What the HAK?
Estimating Ranking Deviations in Incomplete Graphs*

Helge **H**olzmann, Avishek **A**nand, Megha **K**hosla
L3S Research Center
Appelstr. 9a
30167 Hannover, Germany
{holzmann,anand,khosla}@L3S.de

ABSTRACT

Most real-world graphs collected from the Web like Web graphs and social network graphs are *incomplete*. This leads to inaccurate estimates of graph properties based on link analysis such as PageRank. In this paper we focus on studying such deviations in ordering/ranking imposed by PageRank over incomplete graphs. We first show that deviations in rankings induced by PageRank are indeed possible. We measure how much a ranking, induced by PageRank, on an input graph could deviate from the original unseen graph. More importantly, we are interested in conceiving a measure that approximates the rank correlation among them without any knowledge of the original graph. To this extent we formulate the HAK measure that is based on computing the impact redistribution of PageRank according to the local graph structure. Finally, we perform extensive experiments on both real-world Web and social network graphs with more than 100M vertices and 10B edges as well as synthetic graphs to showcase the utility of HAK.

KEYWORDS

Graph Analysis; Incomplete Subgraphs; PageRank

1 INTRODUCTION

Most real-world graphs collected from the Web like Web graphs and social network graphs are *incomplete* or in other words their graph topology is not known in entirety [13, 28]. Especially if not crawled for a particular purpose or subset, but extracted from existing crawls, such as Web archives. The goal of Web archive crawlers is to capture as much as possible starting from some seed set within some national domain or even broader, given the available but limited resources [9]. Incompleteness is an inherent trade-off already in the design decision of such an archive [14]. Complicating matters further, Web archives are often not constructed in one piece but by merging partial crawls [19]. Additional reasons for the incompleteness in Web archives include the restrictive *politeness* policies (i.e., *robots.txt*) or random timeouts of Web servers. Several studies on this topic have shown that incompleteness is indeed a common issue [1–3, 22], inevitably affecting the graphs extracted from such crawls as well.

As a result, important graph properties and measures used for link analysis and structural characterization like *authority of vertices* might be inherently flawed or exhibit deviations from their original values. This is commonly observed where users are typically agnostic to the incompleteness of the obtained graph, hoping that the input graph is a reasonable representative sample of the underlying (unseen) original graph. Some of the well-known measures for computing authority of vertices or relative ordering of vertex authorities based on random walks are PageRank [32] and its variants [15, 18, 25].

As an example, consider PageRank computed over the `.gov` Web graph that we will analyze in detail later in this work. Here, the `women.nasa.gov` (*Women@NASA*) page has a high PageRank value and is subsequently found within the top 300 pages. However, on a closer examination we observe that most of its PageRank is contributed by an in-link from the highly popular NASA homepage (`nasa.gov`). If for some reason this particular in-link is not crawled, e.g., due to a temporary downtime or the decision by *NASA* to exclude their homepage from being crawled, this would cause a large decrease in its PageRank and hence a severe rank deviation in the obtained crawl.

One might argue that this is an unlikely case since *important* pages enjoy a high priority and are therefore commonly crawled, but this might not always be the case in reality. To support our claim we performed the following experiment. We ranked pages in a graph constructed from a `.de` Web archive in 2012[1] based on (1) *inlinks* and (2) PageRanks. The above mentioned graph considered only links that emerged in 2012 [20]. We then checked if the top ranked pages in this incomplete graph were indeed archived in that year. Our experiments show that from among the top 1000 pages, ranked according to inlinks, roughly 30% are contained in the archive. According to PageRank rankings, less than 20% of the top 1000 pages are contained in the archive. With this small experiment we show that high priority vertices can indeed be missed in real world crawls, which can further cause a rank deviation in the obtained incomplete graph.

We, therefore, study the deviation in orderings/rankings imposed by PageRank over incomplete graphs. Vertices in our input crawls are either *completely crawled* (all neighbors are known) or are *uncrawled* (none of their neighbors are known), which we refer to as *ghost vertices*. Based on this, the research questions we ask are the following:

- **RQ I :** *Do incomplete real-world graphs show a deviation in their PageRank orderings when compared to full network topology?*
- **RQ II :** *How can we reliably measure the extent of such ranking deviations for incomplete graphs?*

*This work is partly funded by the European Research Council under ALEXANDRIA (ERC 339233)

14th International Workshop on Mining and Learning with Graphs (MLG'18), co-located with KDD'18, August 20, 2018, London, United Kingdom

[1] the archive has been generously provided to us by the Internet Archive

Towards these, we perform extensive experiments on both real-world Web and social network graphs with more than 100 million vertices and 10 billion edges. We first establish empirically that real-world networks indeed show a deviation in their PageRank orderings when not crawled completely compared to the complete graph (**RQ I**). We observe ranking correlations (measured by *Kendall's Tau*) dropping down to 0.55 on Web graphs when only 50% of it is crawled. Second, users and applications that use rankings induced by PageRank as a feature for downstream ranking and learning tasks would naturally be interested in estimating such a deviation from the (incomplete) input graph at hand as a measure of confidence. Therefore, as an answer to **RQ II**, we propose a measure called HAK (an acronym of the authors' names) that estimates the ranking deviation of an incomplete input graph when compared to the original graph.

2 RELATED WORK

Ng et al. [31] analyzed the conditions under which eigenvector methods like PageRank and HITS can provide reliable rankings under perturbations to the linkage patterns for a given collection. In particular for PageRank they showed that, if perturbed or modified web pages, i.e., links from the page are removed or are not followed, did not have a high PageRank score in the original graph, then the new PageRank score will not be far from the original. However, this would change when top pages in the crawl are perturbed. In particular, when some high ranked page is missed as we discussed in the previous section, the resulting PageRank rankings will be highly unstable. Moreover their paper discusses the ranking deviations only for the top 10 items in either of the considered rankings though. We on the other hand, provide a quantitative evaluation using Kendall's Tau for a much larger fraction of the graph, which is crucial for the use of PageRank in Information Retrieval scenarios where a selected set of relevant documents are ranked. Further, we provide a measure to estimate ranking deviations of vertices in the given graph with respect to their orderings in the original unmodified graph.

