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

[*]Web Research Group
Universitat Pompeu Fabra
Barcelona, Spain

[†]Yahoo Labs
London, UK

[§]Dip. di Scienze dell'Informazione
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: *(i)* gauge their ability to correctly assess the importance of their resources only based on their local knowledge, and *(ii)* take 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

SIGIR'15, August 09 - 13, 2015, Santiago, Chile.
© 2015 ACM. ISBN 978-1-4503-3621-5/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2766462.2767704.

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 where 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 net-works. 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 observe 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 *(i)* when estimating the centrality score of nodes in a *BrowseGraph*, and *(ii)* 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 patterns 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 Rings”. 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 (“rings”). 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 snowingreen 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, which 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]), which 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*, which 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. The dataset used in this work has been extracted from the data built in [27], that was used with the purpose to study the news consumption with respect to the Referrer URL. 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:

$$\langle BCookie, Timestamp, ReferrerURL, URL, UserAgent\rangle$$

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. *URL* 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. We removed traffic derived from Web crawlers by preserving only the entries whose User-Agent field contains a well-known browser identifier. 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

[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, density and Giant Connected Component 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.

are unknown (full *BrowseGraph*). Since it is not possible to use the full Web browsing log, we perform a simulation using different subgraphs extracted from the same *BrowseGraph* that represent the local graphs of different providers. One possible approach would be to simulate each service provider as a different Yahoo News subdomain (*e.g.*, news, sports, finance). However, news articles are often shared on different Yahoo subdomains and, as a consequence, the users jump among different subdomains in each single session. To avoid such an overlap on the subgraphs, we define a different simulation approach. 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 *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, which 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. 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 audience, 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`), which 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 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.

