Stopping Personalized PageRank without an Error Tolerance Parameter

Emmanouil Krasanakis, Symeon Papadopoulos and Ioannis Kompatsiaris
CERTH-ITI, 57001, Thessaloniki, Greece
Email: {maniospas,papadop,ikom}@iti.gr

Abstract—**Personalized PageRank (PPR) is a popular scheme for scoring the relevance of network nodes to a set of seed ones through a random walk with restart process. Calculating the scores of all network nodes often involves the power method, which iterates the PPR formula until convergence to an empirically selected numerical tolerance. However, finding a tolerance that is not so lax as to impact pairwise node comparisons but not so strict as to require a high number of iterations to converge requires time-consuming empirical investigation. In this work we aim to avoid this investigation by stopping power method iterations when node score order is robust against subsequent changes. To do this, we analyse the expected fraction of random walks considered at a given iteration and identify a potential stopping point that depends on a (fixed) confidence level of future iterations preserving node order. Experiments on four real-world networks show that a confidence level of 98% runs in a fraction of the time and yields more than 0.999 Spearman correlation with the node order of 10^{-20} numerical tolerance. Furthermore, that stopping point is comparable to empirically selecting a numerical tolerance that yields robust node order.**

Index Terms—**Network theory; Ranking; Convergence of numerical methods; Robustness**

I. INTRODUCTION

In many domains, data can be organized into networks whose edges reflect relations between nodes. For example, in social media networks, edges between user nodes are used to represent their social relations [1]. A problem that often arises when analyzing networks is scoring nodes based on their relevance to communities whose members are either structurally close or share the same metadata attributes. To find such communities, a popular practice [2], [3], [4], [5] is to utilize sets of known seed nodes that help discover other members. For example, seed nodes could be social media users with the same declared music genre preferences who form communities with others likely to share these preferences.

Methods that score network nodes given a set of seed nodes often follow a *random walk with restart* formulation [6], i.e. they employ a Markov process that, starting from the seed nodes, at each step either jumps to a neighbor or teleports back to one of the seeds. Then, each node assumes a score proportional to the probability of this process arriving on it. The most well-known random walk with restart strategy is *personalized PageRank* [6], [7], [8], [9], in which the probability of jumping from a node v to a particular neighbor u is proportional to the corresponding element $M[u,v]$ of a (column-wise or Laplacian) normalization of the network's adjacency matrix.

If we organize the seed nodes into a personalization vector s that represents a seed node distribution, the scores $r[v]$ of network nodes v calculated through this process converge to the solution of the linear system:

$$r = \alpha M r + (1 - \alpha)s \tag{1}$$

where $1 - \alpha$ is the teleportation probability to a random seed node, also called restart probability on merit that it restarts the random walk.

Personalized PageRank (PPR) is used in a multitude of graph mining tasks, ranging from recommender systems to discovering well-separated structural communities [10], [11] and propagating information in graph neural networks [12], [13]. The simplest PPR algorithm implementation is iterating the above equation (L1-normalizing s and r at each step to ensure convergence) until a small error is reached [14]. This process is known as the *power method* and has been implemented in many graph processing libraries, such as the *networkx*[1] Python graph manipulation library. When M has lots of zeros, as happens in most real-world networks, sparse matrix multiplication can be used to make computations efficient [15].

The power method requires several iterations to diffuse scores many hops away from the seed nodes [16]. Despite this shortcoming, as we explain in Section II, certain problems require scoring all network nodes. For these problems, employing the power method cannot be avoided, but it is still important to use as few computations as possible to calculate robust node scores. PPR algorithms, such as the ones detailed in Section II, traditionally rely on the notion of numerical robustness, i.e. that the found scores are numerically close to the ideal ones. However, in this work we focus on *node order robustness*, in the sense that ordering nodes by their scores should achieve an order similar to their ideal scores. To understand why this task is important, we point out that, although scoring errors are traditionally used as an indication of convergence, comparing and ordering node scores is the true objective of many practical applications of PPR.

Requiring both speed and robustness often creates a dilemma of when to stop the power method; stopping it too early may heavily influence the node order arising from calculated scores, whereas tight numerical convergence leads to disproportionately many calculations. These problems would

---

[1]https://networkx.github.io/

persist even if we stopped the power method using measures of score similarity. In particular, summarizing score order differences between consecutive power method iterations in one quantity would involve aggregating all pairwise node comparisons, as done by the Spearman correlation that measures the statistical correlation of score orders [17]. Therefore, once the differences between low- and high-scored nodes start becoming clear, such measures could detect small node order differences, when in reality the latter could still be shifting to nearby values. For example, in Section V we show that even score orders exhibiting over 0.999 Spearman correlation could be of low quality and need to be corroborated by empirical investigation of node order scatterplots. To solve this problem, it is tempting to compare score orders through more sensitive heuristics, such as counting the number node scores that change between consecutive power method iterations. However, these could also cause early stopping, when multiple iterations are needed to change the order between two nodes, or late stopping, when only a few nodes keep changing scores.

Based on the above, researchers should investigate node order robustness by experimenting with different numbers of iterations (which correspond to different numerical tolerance levels). They then need to capture the point at which obtained scores are empirically determined to be similar to ideal ones. In practice, this could involve calculating scores up to an exceptionally tight (e.g. 1^{-20}) numerical error tolerance and using these as the ground truth with which to assess the efficacy of scores obtained by looser tolerances. Then, the tolerance that achieves the best results on the particular graph or domain can be selected for deploying similar applications. Unfortunately, this investigation would require running many PPR algorithms and up to very tight numerical tolerances. The latter correspond to prohibitively large running times, such as needing $10\times$ as many iterations as the stopping point eventually selected.