Boldi et al. [6] also show the paradoxical effects of PageRank computation on Web graphs. In contrast to our work though, they discuss a measure of effectiveness for crawl strategies based on whether the graph obtained after a partial visit is in some sense representative of the underlying Web graph for the PageRank computation. Similar to our setting, they study how rapidly the computation of PageRank over the visited subgraph yields relative ranks, measured by Kendall's Tau, that agree with the ones the vertices have in the complete graph.

In [34], unlike other approaches that sample vertices, the authors operate on a given subset of vertices and consider the general problem of maintaining multi-scale graph structures by preserving a distance metric based on PageRank among all pairs of sampled vertices.

The other area of related work comprises of graph sampling approaches which can be broadly classified into two categories: *traversal based* methods [26, 29, 35] and random walk based methods [13, 21, 28]. Graph-traversal based methods employ breadth-first search (BFS) or the depth-first search (DFS) algorithm to sample vertices and are typically shown to exhibit bias towards high-degree vertices [35]. Maiya and Berger-Wolf [29] compare various traversal based algorithms and define representativeness of a sample while proposing how to guide the sampling process towards inclusion of desired properties. On the other hand, the random walk based methods are popular for graph sampling because they can produce unbiased samples or generate samples with a known bias [13, 21, 28, 37]. One of the popular sampling algorithms used for Web graphs is the *Forest Fire* algorithm by Leskovec and Faloutsos [26], a generative graph model, in which new edges are added via an iterative "forest fire" burning process where it is shown to produce graphs exhibiting a network community profile plot similar to many real-world graphs. We use this approach in generating synthetic real-world graphs. Other related works dealt with estimating graph properties such as degree distribution estimation [13], clustering coefficient estimation [17], size estimation [23], and average degree estimation [10]. However, most of these works assume a known graph topology. Our work focuses on the unknown graph topology, an arguably more general and useful scenario in Web graphs and social networks gathered by crawlers.

3 PRELIMINARIES AND PROBLEM

PageRank. As originally conceived, PageRank ranks vertices of a directed graph $\mathcal{G} = (V, E)$ where V and E are the vertices and edges respectively, based on the topological structure of the graph using random walks [32]. The problem we are addressing in this paper is attributed to this random walk model behind PageRank, representing the *authority* or *importance* of a vertex. For some fixed probability α, a surfer at vertex $v \in V$ jumps to a random vertex with probability α and goes to a linked vertex with probability $1-\alpha$. The *authority* of a vertex v is the expected sum of the *importance* of all the vertices u that link to v. Consequently, a vertex receives a high PageRank value and is ranked at the top by ordering the webpages by *importance* when it is either connected by many incoming edges or reachable from another *important* page.

We first define the notions of *target graph*, *crawl* and *ghost vertices* in the context of incompleteness in graphs due to their collection process:

Definition 3.1 (Target graph). The subset of vertices (with the induced edges) of a larger graph (e.g., the Web) that is theoretically reachable by a crawler given its seeds, e.g., a domain, a top-level domain, or all webpages that belong to a certain topic in case of focused crawlers. This graph would be available if every link was followed and every page captured by the crawler, illustrated by the *target* in Figure 1.

Definition 3.2 (Crawled graph or Crawl). The (incomplete) graph derived from the set of webpages that have actually been visited by the crawler, discovered/linked yet uncrawled pages are not included. This subset of the target graph is illustrated by the *crawl* in Figure 1.

Definition 3.3 (Ghost vertex). Although a hyperlink on a crawled page points to another page that belongs to the target graph, there is a chance the crawler never visited and saved that page, i.e., it is not part the crawl. Such a page or vertex is referred to as *ghost vertex*, shown by the gray vertices outside the crawl in Figure 1.

Ranking Deviations. The deviation among two rankings induced by PageRank is a global objective, independent of a specific query. Hence, local or relevance-based measures such as nDCG are not applicable here. The most common metrics to quantify rank correlation are *Spearman's Rho* and *Kendall's Tau*, which are both similar as they are special cases of a more general correlation coefficient

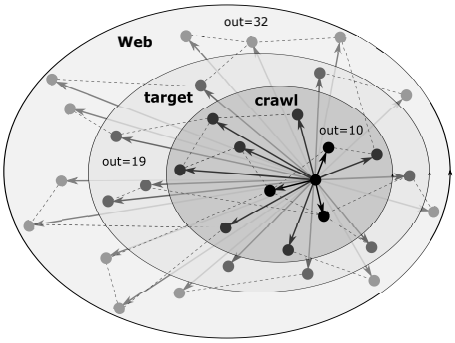

Figure 1: The neighborhood of a webpage in different subgraphs of the Web: The out-degree differs as neighbors become ghost vertices in the target graph or crawl. While the target represents the desired subset to be crawled, the crawl illustrates what has actually been captured, making this an incomplete graph (cp. Sec. 3).

and measure relative displacements. In this work, we use Kendall's Tau [24], ranging from $[-1, 1]$, with 1 corresponding to a perfect rank correlation, 0 corresponding to no correlation and -1 to a perfect inverse correlation. This is used to compare the correlation/deviation of rankings computed on the vertices of a crawl $\mathcal{G}_C$ with respect to that of the target graph $\mathcal{G}_T$.

In Figure 2 we provide a few examples of possible graph structures, where partial knowledge of the graph may affect the ranking returned by the PageRank values. We remark that in the next sections, we will also provide empirical evidence, supporting the fact that there exists a ranking deviations in crawls of some real-world graphs. In the first subfigure (a), we show the positive case of a DAG where the partial knowledge of the graph will not cause any ranking deviations. As only the topmost vertices shown here receive significantly more links than the others, these are also the most *important* vertices. It is easy to see here that generating a crawl from this structure by removing some vertices will not cause any significant changes in the ranking orderings of the crawl. In the next subfigure (b), a *backlink* has been introduced (left) that feeds back the importance of a top most page to a previously unimportant page and its successors. This importance gets propagated through the cycle which has been created due to the inserted *backlink*. In the next subfigure (c), we illustrate the case of a crawl in which vertices are removed uniformly at random. The chances here are that primarily unimportant vertices are removed, which would still not cause much deviations in the ranking orderings. Finally, if we remove any vertex from the cycle as shown in subfigure (d), its succeeding vertices drastically lose in importance and hence, the ranking among the pages in the crawl changes noticeably.

