“ Can you really trust that seed?” : Reducing the Impact of Seed Noise in Personalized PageRank

Shengyu Huang* Xinsheng Li*
Arizona State University
Tempe, AZ 85287, USA
Email: shengyu.huang@asu.edu, lxinshen@asu.edu

K. Selçuk Candan
Arizona State University
Tempe, AZ 85287, USA
Email: candan@asu.edu

Maria Luisa Sapino
University of Torino
I-10149 Torino, Italy
Email: marialuisa.sapino@unito.it

***Abstract*—Network based recommendation systems leverage the topology of the underlying graph and the current user context to rank objects in the database. Random-walk based techniques, such as PageRank, encode the structure of the graph in the form of a transition matrix of a stochastic process from which the significances of the nodes in the graph are inferred. Personalized PageRank (PPR) techniques complement this with a seed node set which serves as the *personalization context*. In this paper, we note (and experimentally show) that PPR algorithms that do not differentiate among the seed nodes may not properly rank nodes in situations where the seed set is *incomplete* and/or *noisy*. To tackle this problem, we propose alternative *robust personalized PageRank* (RPR) strategies, which are insensitive to noise in the set of seed nodes and in which the rankings are not overly biased towards the seed nodes. In particular, we show that novel *teleportation discounting* and *seed-set maximal PPR* techniques help eliminate harmful bias of individual seed nodes and provide effective seed differentiation to lead to more accurate rankings.**

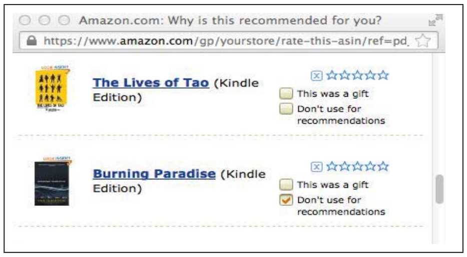
I. Introduction

How a given pair of nodes in a social network are related to each other reflects the underlying network topology. Recommendation systems often use the analysis of the structure of a given data and/or social graph, relative to the user's current context, to generate rankings and recommendations [16].

Significance of a node in a graph needs to reflect both the topology of the graph and the application semantics and measures: The *betweenness* measure [24] for example aims to quantify whether deleting the node would disconnect or disrupt the graph. The *centrality/cohesion* [6] measures quantify how close to a clique the given node and its neighbors are. Other *authority*, *prestige*, and *prominence* measures [5]–[7] measure the significance of the node in the graph through eigen-analysis or random walks. For example, the well-known PageRank algorithm [7] associates an importance score to each node relying on random walks: Let us consider a weighted, directed graph $G(V, E)$, where the weight of the edge $e_j \in E$ is denoted as $w_j (\geq 0)$ and $\sum_{e_j \in outedge(v_i)} w_j = 1.0$. The PageRank score of nodes V is the stationary distribution of a random walk on G, denoted with a vector $\vec{p}$:

$$\vec{p} = (1 - \beta)\mathbf{T}_G \times \vec{p} + \beta\vec{v}, \qquad (1)$$

where $\mathbf{T}_G$ denotes the transition matrix corresponding to the graph G (and the underlying edge weights) and $\vec{v}$ is a so-called *teleportation* vector, where all entries are $\frac{1}{\|V\|}$.



Amazon.com: Why is this recommended for you?
https://www.amazon.com/gp/yourstore/rate-this-asin/ref=pd

The Lives of Tao (Kindle Edition)
☒ ☆☆☆☆☆
□ This was a gift
□ Don't use for recommendations

Burning Paradise (Kindle Edition)
☒ ☆☆☆☆☆
□ This was a gift
☑ Don't use for recommendations

Fig. 1. An example interface enabling the user to explicitely eliminate outliers (e.g., a book purchased as a gift to a friend) in generating recommendations; such explicit corrections is not feasible in all applications of social networks

An early attempt to contextualize the PageRank scores was the *topic sensitive PageRank* [18] approach which adjusts the PageRank scores of the nodes by assigning the *teleportation* probabilities in vector $\vec{j}$ in a way that reflects the graph nodes' degrees of match to a given search topic. In many applications, however, the context relevant to the recommendation is defined not through an explicit query, but a subset of the nodes (often referred to as the “*seed nodes*”) in the graph. Personalized PageRank (PPR) [5], [11], for example, takes into account a user's interest by modifying the teleportation vector taking into account a given set of *important* nodes which are the target of the random jumps: given a set of nodes $S \subseteq V$, instead of jumping to a random node in V with probability β, the random walk jumps to one of the nodes in the seed set, S, given by the user. More specifically, if we denote the *Personalized PageRank* (PPR) scores of the nodes in V with a vector $\vec{\pi}$, then

$$\vec{\pi} = (1 - \beta)\mathbf{T}_G \times \vec{\pi} + \beta\vec{s}, \qquad (2)$$

where $\vec{s}$ is a re-seeding vector, such that if $v_i \in S$, then $\vec{s}[i] = \frac{1}{\|S\|}$ and $\vec{s}[i] = 0$, otherwise.

A. Problem - Noisy Seed Sets

The above formulation of PPR *assumes that all seeds are equally important* in characterizing the user's interest. This, however, may not always be the case, since in practice, the user feedback is often incomplete and noisy [8] (Figure 1). Unfortunately unless (a) each *individual* seed node is a good representative for the entire seed set, (b) the user/system was successful in including all seed nodes relevant for defining the current user context, and (c) most importantly, the user/system did not include any outlier nodes in the seed set, the resulting rankings might be biased and might contain undesirable artifacts, such as movies with low ratings, having high PPR rankings. Consider the following example:

“*” indicates student authors (with equal contributions).This work is funded by the NSF Grant ♯ 1016921,“One Size Does Not Fit All: Empowering the User with User-Driven Integration”.