In this paper, we address this problem by determining an early stopping point for the power method that achieves a robust node order. In particular, in Section III we analyse the fraction of random walks accounted at a given iteration of the power method and in Section IV we use this analysis to determine an iteration whose subsequent ones are expected to on average preserve the found node order within any desired confidence level. In Section V we demonstrate the usefulness of this practice on four real-world networks, on which it automatically (i.e. without needing human-supervised convergence parameter selection) determines convergence points of robust node ordering. These points lie at a similar number of iterations to the ones that would have been selected after manual inspection.

II. BACKGROUND AND MOTIVATION

As the power method is the most well-known scheme for finding the PPR scores of nodes up to a fixed numerical precision [9], [18], it has been the subject of many optimizations that reduce the number of calculations required for it to converge. These include the Jacobi method for handling dangling nodes [19], the Gauss-Seidel method [20] that updates only the score of one node at a time and extrapolation methods [21], [22] that adjust scores based on previous ones. More recent research has moved to developing Arnoldi-based algorithms [22], [23], [24] in the Krylov space $span\{M^k s | k = 1, \ldots, K\}$. The last type of methods also finds numerically robust solutions for low restart probabilities. In this paper we focus on the non-optimized power method, which is easier to analyse, and leave combining our approach with optimization algorithms for future work.

Researchers have also explored fast approximations of PPR with correctness guarantees only for a few nodes. Most of these employ a push-pull procedure, in which nodes push their score changes to their neighbors while pulling the latter's values to improve their own score approximation [25], [26]. This process stops when it achieves guarantees on the correctness of a fixed number of top scores [16], [27], [28]. Other fast-running approaches respect the local aspect of approximations [29] or are focused on stochastic processes that do not parse all network edges [30], [9].

Despite the precision and computational efficiency of top node discovery methods, they are often designed with recommender systems in mind, for which only these nodes are important. Hence, they scale poorly to scoring many graph nodes, especially those pertaining less to metadata groups. For example, in an artist discovery platform where the metadata groups of interest are music genres, it could be important to accurately score the relatedness of artists to genres to obtain a genre distribution for all artists and not only the ones most related to the genre. Another occasion where lower-importance scores may be useful is when they are aggregated (e.g. multiplied) with other insights or when their changes are the ones used to identify emerging nodes. For example, in community detection tasks, a process that quickly finds well-separated communities employs a division of scores with node degrees [31]. In such cases, we need to fall back on calculating scores with some adaptation of the power method.

A common problem of both power method adaptations and the fastest top node discovery algorithms [27] is that they require disproportionately many calculations to converge to numerically precise solutions as the restart probability approaches zero. This problem is not as apparent for 15% restart probability, which has been empirically established as the go-to value of non-personalized PageRank. However, the probability most suitable to discovering communities from a personalization vector often depends on the community's structural characteristics, such as size, density and conductance [31], [32]. For example, more distinguished (e.g. larger and denser) communities require longer random walks to reach all their members from their seed nodes. These longer random walks correspond to restart probabilities as low or lower than 1% [22], [33] and in turn introduce in some form the computational cost of spreading scores more hops away.

Given the above, we recognize that one of the most important problems existing methods encounter is the increased running time of smaller restart probabilities, especially when

243

needing to achieve numerical robustness. To alleviate a portion of this cost, researchers usually select laxer convergence tolerances. However, no guidelines have been provided on where to stop while at least preserving node order robustness. In this work, we take a first step towards exploring this direction.

III. Considering Most Random Walks

To express the power method in a formal manner, we denote by r_n the score vector calculated at iteration n, where $r_n[u]$ is the score of node u at that iteration. Following the formulation of PPR that preserves the stochastic nature of random walks, we focus on a column-wise normalization on the network's adjacency matrix A:

$$r_{n+1} = aMr_n + (1-a)r_0$$
$$r_0[u] = \frac{s[u]}{\sum_u s[u]}, \quad M[u,v] = \frac{A[u,v]}{\sum_u A[u,v]} \tag{2}$$

The power method traditionally repeats the first of these equations until score changes become on average less than a given tolerance. In this work, we focus on removing the need for human-driven estimation of this tolerance; instead, we search for an automatically selected point that provides node order close to the ideal one.

The above formalism interprets scores $r_n[u]$ as the probabilities of arriving at nodes u at iteration n and then either hopping to a random neighbor with probability a or restarting from a random seed node (selected from the distribution r_0) with probability $1-a$. Restarting marks the end of the previous random walk and the beginning of a new one. Since this event has a fixed chance of success or failure, it is a Bernoulli trial. Therefore, the random variable N that shows the length of random walks before restarting follows the geometric distribution $P(N=n) = a^{n-1}(1-a)$.

In this section, we investigate the robustness of node order from the viewpoint of considering a large enough fraction of random walks. To do this, we start by analysing the fraction of random walks considered up to a given iteration of the power method. In particular, we repeat the random walk process an arbitrarily large number of times and extract a large number of walks $W(n)$ up to length n. Thanks to the stochastic definition of the power method and the law of large numbers, $r_n[v]W(n)$ of walks contribute to the scores of nodes v at its n-th iteration. Therefore, node score robustness can be achieved if these arrivals are many more than the number of longer random walks not yet considered. This can be formally expressed by constraining the fraction of expected random walks up to length n to be close to the total number of walks, even if not all lengths are considered, i.e. for the fraction $p = E[W(n)]/E[W(\infty)]$ to reside close to 1, where $E[\cdot]$ is the expected number of walks ($E[W(\infty)]$ is the expected number of walks of any length).

If this happens, walks that arrive later on and permute scores would be (on average) too few to change the order of pairwise node comparisons. For example, let us consider that two nodes u, v exhibit close scores $r_n(u) = 0.2$ and $r_n(v) = 0.188$ at an iteration n, which considers $p = 0.99$ of random walks. At

worst, none of the remainder walks would arrive on u from that point on and all would arrive on v. This yields:

$$E\big[W(\infty)r_\infty(u)\big] \geq E\big[W(n)r_n(u)\big]$$
$$\Leftrightarrow r_\infty[u] \geq pr_n(u) = 0.198$$
$$E\big[W(\infty)r_\infty(u)\big] \leq E\big[W(n)r_n(v) + (W(\infty) - W(n))\big]$$
$$\Leftrightarrow r_\infty[u] \leq pr_n[v] + (1-p) = 0.19612$$

which preserves the order of these two nodes. In practice, it is unlikely that all random walks of greater lengths would end exclusively at node v and avoid node u. Hence, even closer scores could be preserved.

