
en.wikipedia.org	livejournal.com	en.wikipedia.org	google.com	0-00cigarettes.info
tumblr.com	tumblr.com	tumblr.com	wordpress.org	0-0mos00.hi5.com
twitter.com	en.wikipedia.org	twitter.com	flickr.com	0-0new0-0.hi5.com
google.com	twitter.com	google.com	facebook.com	0-0sunny0-0.hi5.com
flickr.com	networkadvertising.org	flickr.com	apple.com	0-1.i.tiexue.net
rtalabel.org	promodj.com	rtalabel.org	vimeo.com	0-1.sxsy.co
wordpress.com	skriptmail.de	wordpress.com	creativecommons.org	0-2.paparazziwannabe.com
mp3shake.com	parallels.com	mp3shake.com	amazon.com	0-311.cn
w3schools.com	tistory.com	w3schools.com	adobe.com	0-360.rukazan.ru
domains.lycos.com	google.com	creativecommons.org	myspace.com	0-5days.com
staff.tumblr.com	miibeian.gov.cn	staff.tumblr.com	w3.org	0-5days.net
club.tripod.com	phpbb.com	domains.lycos.com	bbc.co.uk	0-5kalibr.pdj.ru
creativecommons.org	blog.fc2.com	club.tripod.com	nytimes.com	0-9-0-4-4-9.promoradio.ru
vimeo.com	tw.yahoo.com	vimeo.com	yahoo.com	0-9-0-9.dbass.ru
miibeian.gov.cn	w3schools.com	miibeian.gov.cn	microsoft.com	0-9-0-9.promodj.ru
facebook.com	wordpress.com	facebook.com	guardian.co.uk	0-9-1125.i.tiexue.net
phpbb.com	domains.lycos.com	phpbb.com	imdb.com	0-9-7-16.software.informer.com

Table 11: Top 20 hosts of the Common Crawl host graph.

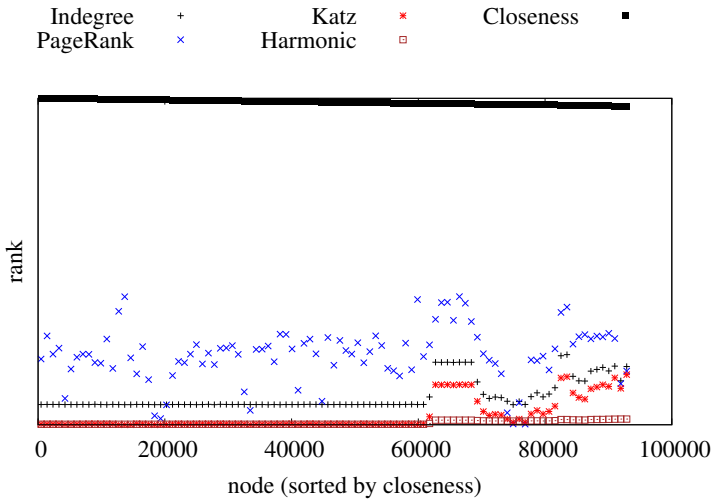


Figure 2: Ranks of components unreachable from the giant component of the Wikipedia graph.

We have then proposed families of weighting schemes that are intuitively appealing, and showed that they can be computed in time $O(n \log n)$ using a generalization of Knight's algorithm, which makes them suitable for large-scale applications. The fact that the main cost of the algorithm is a modified stable sort makes it possible to apply standard techniques to run the algorithm exploiting multicore parallelism, or in distributed environment such as MapReduce [5]. The algorithm can be also used to compute AP correlation [27].

In search for a confirmation of our mathematical intuition, we have then applied our measure of choice τ_{h} (which uses additive hyperbolic weights) to diverse graph such as Wikipedia, the Hollywood co-starship graph and a large host graph, finding that, contrarily to Kendall's τ, τ_{h} provides results that are consistent with an anecdotal examination of lists of top elements.

Our measure was also able to discover a previously unnoticed correlation between PageRank and closeness on small components that are unreachable from the giant component, providing a quantifiable account of the strong bias of PageRank towards small-sized dense communities. This bias might well be the cause of the repeatedly assessed better performance of indegree w.r.t. PageRank in ranking documents [20, 3], as in all our experiments the τ_{h} between PageRank and indegree is above 0.9.

A generalization similar to the one described in this paper can be also applied to *Goodman–Kruskal's* γ, which in the notation of Section 5 is just $(C - D)/(C + D)$. The problem with γ is that the ranking of ties is only implicit (they are simply not counted). Thus, the value of w on tied pairs does not appear at all in the above formula. This "forgetful" behavior can lead to unnatural results, and suggests the Kendall's τ is a better candidate for this approach.

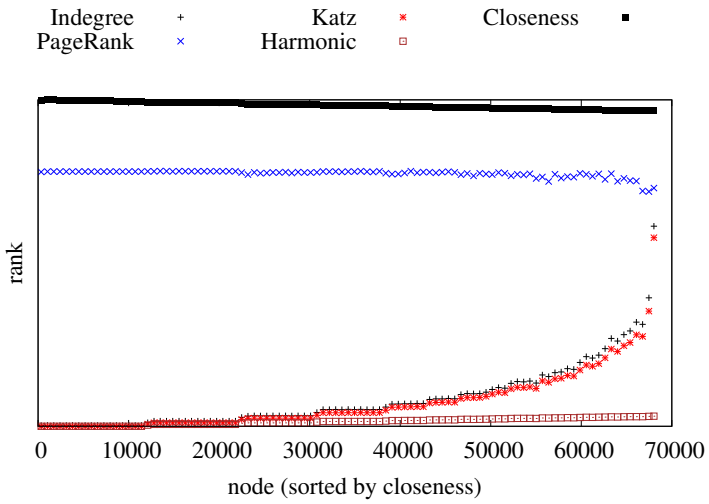


Figure 3: Ranks of components unreachable from the giant component of the Hollywood.

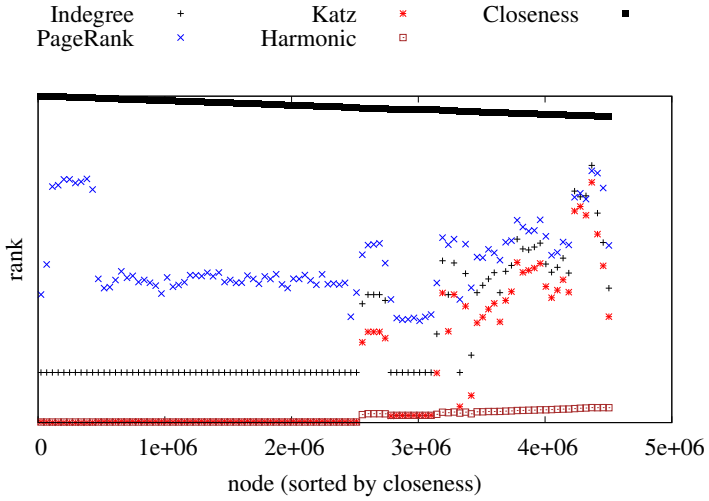


Figure 4: Ranks of components unreachable from the giant component of the Common Crawl host graph.

We remark that an interesting application of additive hyperbolic weighting is that of measuring the correlation between top k lists. By assuming that the rank function ρ returns ∞ after rank k, we obtain a correlation index that weighs zero pairs outside the top k, weights only "by one side" pairs with just one element outside the top k, and weights fully pairs whose elements are within the top k. Formula (3) could provide then in principle a finer assessment than, for instance, the modified Kendall's τ proposed in [6], as the position of each element inside the list, beside the fact that it appears in the top k or not, would be a source of weight. We leave the analysis of such a correlation measure for future work.

References

[1] Alex Bavelas. Communication patterns in task-oriented groups. *J. Acoust. Soc. Am*, 22(6):725–730, 1950.

[2] Paolo Boldi and Sebastiano Vigna. Axioms for centrality. *CoRR*, abs/1308.2140, 2013. To appear in *Internet Mathematics*.