4 THE HAK MEASURE

With our measure, we estimate quantitatively how reliable a crawl is with respect to the relative ordering of the PageRank values on its vertices compared to the corresponding target graph. To this end, we first try to **estimate the size of the target graph**: Given the crawled vertex set and the distinct hyperlinks on the corresponding webpages, some of which are pointing to an uncrawled page (ghost vertex), how big is the target graph or a subgraph that would potentially impact or contribute to the PageRank values of the vertices in

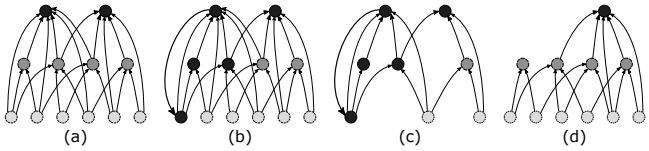

Figure 2: Some graph structures: A darker color of the vertices represents a higher *importance* (cp. Sec. 3).

the crawl? We show that for simple crawling strategies where it can be assumed that each vertex is part of the crawl independently from all other vertices with some sampling probability p_s, the size of the target graph can be estimated in terms of a very simple property of the crawled vertices, namely, the fraction of its crawled neighbors, referred to as *fidelity*. Secondly, we try to **estimate the impact** exerted by the vertices in the target graph on the crawled vertices, which we in turn use to estimate the number of discordant pairs in the expected rankings, like in Kendall's Tau.

Let C denote the set of vertices of the *crawl graph* and let n be the number of vertices in this graph. The main steps in our computation are as follows:

(1) Estimate the size of the target graph by using connectivity properties of the crawl. Let T represent the set of vertices in this target graph.
(2) Estimate the *impact* (as functions of PageRank) of the vertices in C.
(3) Assume that the vertices in T exert similar impacts on other vertices.
(4) Estimate the number of discordant pairs due to impacts exerted by vertices in $T - C$ on vertices in C.

4.1 Estimating the Target Graph

Let $\mathcal{N}$ denote the number of vertices in the target graph. In this section we will estimate the value of $\mathcal{N}$ under the simplified assumption that the crawl is constructed by sampling vertices from the target graph independently and uniformly at random with some probability p_s. Note that if p_s is known, one can easily estimate $\mathcal{N}$ as $\frac{n}{p_s}$. We therefore first estimate p_s from the connectivity of the crawl, using a property that we refer to as **fidelity**: For any vertex $v \in T$, we define fidelity ($\gamma(v)$) of v as the ratio of its immediate neighbors in C to its total out-degree (number of distinct hyperlinks on a webpage pointing to vertex in T). Let $d_c(v)$ count the number of vertices $v' \in C$ reachable from v in one step. $d(v)$ denotes the total out-degree of v in the target graph. This results in the following definition:

Definition 4.1 (Fidelity). The fidelity of a vertex $v \in T$, $\gamma(v)$, is given by $\gamma(v) = \frac{d_c(v)}{d(v)}$ and the average fidelity of all vertices in C is

$$\gamma(C) = \frac{\sum_{v \in C} \gamma(v)}{n}$$

With p_s as the sampling probability, $p_s \cdot \mathcal{N}$ would be the number of vertices in the crawl. Hence, using the observed average $\gamma(C)$ and the observed size of the crawl (n), we approximate $\mathcal{N}$ as $\frac{n}{\gamma(C)}$.

4.2 PageRank and Impacts

Despite its incompleteness, PageRank can be computed on the crawl graph by treating the ghost nodes as dangling nodes. We use the *personalized* variant of PageRank for this, starting from

the available nodes in C as seeds (s. Section 5). Given this, for any vertex v in the crawl C, let $\pi(v)$ denote the value computed by PageRank and let $N(v)$ denote the set of succeeding neighbors of v, reachable from v in one step, hence $d(v) = |N(v)|$. PageRank of any vertex u can now be considered as:

$$\pi(u) = \sum_{v:u\in N(v)} \frac{\pi(v)}{d(v)}$$

Based on these considerations, we introduce a new property, referred to as **impact**. The impact of a vertex $v \in C$ on one of its neighbors $u \in N(v)$ is defined as:

$$Im(v,u) = \frac{\pi(v)/d(v)}{\pi(u)}$$

Hence, the total impact on any vertex $u \in V$, received from all its incoming edges, is $\frac{1}{\pi(u)} \sum_{v:u\in N(v)} \frac{\pi(v)}{d(v)}$, which is always 1. This implies that any extra impact of x on a vertex will increase its PageRank by x times the current PageRank.

The total impact of a vertex v, $Im(v)$ is then defined as:

$$Im(v) = \sum_{u\in N(v)} Im(v,u) = \sum_{u\in N(v)} \frac{\pi(v)/d(v)}{\pi(u)} = \frac{1}{d(v)} \sum_{u\in N(v)} \frac{\pi(v)}{\pi(u)}.$$

We denote the average of impacts of vertices in C by $Im(C)$, i.e. $Im(C) = \frac{\sum_{v\in C} Im(v)}{n}$.

4.3 Estimating the Impact of Ghost Vertices

We next compute the impact that could have been exerted by the ghost vertices on the crawled vertices, if the graph was complete and the ghost vertices existed. In a setting like ours, where the (*personalized*) PageRank is computed from the perspective of the known crawl (see above), the ghost nodes cannot have a bigger impact on the crawl than previously *leaked* to them. Therefore, we build on the assumption that the impact of each vertex in the complete target graph T is on average the same as for the crawl: $Im(C)$. Hence, we approximate the impact exerted by ghost vertices only as follows:

$$\mathcal{I} = |T - C| \cdot Im(C) = n\left(\frac{1}{\gamma(C)} - 1\right) \cdot Im(C).$$

Some of this extra impact, generated due to ghost vertices, will be acquired by some or all of the vertices in C, changing their PageRank values accordingly. This is what eventually will lead to the deviation in rankings, measured by Kendall's Tau as the number of pairs of each two crawled vertices $(v,u) \in C \times C$ for which the order differs, i.e., *discordant pairs*, or is preserved, i.e., *concordant pairs*. Since HAK is meant to predict the deviation as assessed by Kendall's Tau, we also estimate both these classes of pairs in order to compute HAK.