A critical component of the above analysis is comparing the number of random walks of length up to n to the total number of random walks. To help do this, we define several independent sub-processes $walker_k, k = 1, \ldots, n$ that start from the seed nodes and repeatedly perform random walks of exactly k steps before restarting. Furthermore, we consider the random walk with restart process to span intermediate steps $\ell = 1, \ldots, w$ of transitioning to the next node, where $w \to \infty$ is an arbitrarily large number. For example, w could be the number of steps required to infer node scores within a tight numerical precision by counting how many arrive on each node. Then, the random walk with restart process can be equivalently formulated in a sequential (instead of parallel) manner by selecting a sub-process $walker_k$ at random and then performing k steps before restarting from a new seed and sub-process. To express this behavior at any intermediate step ℓ of this sequence, we employ the following random variables:

$S(\ell) = \{1 \text{ if at iteration } \ell, 0 \text{ otherwise}\}$

$X_k(\ell, n) = \{1 \text{ if } \ell \text{ is performed by } walker_k, 0 \text{ otherwise}\}$

$W_k(\ell, n) = \{1 \text{ if } walker_k \text{ restarts at } \ell, 0 \text{ otherwise}\}$

Since each intermediate step occurs once, $E[S(\ell)] = 1$. Furthermore, the random walk sub-processes cannot be interrupted, which makes the probability of selecting one equal to the probability of performing a random walk of the respective length, i.e. $E[X_k(\ell, n)] = P(N = k)$. Additionally, intermediate steps ℓ correspond on average to random points within sub-processes k that perform random walks of that length, which makes the probability of restarting at that particular point $E[W_k(n)] = w/k$. Then, the number of random walks up to iteration n is equal to the number of restarts:

$$W(n) = \sum_{\ell=1}^{w} \left(S(\ell) \sum_{k=1}^{n} X_k(\ell, n) W_k(\ell, n) \right)$$

Finally, the above random variables are independent to each other. Therefore, the expected value operator $E[\cdot]$ yields:

$$E[W(n)] = w \frac{1-a}{a} \sum_{k=1}^{n} \frac{a^k}{k}$$

Since $\sum_{k=1}^{\infty} \frac{a^k}{k} = -\ln(1-a)$, this yields that the fraction p of expected random walks considered at iteration n is:

$$p = \frac{E[W(n)]}{E[W(\infty)]} = -\frac{\sum_{k=1}^{n} \frac{a^k}{k}}{\ln(1-a)} \tag{3}$$

This equation facilitates the discovery of robust node scores by stopping the power method when a large enough fraction of random walks has been considered. A simple take would be selecting a network-agnostic fraction of needed random walks, such as $p = 99\%$. Although this practice does not alleviate the problem of needing to select a stopping parameter, that parameter becomes easier to interpret and empirically justify. Furthermore, this selection can help determine the running time of the power method from its first iteration. In particular, multiplications involving the network's adjacency matrix can be computed in time $O(m)$, where m is the number of network edges through which multiplication iterates. Therefore, the time needed for the power method to converge is $O\big(m \cdot n(a,p)\big)$, where $n(a,p)$ is the number of iterations (and hence matrix multiplications) obtained by the solution of Equation 3 with respect to n.

IV. STOPPING AT A ROBUST NODE ORDER

The diffusion rate of seed node scores through the network is not only affected by the restart probability of the PPR scheme but also by the network's structural characteristics and the placement of seed nodes. For example, random walks need length at least equal to the network's diameter to diffuse the scores of non-central seeds to all other nodes and half that length if the seeds are centrally positioned. At the same time, reaching a robust node order in networks that suffer from slow diffusion requires a larger fraction of random walks, as more and longer walks are needed to propagate scores many hops away from the seed nodes.

To address this problem, in this section we propose a criterion for stopping the power method at the point when differences between node scores are on average sufficiently large to avoid being bridged by the remaining random walks. Since score diffusion speed is implicitly linked to the gaps between node scores, this approach captures to some extent all aspects of the scoring task that affect node order robustness.

To analyse the differences between node scores, we first refer to the example of the previous section, in which we explored the concept of maintaining the order $r_n[u] - r_n[v] > 0$ between two nodes u, v when lengthier random walks are also considered. In that example, we explained that, at worst, $r_\infty[u] \geq p\, r_n[u]$ and $r_\infty[v] \leq p\, r_n[v] + (1 - p)$. Therefore, a sufficient condition for maintaining the order between two such nodes would be:

$$r_\infty[u] - r_\infty[v] > 0 \Leftarrow \frac{1-p}{p} \leq r_n[u] - r_n[v] \qquad (4)$$

Unfortunately, satisfying this equation for all node pairs leads to the adoption of late stopping points, as the worst case is unlikely to occur and hence needs not be guarded against. Even in the scenario where most walks arrive at a small set of nodes (instead of only one), few other walks remain to impact the rest of node scores, hence preserving most pairwise node score comparisons.

Motivated by these considerations, we recognize the importance of preserving the outcome of score comparisons for most but not necessarily all pairs of nodes. This can be equivalently achieved if we preserve comparisons between most consecutively ordered node scores. For example, if we obtain the scores $[0.2, 0.25, 0.1, 0.3, 0.15]$, 0.25 and 0.3 are consecutively ordered, since sorting makes them end up at positions next to each other, and we would focus on directly preserving their order. On the other hand, 0.1 and 0.2 are not consecutively ordered, as 0.15 lies between them; hence, their order is implicitly preserved if we maintain the sub-orders $0.1 < 0.15$ and $0.15 < 0.2$ in future iterations.