[3] Nick Craswell, David Hawking, and Trystan Upstill. Predicting fame and fortune: PageRank or indegree? In *In Proceedings of the Australasian Document Computing Symposium, ADCS2003*, pages 31–40, 2003.

[4] Henry E. Daniels. The relation between measures of correlation in the universe of sample permutations. *Biometrika*, 33(2):129–135, 1943.

[5] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. In *OSDI '04: Sixth Symposium on Operating System Design and Implementation*, pages 137–150, 2004.

[6] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Comparing top k lists. *SIAM Journal on Discrete Mathematics*, 17(1):134–160, 2003.

[7] Farzad Farnoud and Olgica Milenkovic. Aggregating rankings with positional constraints. In *Proc. IEEE Information Theory Workshop (ITW)*, Seville, Spain, 2013.

[8] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[9] Ronald L. Iman and W. J. Conover. A measure of top-down correlation. *Technometrics*, 29(3):351–357, 1987.

[10] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[11] John G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[12] Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[13] Maurice G. Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.

[14] William R. Knight. A computer method for calculating Kendall's tau with ungrouped data. *Journal of the American Statistical Association*, 61(314):436–439, June 1966.

[15] Donald E. Knuth. *Sorting and Searching*, volume 3 of *The Art of Computer Programming*. Addison-Wesley, second edition, 1997.

[16] Ravi Kumar and Sergei Vassilvitskii. Generalized distances between rankings. In *Proceedings of the 19th International Conference on World Wide Web*, pages 571–580. ACM, 2010.

[17] Ronny Lempel and Shlomo Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks*, 33(1):387–401, 2000.

[18] Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, and Christian Bizer. Graph structure in the web — Revisited, or a trick of the heavy tail. In *WWW'14 Companion*, pages 427–432. International World Wide Web Conferences Steering Committee, 2014.

[19] Carl D. Meyer. *Matrix analysis and applied linear algebra*. Society for Industrial and Applied Mathematics, 2000.

[20] Marc Najork, Hugo Zaragoza, and Michael J. Taylor. HITS on the web: how does it compare? In Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando, editors, *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 471–478. ACM, 2007.

[21] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report SIDL-WP-1999-0120, Stanford Digital Library Technologies Project, Stanford University, 1998.

[22] I. Richard Savage. Contributions to the theory of rank order statistics—the two-sample case. *The Annals of Mathematical Statistics*, 27(3):590–615, 1956.

[23] Grace S. Shieh. A weighted kendall's tau statistic. *Statistics & Probability Letters*, 39(1):17–24, 1998.

[24] Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101, 1904.

[25] Ellen M. Voorhees. Evaluation by highly relevant documents. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82. ACM, 2001.

[26] William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst*, 28(4):20:1–20:38, 2010.

[27] Emine Yilmaz, Javed A. Aslam, and Stephen Robertson. A new rank correlation coefficient for information retrieval. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 587–594. ACM, 2008.

Local Ranking Problem on the BrowseGraph

Michele Trevisiol[*†] trevisiol@acm.org
Luca Maria Aiello[†] alucca@yahoo-inc.com
Paolo Boldi[§] boldi@di.unimi.it
Roi Blanco[†] roi@yahoo-inc.com

[†]Yahoo Labs Barcelona, Spain
[*]Web Research Group Universitat Pompeu Fabra Barcelona, Spain
[§]Univ. degli Studi di Milano Milano, Italy

ABSTRACT

The "Local Ranking Problem" (LRP) is related to the computation of a centrality-like rank on a *local* graph, where the scores of the nodes could significantly differ from the ones computed on the *global* graph. Previous work has studied LRP on the hyperlink graph but never on the *BrowseGraph*, namely a graph where nodes are webpages and edges are browsing transitions. Recently, this graph has received more and more attention in many different tasks such as ranking, prediction and recommendation. However, a webserver has only the browsing traffic performed on its pages (*local BrowseGraph*) and, as a consequence, the local computation can lead to estimation errors, which hinders the increasing number of applications in the state of the art. Also, although the divergence between the local and global ranks has been measured, the possibility of *estimating* such divergence using only local knowledge has been mainly overlooked. These aspects are of great interest for online service providers who want to gauge their ability to correctly assess the importance of their resources only based on their local knowledge, and by taking into account real user browsing fluxes that better capture the actual user interest than the static hyperlink network. We study the LRP problem on a *BrowseGraph* from a large news provider, considering as subgraphs the aggregations of browsing traces of users coming from different domains. We show that the distance between rankings can be accurately predicted based only on structural information of the local graph, being able to achieve an average rank correlation as high as 0.8.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; E.1 [**Data Structures**]: Graphs and Networks

Keywords

Local Ranking Problem, BrowseGraph, PageRank, Centrality Algorithms, Domain-specific Browsing Graphs

1. INTRODUCTION

The ability to identify the online resources that are perceived as important by the users of a website is crucial for online service providers. Metrics to estimate the importance of the page from the structure of online links between them are widely used: algorithms that compute the *centrality* of the nodes in a network, such as PageRank [24], HITS [17] and SALSA [19], have been employed extensively in the last two decades in a vast variety of applications. Born and spread in conjunction with the growth of the Web, they can determine a value of importance of a page from the complex network of links that surrounds it. More recently, centrality metrics have been applied to *browsing graphs*, (also referred to as *BrowseGraphs* [22, 28, 27]) where nodes are webpages and edges represent the transitions made by the users who navigate the links between them. Differently from the hyperlink networks, this data source provides the analyst a way of studying directly the dynamics of the navigational patterns of users who consume online content. Also, unlike hyperlinks, browsing traces account for the variation of consumption patterns in time, for instance in the case of online news were articles tend to become rapidly stale. Comparative studies have shown that centrality-based algorithms applied over *BrowseGraphs* provide higher-quality rankings compared to standard hyperlink graphs [23, 22].

Most centrality measures aim at estimating the importance of a node, using information coming from the *global* knowledge of the graph topology—potentially the addition of new nodes and edges, can have a cascade effect on the centrality values of all other nodes in the network. This fact entails high computational and storage cost for big networks. More critically, there are some situations in which a global computation on the entire graph is unfeasible, for example when the information about the entire network is unavailable or if only an estimation for specific web pages is required. This is an important limitation in many real-world scenarios, where the graphs at hand are often very large (Web scale) and, most importantly, their topology is not fully known. This practical issue raises the problem of how well one can estimate the actual centrality value of a node by knowing only a local portion of the graph. This is known as the *Local Ranking Problem* (LRP) [10].

One of the questions behind LRP is whether it is possible to estimate efficiently the PageRank score of a web page using only a small subgraph of the entire Web [9]. In other words, if one starts from a small graph around a page of interest and extends it with external nodes and arcs (*i.e.*, those belonging to the whole graph), how fast will one ob-

serve the computed scores converging to the real values of PageRank?

We extend this line of work in the context of browsing graphs. For the first time we study the LRP on the *BrowseGraph* and shed some light on the bias that PageRank incurs, when estimating the centrality score of nodes in a *BrowseGraph*, when only partial information about the graph is available. To achieve that, we monitor the browsing traffic of the news portal and we extract different browsing subgraphs induced by the browsing traces of users coming from different *domains*, such as search engines (*e.g.*, Google, Yahoo, Bing) and social networks (*e.g.*, Facebook, Twitter, Reddit). In this setting, the local *BrowseGraphs* are the subgraphs induced by the different domains, and the global *BrowseGraph* is the one built using indistinctly all the navigation logs of the news portal. We describe and evaluate models that tell apart a subgraph from the others just by looking at the behavior of a random surfer that navigates through their links. The results show how it is possible to recognize the graph using only the very first few nodes visited by the users, because the graphs are very different among them (even if they are extracted from the same big log of the news portal). The implication of this experiment is two-fold: first it highlights how navigation patters of the users differ among these subgraphs. Second, we learn that it is possible to infer the user domain of origin from the very first browsing steps. This capability enables several types of services, including user profiling [12], web site optimization [31], user engagement estimation [18], and cold-start recommendation [27], even when the referrer URL is not available (*e.g.*when the user comes from mobile social media applications or URL shortening services).