	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 $\in G_k$;
/* For each step, compute a random walk in $G_k$, and
compare the probability to be in all the other $G$ */
for $s \leftarrow 1$ to steps do
    /* Pick the next node of $G_k$ with random walk */
    $x_k$ = next_node( $G_k$, $x_j$ );
    for $i \leftarrow 0$ to $G.length()$ do
        $\langle k_{out} \rangle \leftarrow$ get_outdegree($n_p$);
        if $\langle k_{out} \rangle == 0$ then
            logPr[ i ] ← logPr[ i ] + log(1/$n$);
        else
            $p_i(x) = (1 - \alpha)/n$;
            $Pd_{x_j} \leftarrow$ get_prob_distribution($G_i$, $x_j$);
            $S_{x_j} \leftarrow$ get_successors($G_i$, $x_j$);
            if $x_k \in S_{x_j}$ then
                $p_i(x) \leftarrow p_i(x) + \alpha * Pd_{x_j}(x_k)$;
            logPr[ i ] ← logPr[ i ] + log($p_i(x)$);
return logPr
```

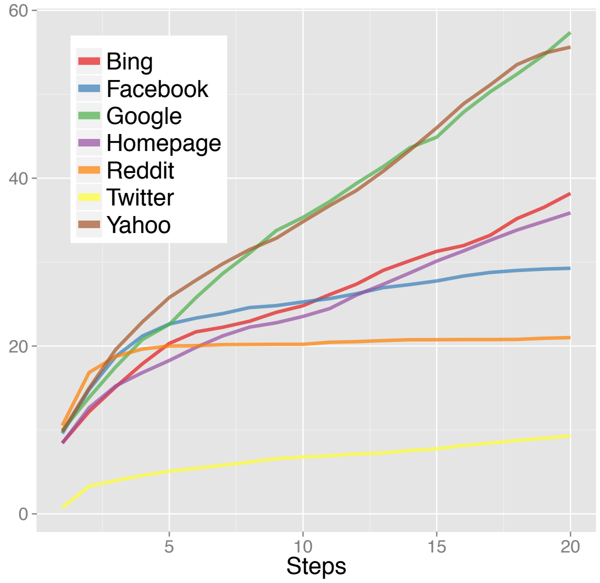

Figure 1: Random Surfer Experiment. Y-axis: log-ratio of the probabilities between the correct graph and the graph with the largest log-probability (as explained in the text). X-axis: number of browsing steps performed by the surfer.

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 distinctive 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 Rings", which compute the distance between the rankings expanding at each step the known nodes (and edges) with the neighbors of the subgraphs.

5.1 "Growing Rings" 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.

We refer to the various steps of this expansion as *"rings"*, where the ring H_0 is the initial subgraph and subsequent rings are obtained by adding all the outgoing arcs that depart from the nodes in the current ring and end in nodes that are not in the ring. 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 ring 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, which 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 (±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, which 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** strategy. 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 Rings on random subgraphs behave differen-

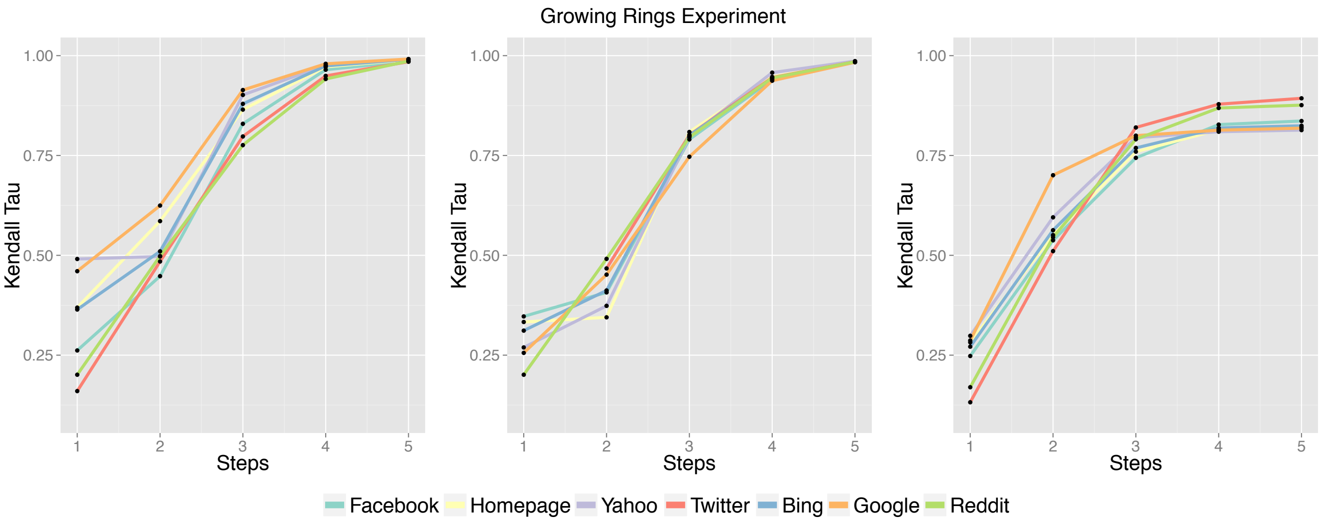

Figure 2: Growing Rings experiment on: (left) original subgraphs built based on the referrer URL, (center) seven subsubgraphs with very similar size, (right) seven subgraphs random selected from the full graph, where each of them has the same size of one of the original.

tly, especially when considering the number of iterations required in order to converge.

5.2 Growing Rings with Selection of Nodes

Besides the selection of the initial graph, the rank convergence depends also on the way the growing rings are built at each iteration. How does the expansion influence convergence 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-rings 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 $Y = \{V_G \backslash V_{H_k}\}$, which 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.

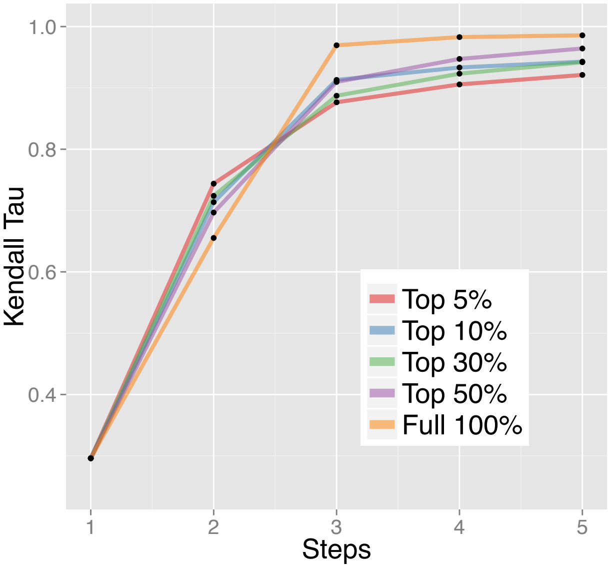

Figure 3: Growing Rings using only the nodes with highest PageRank. The plot shows the average values of the Kendall-τ at each step computed for all the subgraph.

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 engagement [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, which 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), which 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. The *assortativity* features instead, 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

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).

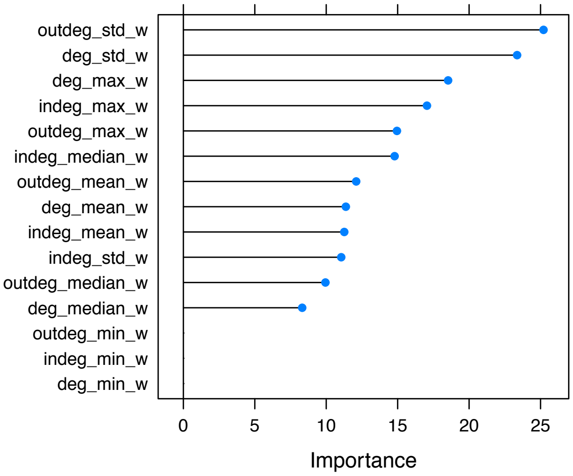

Figure 4: The 15 features of *weighted degree*, the most predictive class, sorted by importance. Note that some of them do not have any contribution to the Kendall-τ prediction, therefore just few features are necessary in order to estimate the distance.

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.

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 Yahoo News. 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 seven 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 Publ 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 Publ 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*, 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 Univ. 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.