Delusive PageRank in Incomplete Graphs

Helge Holzmann and Avishek Anand[1] and Megha Khosla

[1] L3S Research Center, Leibniz University, Appelstrasse 9a, 30169 Hannover, Germany
[2] KBS, Leibniz University, Appelstrasse 4, 30169 Hannover

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.

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 [19], 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 [6]. Incompleteness is an inherent trade-off already in the design decision of such an archive. Complicating matters further, Web archives are often not constructed in one piece but by merging partial crawls [13]. 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 [15], 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 [22] and its variants [17, 11].

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[3] based on (1) *inlinks* and (2) PAGERANKS. The above mentioned graph considered only links that emerged in 2012 [12]. 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?*

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' lastnames) that estimates the ranking deviation of an incomplete input graph when compared to the original graph.

2 Related Work

The authors in [21] 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, when some high ranked page

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

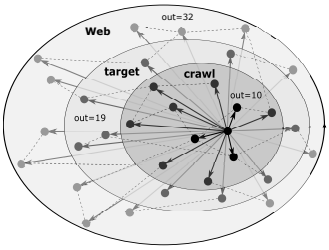

Fig. 1: The neighborhood of a webpage in different subgraphs of the Web.

is missed as we discussed in the previous section, the resulting PAGERANK rankings will be highly unstable. Boldi et al. [3] also show the paradoxical effects of PAGERANK computation on Web graphs. They however focus on crawling strategies to preserve page rank computation. In [26] 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. Other authors investigated this problem before as well, however, none of them focused on random walk algorithms, such as the widely used PAGERANK, neither explored the effect of missing nodes in real-world Web graphs [27, 23, 24].

The other area of related work comprises of graph sampling approaches which can be broadly classified into two categories: *traversal based* methods [18, 28, 20] and random walk based methods [19, 14]. 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 [28]. [20] 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 [29, 19, 14]. One of the popular sampling algorithms used for Web graphs is the *Forest Fire* algorithm by [18], 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.

3 Preliminaries and Problem

PageRank. As originally conceived, PAGERANK ranks vertices of a directed graph $\mathscr{G} = (V, E)$ where V and E are the vertices and edges respectively, based on the topological structure of the graph using random walks [22]. 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 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 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 (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 and measure relative displacements.

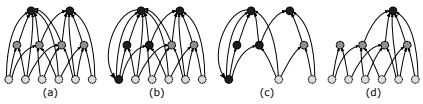

Fig. 2: Example graphs : Darker vertices have a higher *importance* (cp. Sec. 3).

In this work, we use Kendall's Tau [16], ranging from $[-1,1]$, with 1 corresponding to a perfect rank correlation, 0 corresponding to no correlation and -1 to a perfect inverse correlation, to compare the correlation/deviation of rankings computed on the vertices of a crawl $\mathscr{G}_C$ with respect to that of the target graph $\mathscr{G}_T$.
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 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.

Estimating the Target Graph. Let $\mathcal{N}$ denote the number of vertices in the target graph. We assume that the crawl is constructed by sampling vertices from the target graph independently and uniformly at random with some probability p_s. We first estimate p_s from the connectivity of the crawl, using a property that we refer to as **fidelity**. 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.

Definition 4 (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 $\gamma(C) = \frac{\sum_{v \in C} \gamma(v)}{n}$.*

We will now show that $\mathbb{E}(\gamma(v)) = p_s(1 - \mathbb{P}(d(v) = 0))$.

Proposition 1. *Let for some $0 < p_s < 1$, each vertex in the target graph is sampled independently and uniformly at random with probability p_s. For any $v \in T$, $\mathbb{E}(\gamma(v)) = p_s(1 - \mathbb{P}(d(v) = 0))$.*

Proof. The probability that a vertex has fidelity ℓ/k is given by

$$\mathbb{P}\left(\gamma(v)=\frac{\ell}{k}\right)=\mathbb{P}\left(d_c\left(v\right)=\ell|d\left(v\right)=k\right)\cdot\mathbb{P}\left(d\left(v\right)=k\right)=\binom{k}{\ell}p_s^{\ell}(1-p_s)^{k-\ell}\mathbb{P}\left(d\left(v\right)=k\right).$$