Once we show that the subgraphs are different enough, we proceed to perform more involved experiments that we call "Growing Balls". We examine the behavior of the PageRank computed on the local and the global graphs. In order to study how the local PageRank converges to the global one, we apply some strategies of incremental addition ("growing") of external nodes to these subgraphs ("balls").

Finally, we build on these findings by setting up a prediction experiment that, for the first time, tackles the task of estimating the reliability of the PageRank computed locally. We measure *how much* the local PageRank diverges from the global one using only structural features of the local graph, usually available to the local service provider.

To sum up, the main contributions of this work are the following:

- We study the *LRP* on a large-scale *BrowseGraph* built from a very popular news website. To the best of our knowledge we are the first to tackle this problem on the increasingly popular *BrowseGraph* [27, 28, 12, 22]. We present an analysis of the convergence of the PageRank on the local graph to the global one, by incrementally expanding the local graph in a snowball fashion.
- We tackle the problem of discovering the referrer domain of a user session, when this information is missing or hidden. We show that this is possible using a random surfer model, that is able to tell the referrer domain with high accuracy, just after the very first browsing transitions.
- We show that an accurate estimation of the distance between the local and global PageRank can be obtained looking at the structural properties of the local graph, such as degree distribution or assortativity.

The remainder of the paper is organized as follows. In §2, we overview relevant prior work in the area and in §3 we describe our dataset and the extraction of the browsing graphs. In §4 we analyze the (sub-)graphs and we highlight their differences. In §5 we study the LRP problem on the *BrowseGraph* and compare the approximation accuracy of different graph expansion strategies. In §6 we present the prediction experiment of the PageRank errors of the local graph. Last, in §7 we wrap up and highlight possible extensions to the work.

2. RELATED WORK

This work encompasses two main different research areas that we introduce shortly. Our focus is the *Local Ranking Problem* but our contribution relates also to previous work on browsing log data, especially the ones that investigate or make use of centrality-based algorithms.

Local Ranking Problem

The *Local Ranking Problem* (LRP) was first introduced by Chen *et al.* [10] in 2004, who addressed the problem to approximate/update the PageRank of individual nodes, without performing a large-scale computation on the entire graph. They proposed an approach that can tackle this problem by including a moderate number of nodes in the local neighborhood of the original nodes. Furthermore, Davis and Dhillon [14] estimated the global PageRank values of a local network using a method that scales linearly with the size of the local domain. Their goal was to rank webpages in order to optimize their crawling order, something similar to what was done by Cho *et al.* [13] who instead selected the top-ranked pages first. However, this latter strategy results to be in contrast with Boldi *et al.* [6], as they found that crawling first the pages with highest global PageRank actually perform worse, if the purpose is fast convergence to the real (global) rank values. In this work, we partial expand the local graph with the neighboring nodes with highest (local) PageRank showing an initial improvement on the convergence speed. In 2008 the problem was reconsidered by Bar-Yossef and Mashiach [3], where they simplified the problem calculating a local *Reverse PageRank* proving that it is more feasible and computationally cheaper, as the reverse natural graphs tend to have low in-degree maintaining a fast PageRank convergence. Bressan and Pretto [9] proved that, in the general case, an efficient local ranking algorithm does not exist, and in order to compute a *correct* ranking it is necessary to visit at least a number of nodes linear in the size of the input graph. They also raised some of the research questions tackled in our paper that we discuss in Section 6.1. They reinforce their findings in later work [8], where they summarized two key factors necessary for efficient local PageRank computations: *exploring the graph non-locally* and *accepting a small probability error*. These two constraints are also considered in this paper in order to perform our experiments on the browsing graphs. When one wants to estimate PageRank in a local graph, the problem of the missing information is tackled in various ways. In [3, 9] for example, the authors make use of a model called *link server* (also known as *remote connectivity server* [5]), that responds to any query about a given node with all the in-coming and out-going edges and

relative nodes. This approach, with the knowledge about the LRP, allows to estimate the PageRank ranking, or even the score, with the relative costs. A similar problem was studied by Andersen *et al.* [2], where their goal was to compute the PageRank contributions in a local graph motivated by the problem of detecting link-spam: given a page, its PageRank contributors are the pages that contribute most to its rank; contributors are used for spam detection since you can quickly identify the set of pages that contribute significantly to the PageRank of a suspicious page.

The problem we consider here is different and largely unexplored, because we are studying the PageRank of the different subgraphs based on user browsing patterns.

BrowseGraph

In recent years a large number of studies of user browsing traces have been conducted. Specifically, in the last years there was a surge of interest in the *BrowseGraph*, a graph where the nodes are web pages and the edges represent the transitions from one page to another made by the navigation of the users. Characterizing the browsing behavior of users is a valuable source of information for a number of different tasks, ranging from understanding how people's search behaviors differ [32], ranking webpages through search trails [1, 33] or recommending content items using past history [29]. A comparison between the standard hyperlink graph, based on the structure of the network, with the browse graph built by the users' navigation patterns, has been made by Liu *et al.* [22, 23]. They compared centrality-based algorithms like PageRank [24], TrustRank [15], and BrowseRank [22], on both types of graphs. The results agree on the higher quality of ranking based on the browse graph, because it is a more reliable source; they also tried out a combination of the two graphs with very interesting outcomes. The user browsing graph and related PageRank-like algorithms have been exploited to rank different types of items including images [28, 12], photostreams [11], and predicting users demographic [16] or optimizing web crawling [21]. Trevisiol *et al.* [28] made a comparison between different ranking techniques applied to the Flickr *BrowseGraph*. Chiarandini *et al.* [12] found strong correlations between the type of user's navigation and the type of external Referrer URL. Hu *et al.* [16] have shown that demographic information of the users (*e.g.*, age and gender) can be identified from their browsing traces with good accuracy. The *BrowseGraph* has been used also for recommending sequences of photos that users often like to navigate in sequence, following a collaborative filtering approach [11]. In order to implement an efficient news recommender the user's taste have to be considered as they might change over time. Indeed, studying the users browsing patterns, Liu *et al.* [20] showed that more recent clicks have a considerably higher value to predict future actions than the historical browsing record. Finally, Trevisiol *et al.* [27] exploited the *BrowseGraph* in order to build some user models in the news domain, and recommend the next article the user is going to visit. They introduced the concept of *ReferrerGraph*, that is a *BrowseGraph* built with sessions that are generated by the same referrer domain. Even if the purposes of our work are very different, we construct the *ReferrerGraphs* in the same way in order to be in-line with their investigation.

To the best of our knowledge there is no work in the state of the art that tackles the *Local Ranking Problem* on a browsing graphs with the prediction task that we perform and describe in this paper.

3. DATASET

For the purpose of this study, we took a sample of Yahoo News network's[1] user-anonymized log data collected in 2013. In this section we summarize how we built the dataset and the graphs, but the reader may refer to the aforementioned paper for further details. The data is comprised by a large number of pageviews, which are represented as plain text files that contain a line for each HTTP request satisfied by the Web server. For each pageview in the dataset, we gathered the following fields:

$$\langle BCookie, Time, ReferrerURL, CurrentURL, UserAgent \rangle$$

The *BCookie* is an anonymized identifier computed from the browser cookie. This information allowed us to re-construct the navigation session of the different users. *CurrentURL* and *ReferrerURL* represent, respectively, the current page the user is visiting and the page the user visited before arriving at the destination page. Note that the *Referrer URL* could belong to any domain, *e.g.*, it may be external to the Yahoo News network. The *User-Agent* identifies the user's browser, an information that we used to filter out Web crawlers, and *Timestamp* indicates when the page was visited. All the data were anonymized and aggregated prior to building the browsing graphs. After applying the filtering steps described above, our sample contains approximately 3.8 million unique pageviews and 1.88 billion user transitions.