To facilitate our analysis, we denote by $\mathcal{P}_n = \{(u, v) : r_n[u] > r_n[v], \nexists s : r_n[u] > r_n[s] > r_n[v]\}$ the pairs of consecutively ordered nodes at iteration n. Furthermore, we assume that an arbitrarily large number of random walks $W(n)$ are considered at that iteration. This yields:

$$E[W(\infty)r_\infty[v]] = E\big[W(n)r_n[v] + (W(\infty) - W(n))\delta_n[v]\big]$$
$$\Rightarrow r_\infty[v] = p\, r_n[v] + (1 - p)\delta_n[v]$$

where $\delta_n[v]$ are the expected node score differences induced by random walks of length greater than n. This quantity aggregates node arrival probability differences between walks of up to given length n and other walks of increasingly higher lengths. Since each walk of higher length contributes with the same weight to $\delta_n[v]$, the central limit theorem dictates that this quantity follows a normal distribution if n is large enough. The mean value of this distribution is approximated by $r_n[v]$ and hence the random variable $\delta_n(u, v) = \delta_n[u] - \delta_n[v]$ also follows normal distribution with mean value $r_n[u] - r_n[v]$. Furthermore the variance of this new variable can be approximated by $Var_{(u,v)\in\mathcal{P}_n}\big[\delta_n(u, v)\big] = \sigma_n^2$, where:

$$\sigma_n^2 = Var_{(u,v)\in\mathcal{P}_n}\big[r_n[u] - r_n[v]\big] \qquad (5)$$

Given these properties, we can select a confidence level p_{conf} of preserving on average the comparisons between consecutively ordered nodes. In particular, for that confidence level, $\delta_n(u, v) \geq d_n[u] - d_n[v] - \Phi^{-1}(p_{conf})\sigma_n$, where Φ^{-1} is the inverse cumulative distribution function of the normal distribution $N(0, 1)$. Therefore, comparisons are *not* maintained on average as long as:

$$0 \geq \sum_{(u,v)\in\mathcal{P}_n} \big(r_\infty[u] - r_\infty[v]\big)$$
$$= \sum_{(u,v)\in\mathcal{P}_n} \big(p(r_n[u] - r_n[v]) + (1-p)(\delta_n[u] - \delta_n[u])\big)$$
$$\geq \sum_{(u,v)\in\mathcal{P}_n} \big(p(r_n[u] - r_n[v])$$
$$\qquad + (1-p)(r_n[u] - r_n[v]) - (1-p)\sigma\Phi^{-1}(p_{conf})\big)$$
$$= (\max r_n - \min r_n) - (1-p)\Phi^{-1}(p_{conf})|\mathcal{P}_n|\sigma$$

Based on this analysis, node comparisons are on average preserved by future iterations with confidence p_{conf} if we stop at the iteration n that first breaks through the above inequality.

245

In other words, if we calculate p through Equation 3, the power method finds a robust node order for the first time when:

$$1 + \frac{\sum_{k=1}^{n} \frac{a^k}{k}}{\ln(1-a)} < \frac{\max r_n - \min r_n}{\Phi^{-1}(p_{conf})\sigma_n|\mathcal{P}_n|} \quad (6)$$

Due to the sorting operation required to construct $\mathcal{P}_n$, which is in turn used to calculate σ_n, this criterion introduces an overhead of $O(|\mathcal{P}_n| \log |\mathcal{P}_n|)$ computations, where $|\mathcal{P}_n|$ is at most one less than the number of network nodes. This overhead would be important only when scoring networks whose number of nodes grows exponentially faster than their average degree.

As a final remark before closing this section, we point out that, although we have not fully removed the need of selecting a parameter (i.e. p_{conf}) that influences the stopping point, that parameter no longer depends on the characteristics of the scored network. Instead, it is a direct interpretation of the desired confidence towards node order robustness. Therefore, it suffices to investigate the efficacy of that parameter at most once to provide node scores of similar robustness in future applications. For example, throughout this work we select $\Phi^{-1}(p_{conf}) = 2$, which corresponds to $p_{conf} \approx 98\%$, and this can be used to achieve similar node order robustness in other networks without additional investigation.

V. EXPERIMENTS

To explore the added value of our analysis, we experiment on four large networks. Three of these are well-known public networks of the SNAP repository[2] for which ground truth communities are available [34], whereas the fourth one is gathered from Spotify's public API[3] and comprises music artists and links between related ones. In this network, artist genres are considered metadata communities [35].

For these networks, we select some of their known communities to serve as seed nodes and use a PPR scheme with various restart probabilities to score the relevance of the rest of network nodes to them. In Table I we present an overview of experiment communities, where their name prefix indicates the network they belong to and their name suffix their position within the list of that network's available communities:

Community		Network	
Name	Number of Seeds	Nodes	Edges
Amazon-1723	5,969	334,863	925,872
Amazon-3875	20,946	334,863	925,872
DBLP-13089	5,066	317,080	1,049,866
DBLP-13429	7,556	317,080	1,049,866
YouTube-176	2,217	1,134,890	2,987,624
YouTube-268	3,001	1,134,890	2,987,624
Spotify-3	218	733,043	5,883,675
Spotify-11	512	733,043	5,883,675

TABLE I
GROUND TRUTH COMMUNITIES WHOSE NODES WERE USED AS SEEDS.

In our experiments, we first calculate a strict estimation of ground truth scores by iterating the power method until

[2]http://snap.stanford.edu/data/
[3]https://developer.spotify.com/documentation/web-api/

it converges to 10^{-20} average difference between consecutive iterations. We remind that this tolerance level requires exceedingly many iterations to converge. We then compare the running time and node order between the ground truth scores and eight other convergence criteria; stopping at average score differences $10^{-E}, E \in \{6, \ldots, 12\}$, the network-agnostic proposal of stopping when 99% of random walks have been considered and stopping at the iteration when Equation 6 is first satisfied with 98% confidence. To measure the running time of different convergence criteria, we counted the number of iterations they take to stop, since each iteration is dominated by a multiplication with the same sparse matrix.