The expected value of fidelity of T can now be computed as

$$\begin{aligned}\mathbb{E}\left(\gamma(v)\right)=&\sum_{k\geq 1}\sum_{\ell\leq k}\frac{\ell}{k}\binom{k}{\ell}p_s^{\ell}(1-p_s)^{k-\ell}\mathbb{P}\left(d\left(v\right)=k\right)\\ =&p_s\sum_{k\geq 1}\mathbb{P}\left(d\left(v\right)=k\right)\sum_{\ell=1}^{k}\binom{k-1}{\ell-1}{p_s}^{\ell-1}(1-p_s)^{k-\ell}=p_s(1-\mathbb{P}\left(d\left(v\right)=0\right)).\end{aligned}$$

With p_s as the sampling probability, $\mathscr{N}\cdot p_s$ gives us the expected number of vertices in the crawl. Using Proposition 1 we obtain $\mathscr{N}=\frac{\mathbb{E}(|C|)}{\mathbb{E}(\gamma(v))}(1-\mathbb{P}\left(d\left(v\right)=0\right))$. We note that for Web graphs $\mathbb{P}\left(d\left(v\right)=0\right)$ is the probability that a webpage has no links to other webpages, i.e., there exists a page with pure text and no links. Such a scenario is extremely rare on the Web. Moreover, for synthetic graphs generated using $G_{n,p}$ one can show that $\mathbb{P}\left(d\left(v\right)=0\right)=e^{-O(np)}$, which goes to zero for $n\to\infty$ and constant p. Hence, using the observed average $\gamma(C)$ and the observed size of the crawl, i.e., n, ignoring the multiplicative factor of $1-\mathbb{P}\left(d\left(v\right)=0\right)$ (as $\mathbb{P}\left(d\left(v\right)=0\right)=o(1)$ for all practical purposes), we can approximate $\mathscr{N}$ as $\frac{n}{\gamma(C)}$.

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 immediate neighbors of v. PAGERANK of any vertex u can now be considered as: $\pi(u)\ =\ \sum_{v:u\in N(v)}\frac{\pi(v)}{d(v)}$.
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:

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

We denote the average of impacts of vertices in C as $m(C)=\frac{\sum_{v\in C}Im(v)}{n}$.

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 PAGERANK is computed from the perspective of the known crawl, 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: $\mathscr{I}=|T-C|\cdot Im(C)=n\left(\frac{1}{\gamma(C)}-1\right)\cdot Im(C)$.

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. 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 [21] 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 $\mathscr{I}$ equally among all vertices. Hence, the **expected number of impacted vertices** that belong to the crawled set will be: $I = \mathscr{I} \cdot \gamma(C)$. 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

The objective of this evaluation is to assess ranking deviations using 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. In contrast to the Kendall's Tau formula used in HAK, for assessing the real rankings, ties were considered by using a variant, also known as *Tau-b*. We **focus only on high-ranked vertices**, as these are typically more interesting in most practical scenarios [21]. 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.
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. 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.

Experimental Setup. Our experiments require both target graphs and the crawls 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 (Table 1). We additionally simulate alternative topological structures by generating synthetic graphs (Table 2). We then simulate crawls on these graphs using different crawling strategies. For all graph and crawl combinations we ran PAGERANK

	GOV [25]	DE[4]	UK [4]	Friendster [2]
$\#V$	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 ($\#V$: original number of vertices, $\#E$: original number of edges, $\#V_{\text{target}}$: #target vertices, $\#E_{\text{target}}$: #target edges).

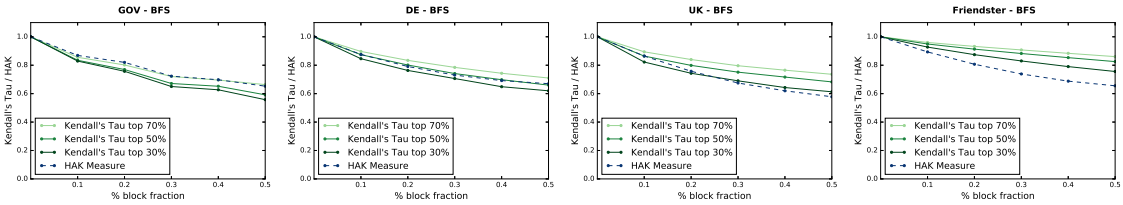

Fig. 3: Ranking deviations measured and estimated for real-world graphs and crawls for different fractions of uncrawled vertices.

on both crawl and target graphs and compared the rankings using Kendall's Tau to evaluate HAK.