3.1 Session Identification and Characteristics

The *BrowseGraph* is a graph whose nodes are web pages, and whose edges are the browsing transitions made by the users. To build it we extract the transitions of users from page to page, and in order to preserve the user behavior (that could vary over time), we group pageviews into *sessions*. We split the activity of a single user, taking the *BCookie* as an identifier, into different sessions when either of these two conditions holds:

- **Timeout:** the inactivity between two pageviews is longer than 25 minutes.
- **External URL:** if a user leaves the news platform and returns from an external domain, the current session ends even if previous visits are within the 25 minute threshold.

Moreover, each news article of the dataset is annotated with a high-level *category* manually assigned by the editors.

3.2 Subgraphs Based on Session Referrer URL

We aim to compare the PageRank scores of the nodes between the full *BrowseGraph*, computed with all the Yahoo News logs, and a subgraph that represents the local graph. This is a way to simulate a real-world scenario in which a service provider knowns only the users navigation logs inside its network (subgraph) while the external navigations are unknown (full *BrowseGraph*). Since it is not possible to use the full Web browsing log, we perform a simulation

[1]We considered a number of different subdomains like *Yahoo news*, *finance*, *sports*, *movies*, *travel*, *celebrity*, *etc.*

Subgraphs	Nodes	Edges	Density	%GCC
Google	142,646	779,185	$3.8 \cdot 10^{-5}$	0.93
Yahoo	101,116	404,378	$3.9 \cdot 10^{-5}$	0.95
Bing	61,531	255,464	$6.7 \cdot 10^{-5}$	0.91
Homepage	60,287	335,836	$9.2 \cdot 10^{-5}$	0.99
Facebook	21,060	70,266	$1.5 \cdot 10^{-4}$	0.95
Twitter	4,206	7,080	$4.0 \cdot 10^{-4}$	0.87
Reddit	2,445	4,868	$8.1 \cdot 10^{-4}$	0.95

Table 1: Size of the extracted subgraphs. Note that there is not a strict relation between the size of the subgraph and the amount of browsing traffic generated in it.

using different subgraphs extracted from the same *BrowseGraph* that represent the local graphs of different providers. In order to do that, we extract from the *BrowseGraph* of the Yahoo News dataset various subgraphs built with sessions of users generated by the same Referrer URL. It has been shown [27] that a *BrowseGraphs* constructed in this way contain very different users sessions in terms of content consumed (nodes visited). In particular we consider users accessing the news portal directly from the homepage, that is the main entry point for regular news consumption, and in addition, from a number of domains that fall outside the Yahoo News network: *search engines* (Google, Yahoo, Bing), and *social networks* (Facebook, Twitter, Reddit). For each source domain we extract a subgraph from the overall *BrowseGraph*, by considering only the browsing sessions whose initial *R*eferrer URL matches that domain. For example, if a user clicks on a link referring to our network that has been posted on Twitter, her *R*eferrer URL will be the Twitter page where she found the link. Next, we consider all the following pageviews belonging to the same session of the user, as being a part of the *twitter-subgraph*, given that all of them have been reached through Twitter. We applied the same procedure for all the sources defined before, and finally, we obtained a weighted graph for each different external URL, where the *Weight* accounts for the number of times a user has navigated from the source page to the destination page. On Table 1 a summary with the size of the graphs (in terms of number of nodes and edges) and with their structure is shown. It is interesting to see that all the graphs, even presenting very different size, are very well connected (%GCC between 0.87 and 0.99).

4. REFERRER GRAPHS ANALYSIS

In this section we describe some analysis on these *ReferrerGraphs*, proving that they are consistently different not only in term of nodes and content but also in term of navigation patterns of the users. We also propose an experiment to understand how much the graphs are distinguishable.

4.1 Subgraphs comparison

We consider the seven subgraphs extracted from the main news portal graph with the procedure discussed in §3. Browsing patterns generated by different types of audiences, can lead to different pieces of news pages to emerge as the most central ones in the *BrowseGraph*. To check that, we ran the PageRank algorithm on each of the (weighted) subgraphs, and for every pair of subgraphs we compared the scores obtained on their common nodes, using Kendall's τ distance. The intersection between the node sets of the networks is always large enough to allow us to compute the τ on the intersection only (> 1000 nodes in the case with less overlap). Kendall's τ will provide a clear measure of how much the importance of the same set of nodes varies among different subgraphs. When the ranking between two subgraphs differs greatly (*i.e.*, low Kendall's τ), it is an indication that they either show different content (*i.e.*, webpages) or that the collective browsing behaviour in the two graphs privileged different sets of pages.

Table 2 reports on the cross-distance among the subgraphs and also with respect to the full graph using Kendall's τ. Interestingly, most of the similarity values tend to be very low (<0.3), confirming the hypothesis that the user's interests are tightly related to the domain where they come from. Some of these similarities, however, are considerably higher, remarkably the ones between the three subgraphs that are originated from search engines traffic, *i.e.*, Bing, Google and Yahoo, which yield the most similar rankings of pages (>0.5). However, for the purpose of this work we expect to find a difference among the subgraphs in order to use them as local *BrowseGraph* and study the LRP with the full graph (*i.e.*, *BrowseGraph* made with the entire news log).

4.2 Random Surfer

In §4.1 we showed how users coming from different sources (*i.e.*, referrer domains) behave differently in terms of content discovery and, as a consequence, the importance of the news articles vary significantly among the different *BrowseGraphs*. It has been shown how the referrer domain might be extremely useful to characterize user sessions [12], to estimate user engagement [18] or to perform cold-start recommendation [27]. However, the user's referrer URL is not always visible and, in many cases, it is hidden or masked by services or clients. For instance, any Twitter or mail client (*i.e.*, third-party application) shows an empty referrer URL in the web logs. A similar situation happens with the widespread URL-shortening services (*e.g.*, `Bitly.com`), that mask the original Web page the user is coming from. Nonetheless, in all these cases, a provider could make use of her knowledge of the user's trail, to identify automatically the source where the user started her navigation in the local graph. As we have shown, the referrer URL might be useful to characterize the interest of the users, especially in the case where the users are unknown (*i.e.*, the user profile is not available). Thus, being able to identify the referrer URL when it is not available, is an advantage for the content provider. In this section we want to understand if it is feasible to detect the referrer URL of a user while he browses and how many browsing steps are required to be able to do so accurately. Moreover, if we find that the user sessions are easily distinguishable, it means that the subgraphs are different enough to be considered, in our experiment, as *local BrowseGraphs* of different service providers.

Therefore, we consider the following scenario: a content provider is observing a user surfing the pages of its web service, but it is unaware of the user's referrer URL. In terms of our experimental dataset, this scenario maps into the problem of observing a browsing trace left by a random surfer on one of the referrer-based subgraphs, and having to identify which graph it is. Intuitively, the larger the number of page visits (or *steps*) the surfer will make, the more distinc-

	Full	Facebook	Google	Bing	Yahoo	Reddit	Homepage	Twitter
Full	1.0000	0.1791	0.3931	0.3278	0.3548	0.0656	0.2797	0.0764
Facebook	0.1791	1.0000	0.3146	0.4111	0.3430	0.2616	0.4070	0.3026
Google	0.3931	0.3146	1.0000	0.5815	0.5860	0.1088	0.4217	0.1297
Bing	0.3278	0.4111	0.5815	1.0000	0.6624	0.1469	0.5238	0.1688
Yahoo	0.3548	0.3430	0.5860	0.6624	1.0000	0.1245	0.4632	0.1386
Reddit	0.0656	0.2616	0.1088	0.1469	0.1245	1.0000	0.1534	0.2309
Homepage	0.2797	0.4070	0.4217	0.5238	0.4632	0.1534	1.0000	0.1523
Twitter	0.0764	0.3026	0.1297	0.1688	0.1386	0.2309	0.1523	1.0000

Table 2: Kendall's τ correlations between PageRank values ($\alpha = 0.85$) between the common nodes of the subgraphs.