The impact of the ghost vertices can be divided among the vertices of the crawl in several ways. For example, it can happen that the vertex with the lowest PageRank receives the total impact, increasing its PageRank by a large factor. In this case the number of discordant pairs is upper bounded by $n-1$. Moreover, we know from [31] that vertices with low original PageRank scores will also have a low PageRank value in slightly modified graphs. Therefore, the effect of the loss of information because of incomplete crawls is observed mostly on the PageRanks of the nodes higher in the original ranking. We checked experimentally several variants for impact distributions and the best variant, which is affirmative with our tests on real-world and synthetic graphs, is to distribute the total impact $\mathcal{I}$ equally among $\mathcal{I}$ vertices. Hence, the **expected number of impacted vertices** that belong to the crawled set will be:

$$I = \mathcal{I} \cdot \gamma(C).$$

In the worst case, each of these impacted vertices will result in forming a discordant pair with each of the unaffected vertex, resulting in a number of discordant pairs of $D = (n-I) \cdot I$. Based on that, HAK is computed with respect to Kendall's Tau as follows:

$$HAK = \frac{\text{\#concordant pairs - \#discordant pairs}}{\text{\# total pairs}} = \frac{\frac{n(n-1)}{2} - D - D}{\frac{n(n-1)}{2}} = 1 - 4 \cdot \frac{D}{n(n-1)}.$$

5 EXPERIMENTS

To validate our research questions enumerated in Section 1 we consider a host of large real-world graphs as well as synthetic graphs of different structures and carefully consider crawling strategies over them. In what follows we first describe our setup and rationale for our evaluation, before we discuss the results of our HAK experiments.

5.1 Experimental Setup

The described experiments require the availability of **crawls as well as the complete target graphs** that these crawls were derived from. This is necessary in order to compute how the rankings on both graphs differ and to evaluate the performance of HAK to estimate this deviation. In reality, neither obtaining the complete target graph is possible nor the actual crawl policy can be determined accurately. To this extent, we consider very large (as complete as possible) real-world graphs under the assumption that those graphs are complete (Sec. 5.1.1). We additionally simulate alternative topological structures by generating synthetic graphs (Sec. 5.1.2). We then simulate crawls on these graphs using different crawling strategies (Sec. 5.1.3). For all graph and crawl combinations we ran PageRank on both graphs (crawl and target graphs) and compared the rankings using Kendall's Tau to evaluate HAK (Sec. 5.1.4).

5.1.1 Real-World Graphs. The experiments on real-world graphs were run on a computer cluster using *Apache Spark* and its graph processing framework *GraphX* [36]. Loading the graphs locally on a single server was impossible with our available infrastructure because of their sizes of up to more than 100M vertices and 10B edges. As discussed earlier, we obtained multiple large real-world graphs that themselves were incomplete and considered them as target graphs by discarding edges that connect to ghost vertices. The following graphs were analyzed and are summarized in Table 1:

- **GOV :** This graph is based on crawled webpages provided by the *Internet Archive* [33]. It was extracted from the latest captures of all their archived webpages under the `.gov` top-level domain (TLD) from 2005 to 2013.
- **DE :** Like GOV, this `.de` TLD graph was also extracted from webpages archived by the *Internet Archive*, crawled in 2012 and generously provided to us in the project ALEXANDRIA[2].

[2] http://alexandria-project.eu

	GOV	DE	UK	Friendster
$\#V$	301,128,778	247,641,473	39,454,746	68,349,466
$\#V_{target}$	5,418,054	133,895,590	38,838,959	61,100,375
$\#E$	2,111,229,433	14,795,732,782	936,364,282	2,586,147,869
$\#E_{target}$	180,657,788	10,085,242,536	928,939,162	2,575,600,737

Table 1: Statistics on the studied real-world graphs, see Sections 5.1.3 and 5.1.1 for details ($\#V$: original number of vertices, $\#E$: original number of edges, $\#V_{\text{target}}$: target number of vertices, $\#E_{\text{target}}$: target number of edges).

- **UK :** This .uk TLD crawl from 2005 is publicly available, already in the form of a graph without corresponding webpages [5, 7].
- **Friendster :** Unlike the previous Web graphs, this is a publicly available social network, extracted from an extensive crawl of the former online platform *Friendster.com* in June 2011 [4].

5.1.2 Synthetic Graphs. In order to investigate ranking deviations caused by different crawling strategies on different graph topologies, we ran a more comprehensive set of experiments on smaller, synthetically generated target graphs. This allowed for more extensive experimentation as the experiments could run locally on a single server, using the *NetworkX* graph analysis framework [16]. All synthetic graphs that we studied in this work (cp. Table 2) were generated with 10,000 vertices to be reasonably sized for a thorough analysis.

The graphs were constructed using well-known graph generators, except for FFBacklinks, which is an extension by us to the ForestFire model. Although *Forest Fire* graphs include cycles, the model never generates *backlinks* from the early created vertices, which are more likely to receive many in-links over time, to newer ones. However, these links are common on the Web, where already prominent pages add links to less important ones, having a strong impact on the value propagation in PageRank (s. Sec. 3). In this graph, we added such edges between 0.05% of all pairs of an old and young vertex.

5.1.3 Seed Selection and Crawling. Crawling can be considered a special case of network sampling from a more practical point of view, where subsequent vertices can only be chosen from already discovered ones or seeds. Maiya and Berger-Wolf [30] define this type of sampling as *link-trace sampling* and give a nice overview of available models for this behavior. Naturally, such approaches commonly exhibit BFS-like (Breadth-First Search) growth, but feature different strategies to prioritize or select the next vertices to be crawled. These variations determine the probability of a vertex to be part of the final sample.