Crawls and Ranking Deviations in Graphs. We aim to answer **RQ I** and justify the need for estimating ranking deviations before employing PAGERANK for incomplete 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. First, we observe a scarcity of *backlinks* in 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 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

Graph generator	*#Edges*	*Parameters*
$G_{n,p}$ **[8, 9]**	299,722	$p \approx 0.0003$ (based on *#E* in Table 1)
ScaleFree[5]	21,732	$\alpha = 0.41, \beta = 0.54, \gamma = 0.05$ (default)
ForestFire[18]	87,060	$p_f = 0.37, p_b = 0.32$ (most realistic [18])
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).

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.

Effectiveness of 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

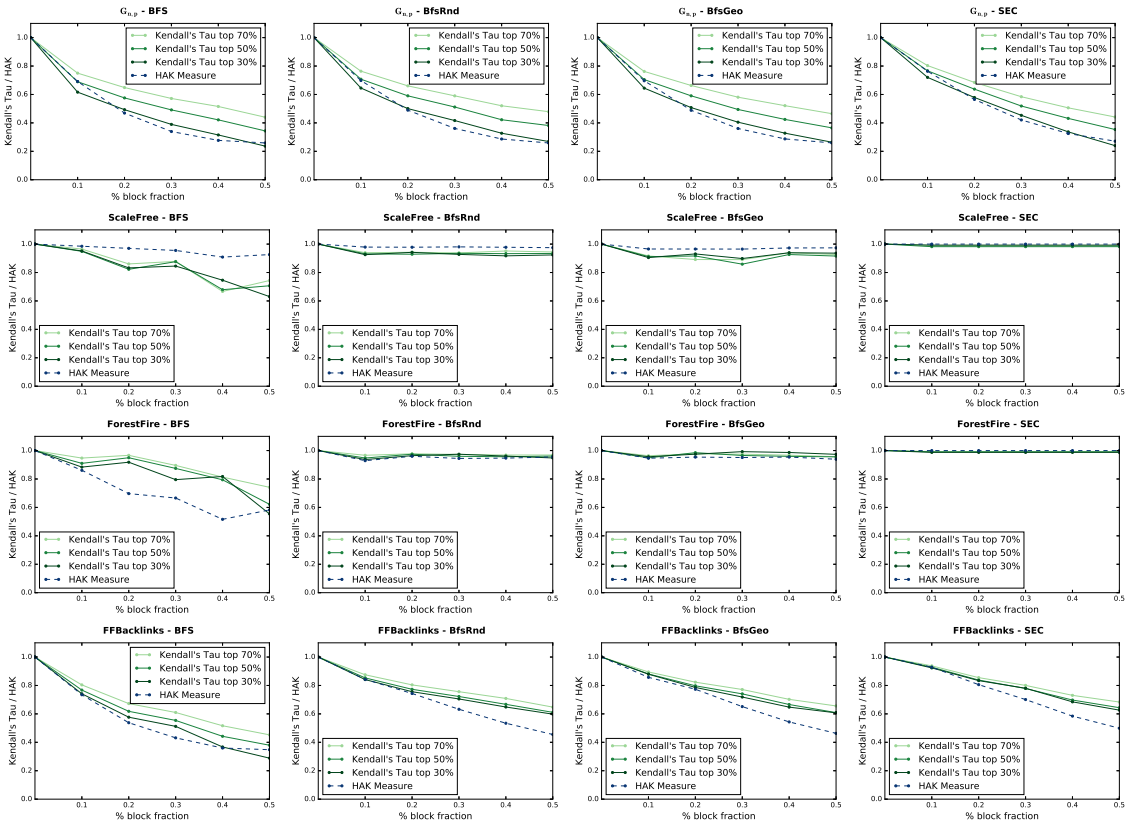

Fig. 4: Ranking deviations measured and estimated with different synthetic graphs and crawls for different fractions of uncrawled vertices (rows) as well as different crawling strategies (columns).

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 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.

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.

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.