Algorithm 1: `RandomSurfer`(k, α, steps, G)

```
logPr ← initialize vector with size G_k.length();
n ← total number of nodes;
x_j ← choose (random) starting node ∈ G_k;
/* For each step, compute a random walk in G_k, and
compare the probability to be in all the other G */
for s ← 1 to steps do
    /* Pick the next node of G_k with random walk */
    x_k = next_node( G_k, x_j );
    for i ← 0 to G.length() do
        ⟨k_out⟩ ← get_outdegree(n_p);
        if ⟨k_out⟩ == 0 then
            logPr[ i ] ← logPr[ i ] + log(1/n);
        else
            p_i(x) = (1 − α)/n;
            Pd_{x_j} ← get_prob_distribution(G_i, x_j);
            S_{x_j} ← get_successors(G_i, x_j);
            if x_k ∈ S_{x_j} then
                p_i(x) ← p_i(x) + α * Pd_{x_j}(x_k);
            logPr[ i ] ← logPr[ i ] + log(p_i(x));
return logPr
```

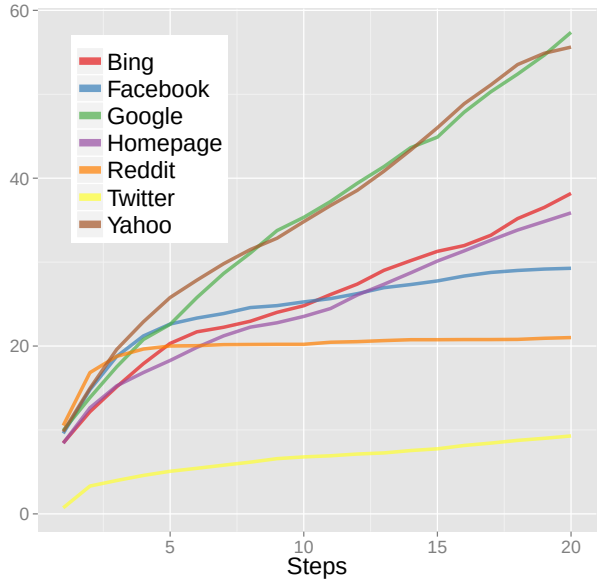


Figure 1: Random Surfer Experiment. On the y-axis: log-ratio of the probabilities (as explained in the text). X-axis: number of browsing steps performed by the surfer.

tive its trace will be, and the easier the identification of the graph. Algorithm 1 shows the pseudocode that describes the process to compute the random surfer experiment.

Formally, observing the sequence of the surfer's visited nodes $\mathbf{x} = (x_1, x_2, \ldots, x_s)$ and computing the probability $p_i(\mathbf{x})$ that the surfer has gone through them given that it is surfing G_i, we need to deduce what is G_i (*e.g.*, by maximum log-likelihood). With this goal in mind, we sort the indices of the subgraphs $i_1, i_2, \ldots$ so that $p_{i_1}(\mathbf{x}) \geq p_{i_2}(\mathbf{x}) \geq \ldots$ and stop as soon as the gap between $\log p_{i_1}(\mathbf{x})$ and $\log p_{i_2}(\mathbf{x})$ is large enough (*e.g.*, $\log p_{i_1}(\mathbf{x}) - \log p_{i_2}(\mathbf{x}) \geq \log 2$), with a maximum of 20 steps that we consider as a representation of a long user session.

In this set of experiments, we considered the seven URL-referral subgraphs $G_1, \ldots, G_7$, one at a time. For each subgraph G_i, we simulated a random surfer moving around in G_i (*i.e.*, calling the function `RandomSurfer`(i, α, steps, G)), computing at each step (*i.e.*, page visited) the probability of the surfer to navigate in each subgraph $G_1, \ldots G_7$: we expect that the probability corresponding to G_i will increase at each step, and will eventually dominate all the others.

To estimate the number of steps required to identify correctly the graph that the surfer is browsing, we measure the difference between log-probabilities for the correct graph G_i and for the graph with the largest log-probability among the other ones. As with PageRank we introduced a certain damping factor ($\alpha = 0.85$); this is necessary to avoid being stuck in terminal components of the graph. Recall that α is the balancing parameter that determines the probability of following in the random walk, instead of teleporting. The results are shown in Figure 1, averaged over 100 executions. The values on the y-axis represent the difference between the log-probabilities (*i.e.*, the logarithm of their ratio): in general, we can observe that the very first steps are enough to understand correctly (and with a huge margin) in which graph the surfer is moving. The inset of Figure 1 displays the first 20 steps and the relative probability to identify the correct graph. Almost all the referrer domains are recognizable at the first step. This translates into a strong advantage for the service provider as it can identify from where the users are coming from, even if they use clients or services that masquerade it. With this information the service provider can personalize the content of the web pages for any users with respect to the referrer.

Interestingly, the plot reveals that some surfers are easier to single out than others; we read this as yet another confirmation that the subgraphs have a distinguished structural difference, or (if you prefer) that users have a markedly different behavior depending on where they come from. This experiment does not only showed that is possible to detect from which referrer domain the surfer is coming from, but that the graphs are quite different and that they can be used for our study.

5. PAGERANK ON THE BROWSEGRAPH

Next, we study the convergence of the PageRank ranking between the *local BrowseGraphs* (*ReferrerGraphs*) and the full *BrowseGraph*. We want to understand how different are the ranking computed using less or more knowledge about the full graph. We present an experiment, called "Growing Balls", that compute the distance between the rankings expanding at each step the known nodes (and edges) with the neighbors of the subgraphs.

5.1 "Growing Balls" Experiment

We first focus on the study of the *Local Ranking Problem* on browsing graphs. An important question related to this problem is how much the PageRank node values vary, when new nodes and edges are added to the local graph. A natural way to determine this is to expand incrementally the graph by adding new nodes and edges in a Breadth-First Search (BFS) fashion, and comparing the PageRank computed on the expanded graph with the one on the global graph.

More formally, given a graph H which is a subgraph of the full graph G, we simulate a growth sequence H_0, $H_1 \dots H_n$ in the following way:

- $H_0 \longleftarrow H$;
- $V_{H_{k+1}} \longleftarrow \{\Gamma_{out}(V_{H_k}) \cup V_{H_k}\}$, with V_x being the set of vertices of a graph, and Γ being the vertex neighborhood function;
- $E_{H_{k+1}} \longleftarrow \{(v_1, v_2) | v_1 \in V_{H_{k+1}} \wedge v_2 \in V_{H_{k+1}}\}$, with E_x being the set of edges of a graph.

Using the standard graph terminology, we refer to the various steps of this expansion as *"balls"*, where the ball H_0 is the initial subgraph and subsequent balls are obtained by adding all the outgoing arcs that depart from the nodes in the current ball and end in nodes that are not in the ball. Observe that, depending on how it is built, H_0 may not be an induced subgraph of G, but $H_1, \cdots, H_n$ are always induced subgraphs, by definition of the expansion algorithm.

Using the Kendall's τ function, we measure the difference between the local PageRank computed for each ball H_i, and the global PageRank computed on G. The main objective is to understand how much the ranking gets close the global one at each consecutive step, and whether the ranking values are able to converge even if we just consider a piece of the information contained in the whole graph.

To check the dependency of results from the initial graph selected, we consider three different sets of initial subgraphs, that we will study separately. We describe them next.

- **Referrer-based (RB)**. The seven browsing subgraphs built by referrer URL: Facebook, Twitter, Reddit, Homepage, Yahoo, Google and Bing;
- **Same size referrer-based (SRB)**. To measure how much the different sizes of the graphs impact on the observed behavior, we fix a number of nodes and extract a portion of each subgraph in order to obtain exactly the same size for all networks. The selection is performed with several attempts of BFS expansion, starting from a random node in each graph, until the resulting graphs have very similar size ($\pm 9.4\%$): other ways of selecting subgraphs would end up with disconnected samples, which of course would void the purpose of this experiment. With this procedure instead, we are able to compare the graphs on equal grounds and at the same time control for the effect of size (about $3K$ nodes and $20K$ edges).
- **Random (R)**. To check whether the observed behavior has to do with the user behavior underlying the graph under examination (*e.g.*, the particular structure of the graph determined by the sessions of users coming from Twitter), we take a set of seven *random* graphs each of them reflecting the size of each of the referrer-based subgraphs. Thus, we can explore the behavior of browsing graphs, that preserve the size of the graphs originated by specific types of users, but that are "artificial" in the sense that destroy any connection with the behavior connected to a particular user class. To make sure that the size is the same, we start from a BFS exploration and we prune the last level to match exactly the size we need.