How we model crawls. Although most crawlers employ BFS-like traversals, there are practical constraints like random timeouts and crawl restrictions on websites that make it hard to model crawls perfectly. Therefore, we focus on the most impartial strategy, which is vanilla BFS, but explicitely **produce partial crawls** by dropping $x\%$ of the vertices of the input graph (where $x \in \{10, 20, 30, 40, 50\}$). We refer to this percentage as the *block fraction* and the remainder as *desired fraction*.

Statistics about the *target graphs* (V_{target} and E_{target}), which are potentially reachable from the seeds by not blocking any vertices are shown in Table 1. Additionally, we discuss a few edge cases by looking at slight variations of BFS as well as SEC with the synthetic graphs. Due to their scale it was computationally infeasible for us given our cluster setup to analyze these on real-world graphs as well. More details on the crawling strategies as well as our seed selection are given in the Appendix A.

Graph generator	*#Edges*	*Parameters*
$G_{n,p}$ **[11, 12]**	299,722	$p \approx 0.0003$ (based on $\#E$ in Table 1)
ScaleFree[8]	21,732	$\alpha = 0.41$, $\beta = 0.54$, $\gamma = 0.05$ (default)
ForestFire[27]	87,060	$p_f = 0.37$, $p_b = 0.32$ (most realistic [27])
FFBacklinks	96,262	$p_f = 0.37$, $p_b = 0.32$, $p_{\text{backlink}} = 0.0005$

Table 2: Synthetic graphs (all have 10,000 vertices).

5.1.4 Evaluation strategy. **The objective** of this evaluation is to assess ranking deviations as quantified by Kendall's Tau (cf. Sec. 3) for rankings induced by PageRank, computed on a complete target graph vs. an incomplete crawl and compare it against our HAK measure, which is designed to yield values on the same scale. For this, we **focus only on high-ranked vertices**, as these are typically more interesting in most practical scenarios [31]: Firstly, because there is no tangible score difference between the PageRank values of the tail vertices. Secondly, ranking deviations in authoritative vertices are typically considered more severe than among the tail ones. Since Kendall's Tau makes no distinctions where rank reversals take place, we compared the ordering among the top 30%, top 50% and top 70% vertices of the crawl and target graph that appeared in both graphs according to the corresponding PageRank values. This also helps us characterize where the rank reversals indeed do appear.

The rankings for each of the graphs are computed based on the PageRank values. While we employed the regular version PageRank on the crawl (with added ghost vertices as sinks), we used the *personalized* variant of PageRank for running it on the target graph. In this version, the algorithm is personalized to a set of vertices, which constitute the starting points as well as teleportation destinations in the algorithm [32]. The resulting PageRank values can be interpreted as their importance with respect to these vertices or the domain represented by the crawl. Both variants of PageRank ran for 30 iterations with the damping factor parameter set to the frequently cited value of 0.85.

5.2 Crawls and Ranking Deviations in Graphs

In this section, we aim to answer **RQ I** and justify the need for estimating ranking deviations before employing PageRank for incomplete graphs. In particular, we argue about the results where we witness noticeable ranking deviations of partial crawls with respect to target graphs.

We clearly observe that all real-world graphs exhibit a decreasing τ with increasing block fraction (see Figure 3). Most acutely, τ decreases to 0.55 for the GOV. Synthetic graphs like $G_{n,p}$ and FFBacklinks (first and last row in Figure 4) exhibit a similar trend with τ decreasing for increasing block fraction. On the other hand, for the ScaleFree (second row) and ForeFire graphs (third row), we do not witness much change in the ranking orderings, except in the BFS crawls.

A detailed study of the crawls reveals the reasons for such disparate trends for ScaleFree and ForeFire: the crawling strategy combined with the underlying structural properties of the graph sometimes lead to extremely small crawls ($n < 1,000$), much below the desired fraction (cp. Sec. 5.1.3). First, we observe a scarcity of *backlinks* in

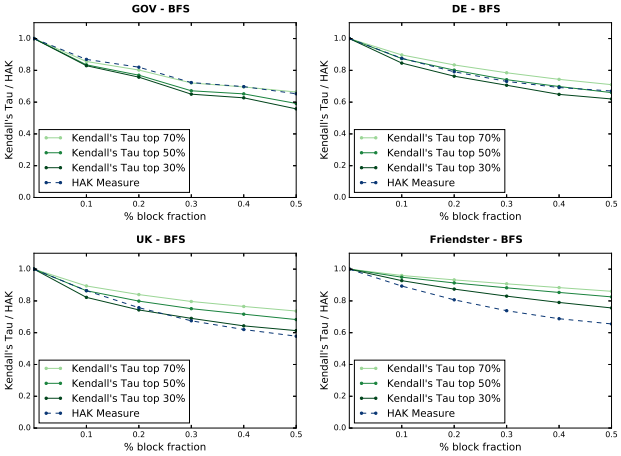

Figure 3: Ranking deviations measured and estimated for real-world graphs and crawls for different fractions of uncrawled vertices.

ForestFire and ScaleFree. That leads to these graphs to be *DAG-like* without an inadequate number of cycles in the corresponding graphs (cp. Sec. 3). PageRank computations over such graphs (or over their subgraphs) tend to finish quickly since the lack cycles prohibit the random walk to re-cycle back into the graph. This results in small high-fidelity crawls that do not exhibit large ranking deviations when highly linked vertices are prioritized, explicitly (SEC) or by chance (BfsRnd and BfsGeo). Only the BFS strategy that explicitly blocks random vertices causes a deviation in these crawls, as top vertices may be missed as well (conceivable on the Web for different reasons, e.g., restrictive policies and random failures).

Reinforcing our claim, the addition of backlinks in FFBacklinks resulted in a growing ranking deviation with increasing block fraction. We argue that most of the real-world graphs will not be DAG-like and will have *backlinks* inducing large cycles. Moreover, the random walk nature of PageRank computation increases the importance of these *backlinks* (or feedback loops) towards reaching an equilibrium state. As the core structure of FFBacklinks still resembles the original ForestFire graph, the observed rank deviation is much less severe as compared to $G_{n,p}$.