Node scores arising from different convergence criteria are compared with the ground truth ones with the Spearman correlation, which determines whether the ordinalities[4] of node scores are correlated. Correlations closer to 1 suggest similar node order. However, due to the large number of nodes involved, substantial qualitative changes are reflected at correlation differences of 0.001 or more. For example, in Figure 1 the scatterplot of calculated (vertical axes) vs. ground truth (horizontal axes) node ordinalities can be considered of much higher quality for the Spearman correlation of 0.99993 compared to 0.98960. As a general guideline, we consider node score ordinalities to be strongly correlated with their ground truth only if their Spearman correlation exceeds 0.999. However, we will later see that this threshold is not always accurate, as even stronger correlations are sometimes needed to identify a high quality node order. Nevertheless, it can be used to filter out significantly different score distributions.
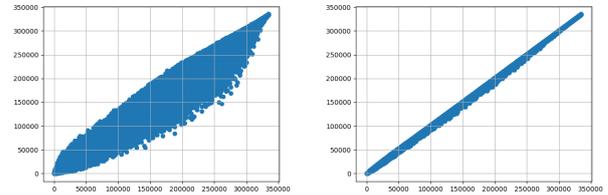
Fig. 1. Node ordinality scatterplots between the ground truth and 10^{-8} numerical tolerance (left), as well as between the ground truth and stopping at Equation 6 (right) for the seed community Amazon-1723 and $a = 0.99$. These achieve 0.98960 and 0.99993 Spearman correlation respectively.

Table II and Table III present the Spearman correlation with ground truth scores and the number of iterations for the PPR of communities across the described convergence criteria and PPR parameters $a \in \{0.85, 0.90, 0.95, 0.99, 0.995, 0.999\}$. Spearman correlation values sp are reported as $-\log_{10}(1-sp)$, which provides a finegrained understanding of the decimal place at which the correlation deviates from 1. For example, a reported value of 3 or greater corresponds to a correlation of $1 - 10^{-3} = 0.999$ or greater. Before discussing these results, we assert that ground truth scores achieve a robust near-exact ordering, as they maintain over $1 - 10^{-5} = 0.99999$ correlation

[4]The highest-scored node is assigned ordinality of 1 and lower-scored nodes assume higher integer ordinalities. Node score ordinalities are often referred to as ranks by recommender systems.

Community	a	Convergence Tolerance							Fraction of Walks	
		10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}	10^{-11}	10^{-12}	0.99	Eq. 6
Amazon-1723	.85	0.4	1.7	2.8	**4.2**	5.6	7.0	8.5	3.1	3.5
Amazon-1723	.90	0.4	1.7	2.9	**4.2**	5.6	7.1	8.6	3.1	3.6
Amazon-1723	.95	0.4	1.5	2.7	**4.1**	5.5	7.1	8.6	3.2	3.8
Amazon-1723	.99	0.5	1.0	1.9	**3.1**	4.6	6.3	8.1	3.4	4.1
Amazon-1723	.995	0.4	0.8	1.5	2.6	**4.1**	5.8	7.7	3.4	4.2
Amazon-1723	.999	0.2	0.3	0.7	1.5	**3.1**	4.8	6.4	4.2	5.3
Amazon-3875	.85	0.7	2.0	**3.1**	4.3	5.7	7.1	8.6	3.5	4.3
Amazon-3875	.90	0.7	2.0	**3.0**	4.3	5.7	7.2	8.7	3.5	4.3
Amazon-3875	.95	0.7	1.8	2.8	**4.1**	5.6	7.1	8.7	3.5	4.4
Amazon-3875	.99	0.7	1.1	1.9	**3.3**	4.7	6.3	8.0	3.4	4.3
Amazon-3875	.995	0.6	0.8	1.5	2.7	**4.1**	5.9	7.7	3.3	4.2
Amazon-3875	.999	0.2	0.4	0.7	1.5	**3.1**	4.9	6.4	4.1	5.1
DBLP-13089	.85	1.2	2.0	**3.4**	5.0	6.6	8.2	9.7	3.6	3.7
DBLP-13089	.90	1.0	1.8	**3.2**	4.7	6.4	7.9	9.4	3.5	3.7
DBLP-13089	.95	0.8	1.4	2.7	**4.2**	5.7	7.3	8.9	3.7	3.9
DBLP-13089	.99	0.4	0.8	1.9	**3.1**	4.4	5.7	7.2	4.8	5.3
DBLP-13089	.995	0.3	0.7	1.8	2.8	**3.9**	5.2	6.8	5.5	5.7
DBLP-13089	.999	0.2	0.6	1.7	2.6	**3.6**	4.7	6.0	9.3	10.5
DBLP-13429	.85	1.2	2.0	**3.4**	4.9	6.6	8.3	9.8	3.5	4.1
DBLP-13429	.90	1.0	1.8	**3.1**	4.8	6.4	8.0	9.6	3.5	4.3
DBLP-13429	.95	0.8	1.5	2.8	**4.3**	5.9	7.5	9.0	3.7	4.7
DBLP-13429	.99	0.4	0.9	2.1	**3.3**	4.5	5.8	7.2	4.8	5.1
DBLP-13429	.995	0.3	0.7	1.9	2.9	**3.9**	5.2	6.8	5.5	5.8
DBLP-13429	.999	0.2	0.6	1.6	2.6	**3.5**	4.6	5.8	8.6	9.6
YouTube-176	.85	0.6	2.1	**3.2**	4.5	6.0	7.5	8.5	3.9	5.5
YouTube-176	.90	0.5	1.8	2.9	**4.2**	5.6	6.9	8.4	3.8	5.9
YouTube-176	.95	0.5	1.4	2.4	**3.6**	5.0	6.4	7.6	4.2	6.5
YouTube-176	.99	0.3	1.1	1.9	2.7	**3.5**	4.5	6.0	5.1	7.6
YouTube-176	.995	0.3	1.1	1.9	2.5	**3.2**	4.0	5.2	5.4	8.2
YouTube-176	.999	0.5	1.0	1.8	2.0	2.8	**4.0**	5.8	7.9	13.0
YouTube-268	.85	0.5	2.1	**3.3**	4.7	6.4	7.8	9.1	4.0	5.9
YouTube-268	.90	0.5	1.9	**3.0**	4.4	6.1	7.6	8.9	4.1	6.2
YouTube-268	.95	0.7	1.6	2.6	**4.0**	5.5	7.1	8.5	4.4	6.7
YouTube-268	.99	0.5	1.2	2.1	**3.1**	4.2	5.4	6.9	5.4	7.7
YouTube-268	.995	0.5	1.2	2.0	2.7	**3.6**	4.5	5.9	5.5	8.4
YouTube-268	.999	0.5	1.1	1.7	2.0	**3.3**	4.2	5.9	8.1	15.0
Spotify-3	.85	0.0	0.9	2.1	**3.5**	5.0	6.6	8.4	2.6	3.0
Spotify-3	.9	0.0	0.8	2.2	**3.5**	5.1	6.8	8.5	2.7	3.3
Spotify-3	.95	0.0	0.9	2.1	**3.6**	5.2	6.9	8.5	3.0	3.7
Spotify-3	.99	0.0	0.7	1.9	**3.3**	4.7	6.0	7.1	4.4	5.1
Spotify-3	.995	0.0	0.6	1.7	**3.0**	4.3	5.4	6.5	5.3	5.9
Spotify-3	.999	0.0	0.5	1.4	2.4	**3.3**	3.9	5.2	6.5	7.8
Spotify-11	.85	0.2	1.5	2.4	**3.7**	5.3	7.0	8.7	2.9	3.3
Spotify-11	.9	0.2	1.2	2.3	**3.7**	5.4	7.0	8.7	3.0	3.4
Spotify-11	.95	0.2	1.1	2.2	**3.6**	5.3	6.9	8.6	3.3	3.8
Spotify-11	.99	0.1	0.7	1.8	**3.1**	4.4	5.6	6.6	4.4	5.0
Spotify-11	.995	0.1	0.7	1.6	2.9	**4.2**	5.2	6.3	5.1	5.8
Spotify-11	.999	0.1	0.5	1.3	2.3	**3.3**	3.9	5.3	6.6	8.0