The results related to the **RB** case are shown in Figure 2 (left). The convergence happens relatively quickly, as the value τ approaches 1 in the first 3 iterations. The curves related to different subgraphs are shifted with respect to each other, apparently mainly due to their different size, the biggest networks starting from higher τ values and converging faster than the smaller ones. To determine the dependency on the graph size, we repeat the same experiment for the **SRB** case. The results for this case are shown in Figure 2 (center). Even if the curves resulted to be more flattened (confirming that the initial size has indeed a role in the convergence), we still observe noticeable differences between the curves for the first two expansion levels. This means that different subgraphs are substantially different from one another in terms of their structure: even after forcing them to have the same size, the convergence rates observed on the different graphs varies. At the first iteration, for instance, all the subgraphs in **SRB** have Kendall's τ between 0.3 and 0.5, whereas the ones in **RB** are between 0.4 and 0.6. Moreover in **SRB** the biggest networks starting from higher τ values are not converging faster. This intuition is confirmed by repeating the experiment on graphs selected with the **R** startegy. Results, displayed in Figure 2 (right), show that convergence in this case is much slower and the difference between the curves is less prominent.

Summarizing, with the previous experiment, we show that the Growing Balls on random subgraphs behave differently, especially when considering the number of iterations required in order to converge.

5.2 Growing Balls with Selection of Nodes

Besides the selection of the initial graph, the rank convergence depends also on the way the growing balls are built at each iteration. How does the expansion influence conver-

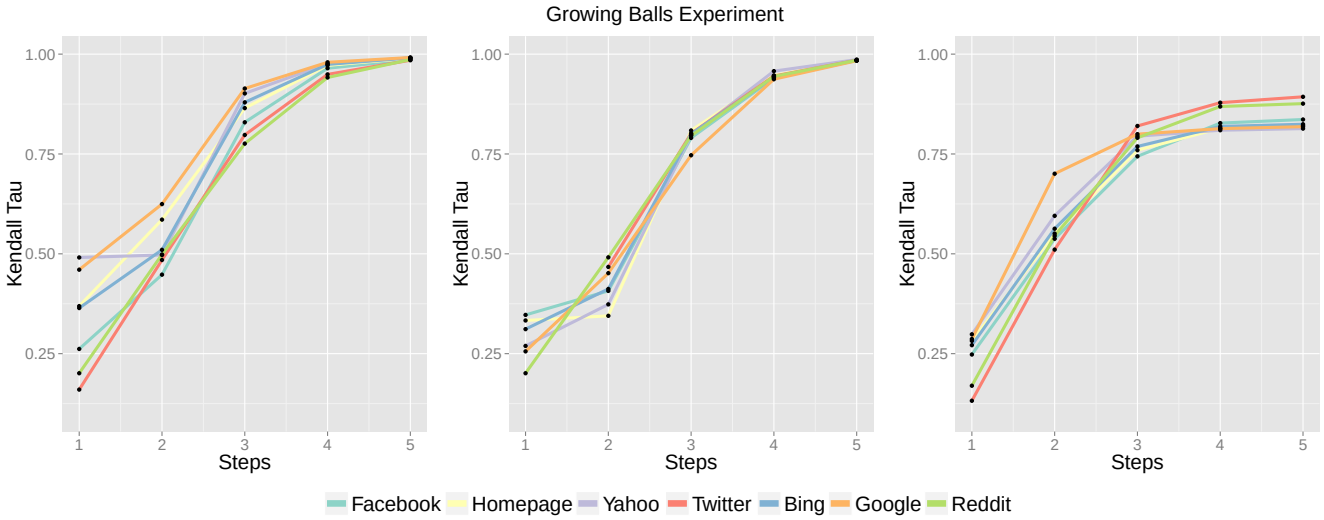


Figure 2: Growing Balls experiment on: (left) original subgraphs built based on the referrer URL, (center) seven subsubgraphs with very similar size, (right) eight subgraphs random selected from the full graph, where each of them has the same size of one of the original.

gence if only few more representative nodes are selected? To what extent a higher *volume* of selected nodes helps a quicker convergence or adds more *noise*? At a first glance, one may argue that using all the nodes is equivalent to injecting all the available information, so the convergence to the values of PageRank computed on the full graph G should be faster. On the other hand, instead, one may observe that we are introducing a huge number of nodes in each iteration (as the growth is at each step larger), adding also the ones that are less important and this can induce an incorrect PageRank for some time, until all the graph becomes known. In order to shed light on this aspect, we introduce a variant in the growing-balls expansion algorithm, and we select only the nodes with highest PageRank.

More formally, considering H_k as the subgraph at iteration k and V_{H_k} its set of nodes, we select all the external nodes in $Y = \{V_G \backslash V_{H_k}\}$, that are connected through outgoing arcs from the nodes in V_{H_k}. We then compute the PageRank values on the subgraph H_k extended with the nodes Y, and obtain a ranked list of nodes. Among all the nodes in Y we select the top $n\%$ with largest PageRank value, and only those ones will be added to H_k in order to build H_{k+1} and advance to the next iteration.

We conducted experiments with this partial expansion at different percentages: 5%, 10%, 30%, 50%, and 100%, and then we computed the average Kendall's τ value for each one of the percentages. The results are shown in Figure 3. Remarkably, the figure highlights how expanding the graph by adding fewer nodes, although the most representative ones, leads to PageRank values that are closer to the *global* ones in the first iterations. Since we are expanding the local graph with a small (highly-central) number of nodes, we could argue that they initially help to boost the local PageRank scores. However, given that we keep on expanding using a few nodes at each iteration, the nodes that have not been added before exclude a large number of nodes among which there might also be highly central ones. This might explain why in the first iteration(s) the convergence rate is high, but on the limit the final convergence values result in a low Kendall's τ. Contrarily, in the long run, expansions that include the highest number of nodes present convergence values closer to 1. This is somehow expected, given that at each iteration any subgraph H closer in size to the full graph G will include almost every node and arc.

Nonetheless, the main significant outcome of this experiment is that it is possible to obtain a yet satisfactory PageRank convergence, with few but very representative nodes. For situations in which including additional pieces of information, in terms of node/arc insertions, implies a non-negligible cost, requesting just a little amount of well-selected information allows to obtain good approximations while minimizing the costs.

6. PAGERANK PREDICTION

In the previous section we have shown how the approximation to the global PageRank varies with the expansion of the initial subgraph. The ranking of the nodes converges quite fast on all the subgraphs: they differ in terms of their content, although they are similar in terms of structure in that all of them are built based on users' navigational patterns. Building upon the findings about how local and global PageRank computed on the *BrowseGraphs* relate to each other, we designed an experiment to assess how well a learned model could perform in predicting this relationship.