In addition, we observe that the ranking deviations (in most of the presented cases) increase when we consider a small fraction of the most important vertices. This indicates that most of the low rank vertices in the target graph do not flip their ranks with the more important ones in the crawl, leading to a lower ratio of discordant pairs to the overall total number of pairs. On the other hand, crucial to most applications are the ranking deviations of the *high* PageRank vertices, thus making it essential to monitor them.

Finally, we observe that ranking deviation in the Web graphs shown in Figure 3 are interestingly similar to the random graphs in Figure 4 and less so with other generative models like ForestFire or ScaleFree graphs. This, we believe, has strong implications in explaining the structure of Web graphs.

5.3 Effectiveness of HAK

We first discuss about the general applicability of the HAK measure and then argue about the supporting experimental evidence reported in Figures 3 and 4. We recall that the main assumption behind the construction of HAK is that each of the unseen or ghost vertices from the target graph would exert the same fraction of impact (on average) to the crawled set as the actual vertices in the crawl (cp. Sec 4). We ensure this by constructing the target graph such that each of its vertex has the same fraction of crawled neighbors as the crawled vertices (computed by fidelity). This assumption would not be followed by target graphs, which for example are *DAG-like*, because the ghost vertices there might not have edges back into the crawl. We remark that HAK cannot identify structures in target graph which are not similar to the crawl, yet leading to severe ranking changes in the crawl. For instance, consider a very small crawl with a very high fidelity and low impact. In such a case HAK would always estimate a very low ranking deviation. It could in the worst happen that there exist a few ghost vertices in the target graph with very high PageRank, having outgoing edges to only the low rank vertices in the crawl. Our results in figures 3 and 4, on the other hand, support the effectiveness of HAK in most of the studied graphs and therefore also validate our assumptions behind HAK.

We first discuss our findings on synthetic graphs. HAK performs fairly well for $G_{n,p}$, for instance with the BFS crawl strategy with 50% block fraction, we record an absolute error of 0.02 (actual: 0.24, estimated: 0.26) for rank correlation of top 30% vertices. The little ranking deviations in ScaleFree and ForestFire can be attributed to the small crawls with high fidelity ($\gamma \in [0.93, 1.0]$). As already discussed, HAK in these cases would always result in a high value, which also explains HAK adapting to the trends. However, we observe a larger deviation for BFS crawls in ScaleFree graphs. Here, HAK underestimates the ranking deviation, which might reflect the existence of the worst case (caused by the random vertex removal in BFS, cf. Sec. 5.1.3), resulting in a similar estimation as the one described above for very small crawls. However, HAK overestimates the deviation in FFBacklink (see the last 3 plots shown in Figure 4). We attribute this to the fact that the average impact of the crawl increases in presence of *backlinks* (cp. Sec. 3), which is an overestimation of the actual impact since *Forest Fire* is nevertheless the dominant topology in this graph. For our measure, a higher average impact corresponds to higher impact on the crawl from the ghost vertices (in our constructed target graph, cp. Sec. 4), leading to increased number of discordant pairs. The uniformly random blocking strategy in the BFS crawls on the other hand might break such *backlinks*, which lead to a more realistic ranking deviation as well as a better estimation of this deviation by HAK (actual: 0.29, estimated: 0.35).

We report more promising results in case of real-world graphs (s. Fig. 3). For instance, for the UK graph we report an almost precise estimation (actual: 0.58, estimated: 0.61). The observed trend in UK is more similar to that seen in to $G_{n,p}$ and FFBacklinks, which might also suggest existence of more *backlinks* in this graph, leading to large cycles (cp. Fig. 2). In contrast, the deviation in Friendster is less strong and slightly overestimated by HAK (actual: 0.76, estimated: 0.66) similar to ForestFire. We remark here that ForestFire also aims to model social networks and we believe that the similarity of these trends might be caused by the scarcity of *backlinks* in these graphs. We also note that our estimates reflect more closely the ranking deviations among the top PageRank vertices (in either of the compared rankings, cp. Sec. 5.1.4), which we believe to be more interesting for most practical purposes than deviations in less *important* vertices or the entire graph.