TABLE II

$-\log_{10}$ OF HOW MUCH SPEARMAN CORRELATION WITH GROUND TRUTH SCORES DEVIATES FROM 1.
BOLD VALUES INDICATE THE FIRST TIME CORRELATION EXCEEDS 0.999 .

with the ordering of the much laxer (although strict by itself) 10^{-12} mean absolute error tolerance.

In the reported experiments, even the simplistic stopping criterion that considers up to 99% of random walks achieves a strong (i.e. at least 0.999) correlation with ground truth scores in 45 out of the 48 community and parameter value combinations. Furthermore, the proposed stopping point of Equation 6 boasts a strong correlation in every experiment. Both of these approaches stop a lot earlier than the "safe" approach of stopping at $1-10^{-20}$ numerical tolerance. Experiment results also show that the numerical tolerance needed to achieve equally strong correlation varies between parameters and networks; this corroborates the need for time-consuming investigation over various potential tolerances when our approach is not applied.

For the larger restart probabilities corresponding to parameters $a \in \{0.85, 0.90, 0.95\}$, both methods aiming to provide a robust node order exhibit strong correlations with the ground truth scores. Furthermore, they converge within a number of iterations that is comparable to the laxest found numerical convergence tolerance of those also strongly correlated with the ground truth scores. In other words, for these restart probabilities our approach can be considered a good automated process for finding robust scores that also require few iterations to compute.

When moving to low (i.e. 5% or lower) restart probabilities, our two proposed approaches require a significantly higher number of iterations to converge than the frequently suggested 10^{-9} and 10^{-10} numerical tolerances. However, after a more detailed investigation of score scatterplots, a representative example of which is given in Table 2, we find that the criterion of 0.999 correlation fails to maintain a robust node order, despite being adequate for lower restart probabilities. In those cases, it matters not how fast the empirically selected numerical tolerances converge, since their found node order is