We address the problem of predicting the Kendall's τ between the local and the global PageRank, only considering information available on the local graph such as topological features. This is an extremely common situation given that, in general, the information pertaining the local graph is the only one that is readily available, and usually of a limited size. Computing this distance accurately has a high value for service providers, since it translates directly into an estimation of the reliability of the PageRank scores computed on their local subgraphs. As a direct consequence one can apply, with different levels of confidence, methods for optimizing web sites [31], studying user en-

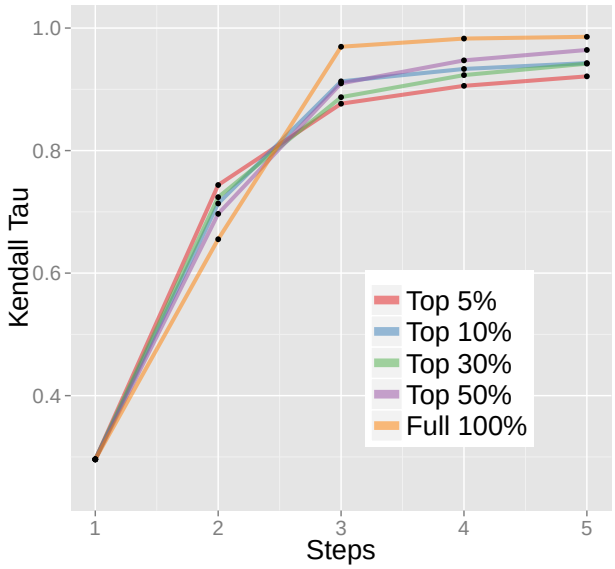


Figure 3: Growing Balls using only the nodes with highest PageRank. The plot shows the average values of the Kendall-τ at each step computed for all the subgraph.

gagement [18], characterizing user's session [12] or content recommendation [27].

6.1 Prediction of Kendall τ Distance

We have seen that the deviation of the local PageRank with respect to the global one can be relevant, depending on factors such as the size of the local graph and the different behavior of the users who browse it (see §5.1 and particularly Figure 2). Recall that we compute the distance comparing the rankings with Kendall's τ, since we are interested in obtaining a ranking as close as possible to the one computed on the entire graph. Although we have previously shown how to expand the view on the local graphs with nodes residing at the border, this practice might not always be possible in a real-world scenario, since service providers often can have access only to the browsing data *within* their domain.

Previous work on local ranking on graphs raised several questions related to this scenario, highlighting practical applications of the local rank estimation non only for web pages but also in social networks [9]. Critically, so far it is not clear whether there are some topological properties of the local graph that make the local ranking problem easier or harder, and if these properties can be exploited by local algorithms to improve the quality of the local ranking. We explore this research direction by studying a fundamental aspect that is at the base of the open questions in this area, namely the possibility of estimating the deviation of the local PageRank from the global one, using the structural information of the local network. The intuition is that, some structural properties of the graph could be good proxies for the τ value difference, computed between local and global ranks. Being able to estimate the Kendall's τ distance between the subgraph available to the service provider and the global graph, implies the ability to estimate the reliability of the current ranking using only information of the local subgraph.

To verify this hypothesis we resort to regression analysis. Starting from the seven subgraphs in the dataset, we build a training set using the jackknife approach, by removing nodes in bulks (1%, 5%, 10%, 20%) and computing the τ value between the full subgraph and their reduced versions. Then, for each instance in the training set, we compute 62 structural graph metrics [30, 4] belonging to the following categories:

- **Size and connectivity (S)**. Statistics on the size and basic wiring properties, such as number of nodes and edges, graph density, reciprocity, number of connected components, relative size of the biggest component.
- **Assortativity (A)**. The tendency of node with a certain degree, to be linked with nodes with similar degree. We computed different combinations that take into account the in/out/full degree of the target node vs. the in/out/full degree of the nodes that are connected with it.
- **Degree (D)**. Statistics (average, median, standard deviation, *etc.*) on the degree distribution of nodes.
- **Weighted degree (W)**. Same as **degree**, but considering the weight on edges, that usually referred as node strength. As the edges are the transitions made by the users during the navigation, the weight stand for the number of times the users have navigated the transition.
- **Local Pagerank (P)**. Statistics on the distribution of the PageRank values computed on the local graph.
- **Closeness centralization (C)**. Statistics on the distances (number of hops), that separate a node to the others in the graph, in the spirit of the closeness centralization [30].

We employed different regression algorithms, although we report the performance using random forests [7], which performed better in this scenario than other approaches like support vector regression [25]. We computed the mean square error (MSE) across all examples in all sampled subgraphs. The random forest regression has been computed over a five-fold cross validation averaged over 10 iterations. The mean square residuals that we obtained is very low, around $2.4 \cdot 10^{-6}$. Results, computed for the full set of features and for each category separately, are given in Table 3. The most predictive feature category is the *weighted degree*, which yields a performance that is better (or comparable) than the model using all the features, whereas the *assortativity* features seem to be the ones that have the less predictive power on their own. This might be due to the fact the model with 62 features is too complex for the amount of training data available. On the other hand, the *weighted degree* that is the best performing class of features, contains the statistics of the degree distribution on the weighted edges. In Figure 4 the features included in *weighted degree* are ranked by their discriminative power in predicting the Kendall τ distance using the permutation test proposed by Strobl *et al.* [26]. These features, which are based on the distribution of the out- and in-degree of the nodes, are straightforward to compute from the local graph—a very affordable task for service providers.

Feature Class	No. Features	MSE
weighted degree	15	$2.2 \cdot 10^{-6}$
degree	15	$2.9 \cdot 10^{-6}$
local PageRank	10	$3.3 \cdot 10^{-6}$
size and connectivity	9	$3.4 \cdot 10^{-6}$
closeness	5	$4.1 \cdot 10^{-6}$
assortativity	8	$9.3 \cdot 10^{-6}$
ALL features	62	$2.4 \cdot 10^{-6}$

Table 3: MSE of cross validation. Average differences are statistically significant with respect to *weighted degree* and *ALL features* (t-test, p<0.01).

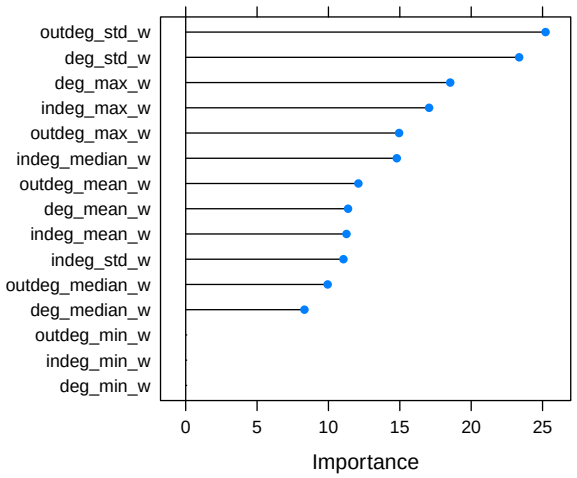


Figure 4: The 15 features of *weighted degree*, the most predictive class, sorted by importance. Note that some of them do not have any contribution to the Kendall-τ prediction, therefore just few features are necessary in order to estimate the distance.

We then use the learned model to predict the τ values of the seven subgraphs. When we applied the predictive models learned in the subsamples to regressing the full graphs, the MSE, is less than 0.026 on average, which, even if relatively low, it is higher than the cross-validated performance in the sub-samples. However, the model was able to rank the seven different subgraphs by their Kendall's τ almost perfectly. When using all the features the Spearman's correlation coefficient between the true order and the predicted one is 0.85 (high correlation), and when we used the most predictive features (weighted degree) the correlation was as high as 0.80 (moderate high correlation). Overall, the final rankings are just one swap away (Kendall's τ is over 0.70 in this case). This kind of information can be very helpful when comparing different local sub-domains to determine which one has pages that better estimate the global PageRank.

7. CONCLUSION

In this paper we tackled the *Local Ranking Problem*, *i.e.*, how to estimate the PageRank values of nodes when a portion of the graph is not available, which arises commonly in real use cases of PageRank. We investigated this problem for a novel environment, namely estimating PageRank on a large user-generated browsing graph from a large news provider. The peculiar characteristic of this graph is that it is built from user's navigation patterns, where nodes represent web pages and edges are the transitions made by the users themselves. Moreover, the information about the domain of origin of the users (namely the referrer URL of their sessions), is also available.