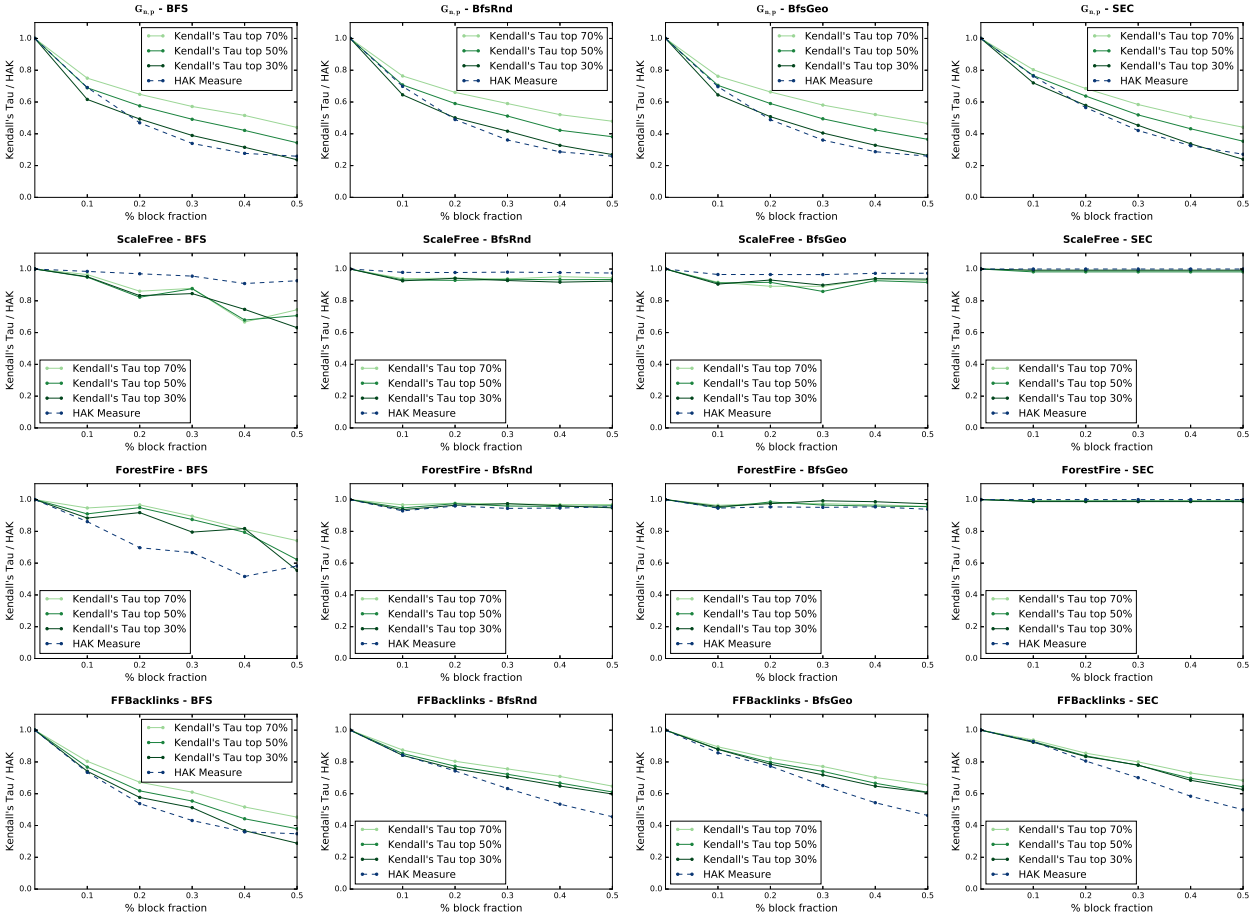

Figure 4: Ranking deviations measured and estimated with different synthetic graphs and crawls for different fractions of uncrawled vertices (rows) as well as different crawling strategies (columns).

In summary, the effectiveness of HAK is dependent on the fact that how well we estimate the target graph using properties like fidelity. Moreover, we would require big enough crawls to allow for a representative estimation[3] of the target graph. We believe that a more sophisticated use of the fidelity and impact of vertices in crawls, for instance using their distributions instead of simple average, will allow us to estimate the target graph and hence the ranking deviation more accurately. In addition, we plan to investigate more properties of the crawled graph, which can be used to predict the corresponding target graph. As our final goal, we will like to extend HAK as a black-box of measures, from which a suitable measure can be chosen in order to estimate ranking deviations in some particular crawl.

6 CONCLUSION

In this paper, we focused on the problem of PageRank deviations in Web graphs, typically caused by incomplete crawling. We established that deviations in ranking indeed do occur and can be drastic, as shown in our GOV graph where the correlation among the rankings is only 0.55, measured by Kendall's Tau. To this effect, we proposed the HAK measure, which can reliably estimate such deviations purely on the crawl without any knowledge of the original graph.

Our results suggest that incomplete Web graphs behave surprisingly similar to random graph models and quite different from other generative Web models, such as Forest Fire, in terms of PageRank deviations. Thus, this study on incompleteness in Web graphs could be important in studying the structure of the Web as well. For future work, it would be interesting to check if Web graphs are indeed composed of local random structures. Further, from the insights on the effect of backlinks in this work, we intend to look into other representative formal Web models. Finally, we would like to investigate the applicability of our measure to determine the confidence of results produced by other algorithms on incomplete graphs, such as random walk algorithms similar to PageRank.

A APPENDIX

A.1 Crawling Strategies

- **BFS :** The breadth-first search (BFS) starts from a set of seed vertices and runs until all vertices are reached. A number of vertices according to the block fraction were chosen uniformly at random and blocked/discarded before the BFS, simulating vertices that

[3]We do not want to give recommendations for a minimal size as this is dependent on the target graph and requires some knowledge about it, which should anyway exist when working with a crawl.

cannot be crawled, e.g., due to *robots.txt*, slow response times, etc.

- **BfsRnd :** Instead of blocking vertices from the beginning we determine a number of vertices to pursue at each vertex, chosen uniformly at random from its outgoing edges. Additionally, we remove a random number of vertices according to the block fraction from the seed set and run the BFS until the specified desired fraction is discovered.
- **BfsGeo :** Similar to BfsRnd, but the number of edges to follow was geometrically distributed with parameter $p = 0.3$, resembling *Forrest Fire Sampling* [26].
- **SEC :** In this *Sample Edge Count* stategy [30], at each step the number of edges from the crawled vertices to all remaining vertices are tracked and those with incoming edges are prioritized.

A.2 Seed Selection

We found out that the most realistic seed selection strategy is to pick the most important vertices as seeds. This is also the case for real crawls as these correspond to more well-known pages. To identify such pages in our target graphs, we first ran PageRank on them and constructed the seed set from the top 1%. This allowed us to reduce the size of the large real-world graphs by pre-computing the actual target graphs, consisting only of vertices that are reachable from the seeds (s. Table 1, V_{target} and E_{target}). Interestingly, for the GOV and DE graphs, the size difference of the target graphs compared to the originally provided graphs is huge, which confirms common characteristics of these Web archive graphs, i.e., they are not constructed in one crawl, leading to a fairly large number of unimportant vertices (with no in-edges) that were discovered from crawls outside target graphs. The UK and Friendster graphs on the other hand remained at almost the same size, suggesting that they have already been created that way in the first place, which proves our seed selection strategy actually realistic.

REFERENCES

[1] Scott G. Ainsworth, Ahmed Alsum, Hany SalahEldeen, Michele C. Weigle, and Michael L. Nelson. 2011. How much of the web is archived?. In *Proceeding of the 11th annual international ACM/IEEE joint conference on Digital libraries - JCDL '11*. ACM Press. DOI: https://doi.org/10.1145/1998076.1998100

[2] Sawood Alam, Michael L. Nelson, Herbert Van de Sompel, and David S. H. Rosenthal. 2016. Web Archive Profiling Through Fulltext Search. In *Research and Advanced Technology for Digital Libraries*. Springer International Publishing, 121–132. DOI: https://doi.org/10.1007/978-3-319-43997-6_10

[3] Ahmed Alsum, Michele C. Weigle, Michael L. Nelson, and Herbert Van de Sompel. 2013. Profiling Web Archive Coverage for Top-Level Domain and Content Language. In *Research and Advanced Technology for Digital Libraries*. 60–71. DOI: https://doi.org/10.1007/978-3-642-40501-3_7

[4] Archiveteam. 2011. Friendster Social Network Dataset: Friends. (2011). https://archive.org/details/friendster-dataset-201107 published under CC0 1.0 Universal.