247

Community	a	Convergence Tolerance								Fraction of Walks	
		10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}	10^{-11}	10^{-12}	10^{-20}	0.99	Eq. 6
Amazon-1723	.85	4	8	16	**28**	40	52	65	163	18	21
Amazon-1723	.90	4	9	22	**39**	57	75	94	238	25	32
Amazon-1723	.95	4	11	35	**67**	101	137	174	439	47	61
Amazon-1723	.99	5	15	81	**184**	323	489	672	1831	204	252
Amazon-1723	.995	5	16	100	257	**509**	825	1167	3196	388	480
Amazon-1723	.999	5	17	128	444	**1195**	2111	3072	16447	1754	2596
Amazon-3875	.85	4	8	**15**	25	38	50	63	170	18	25
Amazon-3875	.90	4	9	**19**	36	54	73	93	256	25	36
Amazon-3875	.95	4	11	30	**62**	98	136	175	504	47	69
Amazon-3875	.99	5	13	71	**190**	335	503	686	2265	204	293
Amazon-3875	.995	5	13	91	274	**531**	848	1192	4065	388	553
Amazon-3875	.999	5	14	123	486	**1256**	2168	3124	16451	1754	2278
DBLP-13089	.85	4	8	**17**	27	37	48	59	156	18	19
DBLP-13089	.90	4	10	**22**	36	51	66	82	228	25	27
DBLP-13089	.95	4	12	31	**54**	80	108	138	430	47	50
DBLP-13089	.99	4	17	52	**102**	173	266	386	1628	204	237
DBLP-13089	.995	4	18	57	118	**212**	357	552	2588	388	414
DBLP-13089	.999	4	19	62	137	**273**	539	912	5190	1754	2126
DBLP-13429	.85	4	8	**17**	27	38	49	61	157	18	22
DBLP-13429	.90	4	10	**22**	37	52	68	85	229	25	33
DBLP-13429	.95	4	13	32	**57**	83	112	142	431	47	63
DBLP-13429	.99	4	17	56	**110**	179	271	389	1635	204	222
DBLP-13429	.995	4	18	63	128	**219**	363	556	2606	388	429
DBLP-13429	.999	4	19	69	148	**282**	545	922	5251	1754	2085
YouTube-176	.85	3	8	**14**	22	31	41	51	143	18	28
YouTube-176	.90	3	9	17	**28**	40	54	69	208	25	43
YouTube-176	.95	3	10	22	**39**	60	84	110	393	47	86
YouTube-176	.99	3	12	30	62	**107**	168	267	1537	204	393
YouTube-176	.995	3	13	32	68	**121**	208	368	2549	388	762
YouTube-176	.999	4	13	33	74	136	**277**	576	5646	1754	3583
YouTube-268	.85	3	8	**14**	22	32	42	53	148	18	29
YouTube-268	.90	3	9	**17**	28	42	57	73	217	25	43
YouTube-268	.95	4	11	23	**41**	64	91	119	407	47	84
YouTube-268	.99	4	13	32	**70**	127	208	317	1552	204	392
YouTube-268	.995	4	13	34	78	**150**	270	433	2562	388	761
YouTube-268	.999	4	13	36	87	181	**371**	656	5673	1754	3584
Spotify-3	.85	3	6	14	**25**	37	49	62	163	18	21
Spotify-3	.9	3	6	19	**35**	52	70	89	238	25	32
Spotify-3	.95	3	8	29	**58**	90	122	155	439	47	61
Spotify-3	.99	3	10	60	**136**	220	314	420	1831	204	252
Spotify-3	.995	3	10	71	**165**	276	407	582	3196	388	480
Spotify-3	.999	3	10	83	201	**353**	585	1179	7958	1754	4967
Spotify-11	.85	3	7	14	**24**	35	47	59	160	18	21
Spotify-11	.9	3	7	18	**33**	50	67	85	236	25	30
Spotify-11	.95	3	9	27	**53**	84	116	149	439	47	56
Spotify-11	.99	3	11	49	**120**	208	307	418	1838	204	256
Spotify-11	.995	3	12	56	147	**264**	404	584	3209	388	502
Spotify-11	.999	3	12	65	181	**345**	590	1207	7988	1754	2364

TABLE III

NUMBER OF ITERATIONS (CLOCK TIME PER ITERATION IS CONSTANT IN EACH SETUP).
BOLD VALUES CORRESPOND TO THE FIRST TIME SPEARMAN CORRELATION EXCEEDS 0.999.
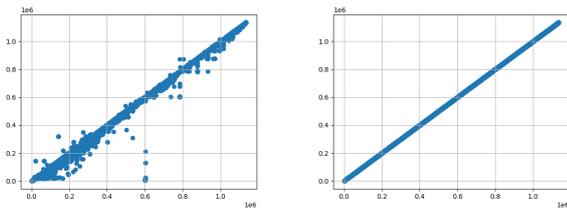
Fig. 2. Node ordinality scatterplots between the ground truth scores and converging at 10^{-11} tolerance (left) and Equation 6 (right) for the seed community YouTube-268 and $a = 0.999$.

not truly robust, despite their summary into Spearman correlation suggesting otherwise. This phenomenon demonstrates the usefulness of our approach in avoiding pitfalls a surface-level empirical investigation could fall into.

Overall, our experiments show that the number of power method iterations we propose is able to discover a robust node ordering in every case without needing to perform a human-driven exploration up to stricter convergence tolerances. For example, in Figure 3 we can see that even the worst node ordering obtained by Equation 6 closely follows the ground truth. Furthermore, although low restart probabilities can induce many iterations under our criteria, these are to a large degree justified if we want to also maintain a robust node ordering. At worst, our approach provides a high quality upper bound of when to stop iterating.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we explored node order convergence for the power method of calculating personalized PageRank. In particular, we analysed the fraction of random walks considered during its iterations and used that analysis to suggest a stopping point that reaches a robust node score order with high confidence. We experimented on four real-world networks, in

Authorized licensed use limited to: Centre for Research and Technology (C.E.R.T.H.). Downloaded on September 01,2022 at 12:35:23 UTC from IEEE Xplore. Restrictions apply.
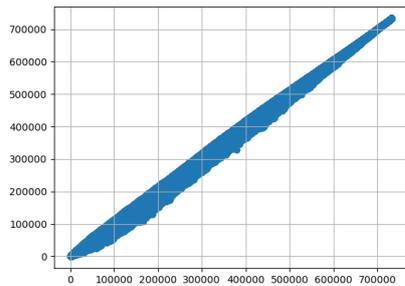
Fig. 3. Weakest-correlated ordinality scatterplot between ground truth scores and stopping at Equation 6, encountered for the seed community Spotify-3 and $a = 0.85$.

which we demonstrated the effectiveness of our approach and that it avoids the time-consuming (and sometimes misleading if not performed in detail) empirical investigation into which convergence tolerances to consider robust.

In future work, we are interested in conducting more detailed experiments, as well as adapting our stopping criteria for use alongside other optimizations of the power method.

REFERENCES

[1] D. L. Banks and N. Hengartner, "Social networks," *Encyclopedia of Quantitative Risk Analysis and Assessment*, vol. 4, 2008.