References

1. Scott G. Ainsworth, Ahmed Alsum, Hany SalahEldeen, Michele C. Weigle, and Michael L. Nelson. How much of the web is archived? In *Proceeding ACM/IEEE- JCDL'11*.
2. Archiveteam. Friendster Social Network Dataset: Friends, 2011. published under CC0 1.0 Universal.
3. Paolo Boldi, Massimo Santini, and Sebastiano Vigna. Do your worst to make the best: Paradoxical effects in pagerank incremental computations. In *WAW*, 2004.
4. Paolo Boldi and Sebastiano Vigna. The WebGraph framework I: Compression techniques. In *Proc. of the Thirteenth International World Wide Web Conference (WWW 2004)*, pages 595–601, Manhattan, USA, 2004. ACM Press.
5. Béla Bollobás, Christian Borgs, Jennifer Chayes, and Oliver Riordan. Directed scale-free graphs. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, SODA '03.
6. Miguel Costa, Daniel Gomes, and Mário J. Silva. The evolution of web archiving. *International Journal on Digital Libraries*,191–205, 2016.
7. Anirban Dasgupta, Ravi Kumar, and Tamas Sarlos. On estimating the average degree. In *Proceedings of conference on World wide web*, pages 795–806. ACM, 2014.
8. Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.
9. E. N. Gilbert. Random graphs. *Ann. Math. Statist.*, 30(4):1141–1144, 12 1959.
10. Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *SciPy2008*, 2008.
11. Taher H Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM, 2002.
12. Helge Holzmann, Wolfgang Nejdl, and Avishek Anand. Exploring Web Archives through Temporal Anchor Texts. In *Proceedings of ACM Web Science Conference - WebSci'17*.

13. Helge Holzmann, Wolfgang Nejdl, and Avishek Anand. The dawn of today's popular domains: A study of the archived german web over 18 years. In *Digital Libraries (JCDL)*, 2016.
14. Christian Hübler, Hans-Peter Kriegel, Karsten Borgwardt, and Zoubin Ghahramani. Metropolis algorithms for representative subgraph sampling. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 283–292. IEEE, 2008.
15. Hugo C. Huurdeman, Anat Ben-David, Jaap Kamps, Thaer Samar, and Arjen P. de Vries. Finding pages on the unarchived web. In *IEEE/ACM JCDL*,2014.
16. Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.
17. Jon M Kleinberg. Authoritative sources in a hyperlinked environment.
18. Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007.
19. Rong-Hua Li, Jeffrey Xu Yu, Lu Qin, Rui Mao, and Tan Jin. On random walk based graph sampling. In *Data Engineering (ICDE), 2015* , 2015.
20. Arun S Maiya and Tanya Y Berger-Wolf. Benefits of bias: Towards better characterization of network sampling. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 105–113. ACM, 2011.
21. Andrew Y. Ng, Alice X. Zheng, and Michael I. Jordan. Link analysis, eigenvectors and stability. In *Proceedings of International Joint Conference on Artificial Intelligence*, 2001.
22. Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
23. Jeffrey A. Smith and James Moody Structural effects of network sampling coverage I: Nodes missing at random. In *Social Networks*, volume 35, pages 652–668. Elsevier, 2013.
24. Jeffrey A. Smith, James Moody and Jonathan H. Morgan Network sampling coverage II: the effect of non-random missing data on network measurement. In *Social Networks*, volume 48, pages 78–99. Elsevier, 2017.
25. The Internet Archive. The Internet Archive, 1996-2017.
26. Andrea Vattani, Deepayan Chakrabarti, and Maxim Gurevich. Preserving personalized pagerank in subgraphs. In *Proceedings of ICML*, 2011.
27. Dan J. Wang, Xiaolin Shi, Daniel A. McFarland and Jure Leskovec. Measurement error in network data: A re-classification. In *Social Networks*, volume 34, pages 396–409. Elsevier, 2012.
28. Tianyi Wang, Yang Chen, Zengbin Zhang, Peng Sun, Beixing Deng, and Xing Li. Unbiased sampling in directed social graph. In *ACM SIGCOMM Computer Communication Review*, volume 40, pages 401–402. ACM, 2010.
29. Zhuojie Zhou, Nan Zhang, Zhiguo Gong, and Gautam Das. Faster random walks by rewiring online social networks on-the-fly. *ACM Transactions on Database Systems (TODS)*, 2016.