[5] Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. 2011. Layered Label Propagation: A MultiResolution Coordinate-Free Ordering for Compressing Social Networks. In *Proceedings of the 20th international conference on World Wide Web*, Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar (Eds.). ACM Press, 587–596.

[6] Paolo Boldi, Massimo Santini, and Sebastiano Vigna. 2004. Do your worst to make the best: Paradoxical effects in pagerank incremental computations. In *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 168–180.

[7] Paolo Boldi and Sebastiano Vigna. 2004. The WebGraph Framework I: Compression Techniques. In *Proc. of the Thirteenth International World Wide Web Conference (WWW 2004)*. ACM Press, Manhattan, USA, 595–601. http://law.di.unimi.it/datasets.php

[8] Béla Bollobás, Christian Borgs, Jennifer Chayes, and Oliver Riordan. 2003. Directed Scale-free Graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '03)*.

[9] Miguel Costa, Daniel Gomes, and Mário J. Silva. 2016. The evolution of web archiving. *International Journal on Digital Libraries* 18, 3 (may 2016), 191–205. DOI: https://doi.org/10.1007/s00799-016-0171-9

[10] Anirban Dasgupta, Ravi Kumar, and Tamas Sarlos. 2014. On estimating the average degree. In *Proceedings of the 23rd international conference on World wide web*. ACM, 795–806.

[11] Paul Erdős and Alfréd Rényi. 1959. On random graphs. *Publicationes Mathematicae Debrecen* 6 (1959), 290–297.

[12] E. N. Gilbert. 1959. Random Graphs. *Ann. Math. Statist.* 30, 4 (12 1959), 1141–1144.

[13] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. 2011. Practical recommendations on crawling online social networks. *IEEE Journal on Selected Areas in Communications* 29, 9 (2011), 1872–1892.

[14] Daniel Gomes, Sérgio Freitas, and Mário J. Silva. 2006. Design and Selection Criteria for a National Web Archive. In *Research and Advanced Technology for Digital Libraries*. 196–207. DOI: https://doi.org/10.1007/11863878_17

[15] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. 2004. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 576–587.

[16] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*.

[17] Stephen J Hardiman and Liran Katzir. 2013. Estimating clustering coefficients and size of social networks via random walk. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 539–550.

[18] Taher H Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*. ACM, 517–526.

[19] Helge Holzmann, Wolfgang Nejdl, and Avishek Anand. 2016. The Dawn of today's popular domains: A study of the archived German Web over 18 years. In *Digital Libraries (JCDL), 2016 IEEE/ACM Joint Conference on*. IEEE, 73–82.

[20] Helge Holzmann, Wolfgang Nejdl, and Avishek Anand. 2017. Exploring Web Archives through Temporal Anchor Texts. In *Proceedings of the 2017 ACM on Web Science Conference - WebSci '17*. ACM Press, Troy, New York, USA. DOI: https://doi.org/10.1145/3091478.3091500

[21] Christian Hübler, Hans-Peter Kriegel, Karsten Borgwardt, and Zoubin Ghahramani. 2008. Metropolis algorithms for representative subgraph sampling. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 283–292.

[22] Hugo C. Huurdeman, Anat Ben-David, Jaap Kamps, Thaer Samar, and Arjen P. de Vries. 2014. Finding pages on the unarchived Web. In *IEEE/ACM Joint Conference on Digital Libraries*. IEEE. DOI: https://doi.org/10.1109/jcdl.2014.6970188

[23] Liran Katzir, Edo Liberty, and Oren Somekh. 2011. Estimating sizes of social networks via biased sampling. In *Proceedings of the 20th international conference on World wide web*. ACM, 597–606.

[24] Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika* 30, 1/2 (1938), 81–93.

[25] Jon M Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46, 5 (1999), 604–632.

[26] Jure Leskovec and Christos Faloutsos. 2006. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 631–636.

[27] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2007. Graph Evolution: Densification and Shrinking Diameters. *ACM Trans. Knowl. Discov. Data* 1, 1, Article 2 (March 2007).

[28] Rong-Hua Li, Jeffrey Xu Yu, Lu Qin, Rui Mao, and Tan Jin. 2015. On random walk based graph sampling. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, 927–938.

[29] Arun S Maiya and Tanya Y Berger-Wolf. 2011. Benefits of bias: Towards better characterization of network sampling. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 105–113.

[30] Arun S. Maiya and Tanya Y. Berger-Wolf. 2011. Benefits of Bias: Towards Better Characterization of Network Sampling. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*.

[31] Andrew Y. Ng, Alice X. Zheng, and Michael I. Jordan. 2001. Link Analysis, Eigenvectors and Stability. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2 (IJCAI'01)*. 903–910.

[32] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[33] The Internet Archive. 1996-2017. The Internet Archive. (1996-2017). http://archive.org

[34] Andrea Vattani, Deepayan Chakrabarti, and Maxim Gurevich. 2011. Preserving personalized pagerank in subgraphs. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 793–800.

[35] Tianyi Wang, Yang Chen, Zengbin Zhang, Peng Sun, Beixing Deng, and Xing Li. 2010. Unbiased sampling in directed social graph. In *ACM SIGCOMM Computer Communication Review*, Vol. 40. ACM, 401–402.

[36] Reynold S. Xin, Joseph E. Gonzalez, Michael J. Franklin, and Ion Stoica. 2013. GraphX: A Resilient Distributed Graph System on Spark. In *First International Workshop on Graph Data Management Experiences and Systems (GRADES '13)*.

[37] Zhuojie Zhou, Nan Zhang, Zhiguo Gong, and Gautam Das. 2016. Faster random walks by rewiring online social networks on-the-fly. *ACM Transactions on Database Systems (TODS)* 40, 4 (2016), 26.