[2] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics Reports*, vol. 659, pp. 1–44, 2016.

[3] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th international conference on World wide web*, pp. 631–640, ACM, 2010.

[4] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *Acm computing surveys (csur)*, vol. 45, no. 4, p. 43, 2013.

[5] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.

[6] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Sixth International Conference on Data Mining (ICDM'06)*, pp. 613–622, IEEE, 2006.

[7] T. Haveliwala, S. Kamvar, and G. Jeh, "An analytical comparison of approaches to personalizing pagerank," tech. rep., Stanford, 2003.

[8] D. Fogaras, B. Rácz, K. Csalogány, and T. Sarlós, "Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments," *Internet Mathematics*, vol. 2, no. 3, pp. 333–358, 2005.

[9] B. Bahmani, A. Chowdhury, and A. Goel, "Fast incremental and personalized pagerank," *Proceedings of the VLDB Endowment*, vol. 4, no. 3, pp. 173–184, 2010.

[10] C. Borgs, J. Chayes, M. Mahdian, and A. Saberi, "Exploring the community structure of newsgroups," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 783–787, ACM, 2004.

[11] D. Gleich and K. Kloster, "Seeded pagerank solution paths," *European Journal of Applied Mathematics*, vol. 27, no. 6, pp. 812–845, 2016.

[12] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," 2018.

[13] J. Klicpera, A. Bojchevski, and S. Günnemann, "Personalized embedding propagation: Combining neural networks on graphs with personalized pagerank," *arXiv preprint arXiv:1810.05997*, 2018.

[14] A. N. Langville and C. D. Meyer, "A survey of eigenvector methods for web information retrieval," *SIAM review*, vol. 47, no. 1, pp. 135–161, 2005.

[15] A. N. Langville and C. D. Meyer, "Deeper inside pagerank," *Internet Mathematics*, vol. 1, no. 3, pp. 335–380, 2004.

[16] Y. Fujiwara, M. Nakatsuji, H. Shiokawa, T. Mishima, and M. Onizuka, "Fast ad-hoc search algorithm for personalized pagerank," *IEICE TRANSACTIONS on Information and Systems*, vol. 100, no. 4, pp. 610–620, 2017.

[17] C. Cui, M. Hu, J. D. Weir, and T. Wu, "A recommendation system for meta-modeling: A meta-learning based approach," *Expert Systems with Applications*, vol. 46, pp. 33–44, 2016.

[18] D. Gleich, L. Zhukov, and P. Berkhin, "Fast parallel pagerank: A linear system approach," *Yahoo! Research Technical Report YRL-2004-038, available via http://research. yahoo. com/publication/YRL-2004-038. pdf*, vol. 13, p. 22, 2004.

[19] A. Arasu, J. Novak, A. Tomkins, and J. Tomlin, "Pagerank computation and the structure of the web: Experiments and algorithms," in *Proceedings of the Eleventh International World Wide Web Conference, Poster Track*, pp. 107–117, 2002.

[20] M. Bianchini, M. Gori, and F. Scarselli, "Inside pagerank," *ACM Transactions on Internet Technology (TOIT)*, vol. 5, no. 1, pp. 92–128, 2005.

[21] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, "Extrapolation methods for accelerating pagerank computations," in *Proceedings of the 12th international conference on World Wide Web*, pp. 261–270, ACM, 2003.

[22] X. Tan, "A new extrapolation method for pagerank computations," *Journal of Computational and Applied Mathematics*, vol. 313, pp. 383–392, 2017.

[23] G. H. Golub and C. Greif, "An arnoldi-type algorithm for computing page rank," *BIT Numerical Mathematics*, vol. 46, no. 4, pp. 759–771, 2006.

[24] K. He, P. Shi, D. Bindel, and J. E. Hopcroft, "Krylov subspace approximation for local community detection in large networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 5, p. 52, 2019.

[25] F. McSherry, "A uniform approach to accelerated pagerank computation," in *Proceedings of the 14th international conference on World Wide Web*, pp. 575–582, ACM, 2005.

[26] Q. Liu, Z. Li, J. Lui, and J. Cheng, "Powerwalk: Scalable personalized pagerank via random walks with vertex-centric decomposition," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 195–204, ACM, 2016.

[27] P. Lofgren, S. Banerjee, and A. Goel, "Personalized pagerank estimation and search: A bidirectional approach," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 163–172, ACM, 2016.

[28] S. Wang, Y. Tang, X. Xiao, Y. Yang, and Z. Li, "Hubppr: effective indexing for approximate personalized pagerank," *Proceedings of the VLDB Endowment*, vol. 10, no. 3, pp. 205–216, 2016.

[29] S. Wang, R. Yang, X. Xiao, Z. Wei, and Y. Yang, "Fora: Simple and effective approximate single-source personalized pagerank," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 505–514, ACM, 2017.

[30] K. Avrachenkov, N. Litvak, D. Nemirovsky, E. Smirnova, and M. Sokol, "Quick detection of top-k personalized pagerank lists," in *International Workshop on Algorithms and Models for the Web-Graph*, pp. 50–61, Springer, 2011.

[31] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using pagerank vectors," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 475–486, IEEE, 2006.

[32] K. Avrachenkov, A. Kadavankandy, and N. Litvak, "Mean field analysis of personalized pagerank with implications for local graph clustering," *Journal of statistical physics*, vol. 173, no. 3-4, pp. 895–916, 2018.

[33] E. Krasanakis, E. Schinas, S. Papadopoulos, Y. Kompatsiaris, and A. Symeonidis, "Boosted seed oversampling for local community ranking," *Information Processing & Management*, p. 102053, 2019.

[34] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, 2015.

[35] T. Lidy, A. Lecoutre, K. Boulkenafet, M. Schinas, C. Koutlis, S. Papadopoulos, and P. Charitidis, "D3.2–predictive analytics and recommendation framework v2," 2019.