We built a set of *ReferrerGraphs* including the browsing subgraphs based on different referrer URLs, and then we studied their difference in terms of user navigation patterns. We found that all of the browsing patterns initiated from different domains exhibit remarkable differences in terms of which pages users visited next. The referrer URL (or domain) has been found to be extremely useful for characterizing the user behavior [12] or for recommendation of content [27]. With this observation in mind and motivated by the cases where the domain from where the user is coming is not available, such as Facebook and Twitter clients or URL shortening services, we performed a series of experiments with the aim of predicting from which referrer URL the user joined the network, *i.e.*, if a model can predict reliably where the user is entering our network. In general, just a few steps (*i.e.*, visited pages) suffice to recognize the referrer URL correctly because the surfing behavior is very distinctive of the domain the user is coming from.

Then, using the *ReferrerGraphs*, we performed several experiments using a very large network of sites (with almost two billions of user transitions) to assess to what extent the browsing patterns information can be generalized, if one is only provided with information from smaller subgraphs. First, we computed the PageRank of the subgraphs and on their step-by-step BFS expansion, measuring the distance in terms of Kendall's τ with the PageRank computed on the full graph. To control for the subgraph size and type, and to study the impact of the expansion strategy on the PageRank convergence, we used two flavors of BFS and three different sets of initial subgraphs. We found that expanding the local graph with few nodes of largest value of PageRank leads to a faster (74% at the first expansion step), although less accurate convergence in the long run. On the other hand, adding more nodes lead to a slower converge rate in the first steps (65%). Therefore, in all the cases where a strong convergence with the values of the global PageRank is not required, selecting few specific nodes is enough to significantly improve the PageRank values of the local nodes, without having to request and process a larger amount of data.

Finally, we considered the case of a service provider that wants to estimate the reliability of the scores of PageRank computed on its local *BrowseGraph*, with respect to the ones computed on the global graph. Therefore, we performed another experiment trying to predict the value of the Kendall's τ between the local and the global PageRank, only considering information available on the local graph. We explored six different sets of topological and structural features of the browse graph, namely size and connectivity, assortativity, degree, weighted degree, local PageRank and closeness. Then we computed those features on a training set that we obtained by applying a jackknife sampling of our subgraphs, and we ran a regression on the Kendall's τ of the PageRank of the full subgraph and the various samplings.

We found that a random forest ensemble built on *weighted degree*, outperforms all the other in terms of mean square error. When applying the regression to the task of predicting the τ value of the global graph with the eight subgraphs at hand, we were able to reproduce quite well the ranking of their estimated τ values with their actual ranking, up to a Spearman's coefficient of 0.8.

Future Work. We envision different routes worth being taken into consideration for future work. One line of research we plan to investigate deals with the problem of user browsing prediction. In other words, what extent it may be possible to identify what are the most common patterns of topical behavior in the network and also, to build per-user browsing models to predict what would be the page to be visited next. Further, motivated by real use case scenarios, we considered subgraphs determined by the referrer URL of user sessions; we believe that interesting analytical results could be found, when considering other types of subgraphs, such as networks induced by nodes that belong to the same topical area.

8. ACKNOWLEDGMENTS

This work was partially funded by Grant TIN2009-14560-C03-01 of the Ministry of Science and Innovation of Spain, by the EU-FET grant NADINE (GA 288956) and by a Yahoo Faculty Research Engagement Program.

9. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR*, pages 19–26, New York, NY, USA, 2006. ACM.

[2] R. Andersen, C. Borgs, J. Chayes, J. Hopcraft, V. S. Mirrokni, and S.-H. Teng. Local computation of pagerank contributions. In *WAW*, pages 150–165, San Diego, CA, USA, 2007. Springer-Verlag.

[3] Z. Bar-Yossef and L.-T. Mashiach. Local approximation of pagerank and reverse pagerank. In *CIKM*, pages 279–288, Napa Valley, California, USA, 2008. ACM Press.

[4] A. Barrat, M. Barthlemy, and A. Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press, New York, NY, USA, 2008.

[5] K. Bharat, A. Broder, M. Henzinger, P. Kumar, and S. Venkatasubramanian. The connectivity server: fast access to linkage information on the web. In *WWW*, volume 30, pages 469–477, Brisbane, Australia, 4 1998. Elsevier Science Publishers B. V.

[6] P. Boldi, M. Santini, and S. Vigna. Do your worst to make the best : Paradoxical effects in pagerank incremental computations. In *WAW*, pages 168–180. Springer, 2004.

[7] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, oct 2001.

[8] M. Bressan, E. Peserico, U. Padova, and L. Pretto. The power of local information in pagerank. In *WWW Companion*, pages 179–180, Rio de Janeiro, Brazil, 2013.

[9] M. Bressan and L. Pretto. Local computation of pagerank: the ranking side. In *CIKM*, pages 631–640. ACM, 2011.

[10] Y.-Y. Chen, Q. Gan, and T. Suel. Local methods for estimating pagerank values. In *CIKM*, pages 381–389, New York, NY, USA, 2004. ACM.

[11] L. Chiarandini, P. Grabowicz, M. Trevisiol, and A. Jaimes. Leveraging browsing patterns for topic discovery and photostream recommendation. In *ICWSM*, Cambridge, MA, USA, 2013. AAAI.

[12] L. Chiarandini, M. Trevisiol, and A. Jaimes. Discovering social photo navigation patterns. In *ICME*, pages 31–36. IEEE, 2012.

[13] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through url ordering. In *WWW*, volume 30, pages 161–172, Brisbane, Australia, 4 1998. Elsevier Science Publishers B. V.

[14] J. V. Davis and I. S. Dhillon. Large scale analysis of web revisitation patterns. In *KDD*, volume 08, pages 116–125, Philadelphia, PA, USA, 2006. ACM Press.

[15] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *VLDB*, pages 576–587, Toronto, ON, Canada, 2004.

[16] J. Hu, H.-J. Zeng, H. Li, C. Niu, and Z. Chen. Demographic prediction based on user's browsing behavior. In *WWW*, pages 151–160, New York, NY, USA, 2007. ACM.

[17] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[18] J. Lehmann, M. Lalmas, and R. Baeza-Yates. Measuring inter-site engagement. In *Big Data, 2013 IEEE International Conference on*, pages 228–236. IEEE, 2014.

[19] R. Lempel and S. Moran. Salsa : The stochastic approach for link- structure analysis. *Challenge*, 19(2):131–160, 2001.

[20] J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. In *IUI*, pages 31–40, New York, NY, USA, 2010. ACM.

[21] M. Liu, R. Cai, M. Zhang, and L. Zhang. User browsing behavior-driven web crawling. In *CIKM*, pages 87–92, New York, NY, USA, 2011. ACM.

[22] Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li. Browserank: letting web users vote for page importance. *SIGIR*, 31:451–458, 2008.

[23] Y. Liu, T.-Y. Liu, B. Gao, Z. Ma, and H. Li. A framework to compute page importance based on user behaviors. *Information Retrieval*, 13(1):22–45, 6 2009.

[24] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *World Wide Web Internet And Web Information Systems*, 54(2):1–17, 1998.

[25] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. Technical report, Statistics and Computing, 2003.

[26] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1):307, 2008.

[27] M. Trevisiol, L. M. Aiello, R. Schifanella, and A. Jaimes. Cold-start news recommendation with domain-dependent browse graph. In *RecSys*, Foster City, CA, 2014. ACM.

[28] M. Trevisiol, L. Chiarandini, L. M. Aiello, and A. Jaimes. Image ranking based on user browsing behavior. In *SIGIR*, pages 445–454, New York, NY, USA, 2012. ACM.

[29] M. Tsagkias and R. Blanco. Language intent models for inferring user browsing behavior. In *SIGIR*, pages 335–344, New York, NY, USA, 2012. ACM.

[30] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[31] B. Weischedel and E. K. R. E. Huizingh. Website optimization with web metrics: A case study. In *ICEC*, pages 463–470, New York, NY, USA, 2006. ACM.

[32] R. W. White. Investigating behavioral variability in web search. In *In Proc. WWW*, pages 21–30, 2007.

[33] R. W. White and J. Huang. Assessing the scenic route: measuring the value of search trails in web logs. In *SIGIR*, pages 587–594, New York, USA, 2010. ACM.