ASONAM 2014, August 17-20, 2014, Beijing, China
978-1-4799-5877-1/14/$31.00 ©2014 IEEE

Rank Statistics of 9 Noisy Seeds (out of 49 Seeds in 2500 Movies)		
Best Rank	*Avg. Rank*	*Worst Rank*
18	42.9	52

TABLE I. BIAS AND LACK OF SEED DIFFERENTIATION IN PPR SCORES: MOST OF THE 49 SEEDS (OUT OF 2500 MOVIES) HAVE VERY HIGH PPR RANKS, EVEN IF THEY ARE OUTLIERS IN THE SEED SET

Example 1: (Impact of the Noise in the Seed Set) Table I shows the PPR scores and ranks of the noisy seed nodes in a movie graph (see Section IV for more details about this data set) for a sample user. In this example, we construct an *imperfect seed set* as follows: we select a random user and include in the seed set 40 movies rated "↑" (for "like") and 9 movies rated "↓" (for "dislike") by this user (out of 81 movies rated ↑and 25 movies rated ↓ by this user).

Table I studies the relationship between the original user rating and the PPR ranking. As we see here, while as expected the average PPR rankings of the 9 noisy seed movies is poor among the rankings of the seed movies ($\sim$ 43 out of 52), it is also true that all the seed movies (including the 9 ↓-rated outlier movies) rank highly (i.e., better than 53 out of 2500).

Note that, while we often do not care about the ranks of the seed movies, high PPR-ranking of ↓-rated seeds implies that those movies neighboring these noisy seeds are also likely to be ranked highly.

B. Our Contributions: Robust Personalized PageRank (RPR)

Intuitively, a *noisy seed* is a seed node (provided by the user or selected by the system) which does not properly reflect the user's focus in that it does to fit in the context defined by the seed set as a whole. The above example illustrates that *a poor seed may overestimate the rankings of its (equally poor) neighbors in the final ranking*. This is primarily due to *(a) Teleportation-bias:* Firstly, as discussed in Section I, the teleportation-vector based seeding algorithms jump on the seed set, S, relatively often (the common transportation probability, β, is 0.15) and the size of the seed set is often much smaller relative to the size of the data graph: as a result, the PPR value of the least significant seed node will be at least $\frac{\beta}{|S|}$, which is likely to be much higher than the PPR of non-seed nodes in the graph. *(b) Need for seed differentiation:* Secondly, as we commented above, the negative impact of the teleportation-bias can be alleviated if the teleportation rates to the seeds can be *differentiated* identify which seeds are truly better than others as this information is not available *a priori*[1]:

In this paper, we propose *techniques to eliminate teleportation-bias and/or provide seed differentiation*:

- First and foremost, we discuss how the node rankings can be negatively affected by possible incompleteness and/or imperfection in the seeds set, and we experimentally establish that the conventional PPR metrics might not properly differentiate seed nodes in a graph.
- Secondly, we propose alternative *Robust personalized PageRank* (RPR) strategies, (a) which are insensitive to noise in the set of seed nodes (and thus differentiate seeds well) and (b) in which the rankings are not overly biased towards the seed nodes (Table II).

Rank Statistics of 9 Noisy Seeds (out of 49 Seeds in 2500 Movies)		
Best Rank	*Avg. Rank*	*Worst Rank*
94	1109.8	1568

TABLE II. SEED DIFFERENTIATION AND BIAS ELIMINATION THROUGH *robust personalized PageRank* (RPR) SCORES: IN THE SAME SITUATION AS IN TABLE I, THE AVERAGE RATING OF THE NOISY SEEDS IS GREATER THAN 1000 OUT OF 2500 MOVIES

In the next section, we discuss the related work. In Section III, we first formally introduce the problem and then present our solutions for seed differentiation, seed-bias elimination, and *Robust personalized PageRank* (RPR) computation. In Section III, we discuss optimization and parallelization opportunities. We evaluate RPR for different data sets and under different scenarios in Section IV and conclude in Section V.

II. Related Works

[9] and [10] were among the first works which recognized that random-walks can also be used for measuring the *significance* of the graph nodes relative to a given *seed node set*, $S \subseteq V$: More specifically, in [9] the authors proposed to construct a transition matrix, $\mathbf{T}_S$, where edges leading away from the seed nodes are weighted less than those edges leading towards the seed nodes. A second approach to contextualizing PageRank scores is to use the PPR techniques [5], [11] discussed in Section I. One key advantage of this teleportation vector modification based approach over modifying the transition matrix, as in [9], is that the term β (in Equation 2) can be used to directly control the *degree of seeding (or personalization)* of the PPR score. In fact, these personalized random-walk and PageRank based measures of node significance have been shown to be highly effective in many prediction and recommendation applications [1], [19]. An alternative, *hitting time* based approach, where the hitting time is defined as the expected number of steps a random walk from the source vertex to the destination vertex will take, was also considered in the literature [12], [14], [21], [22].

Recent advances on PPR computation include top-k and approximate personalized PageRank algorithms [2], [4], [11], [13], [15], [17], [23] and parallelized implementations on MapReduce or Pregel based systems [3], [20].

III. Robust Personalized PageRank

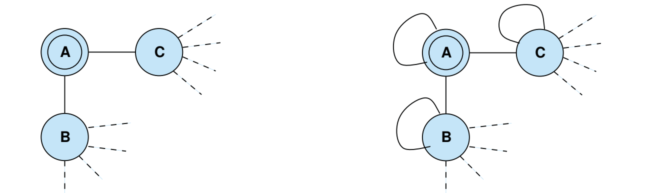
Addressing the problem of noisy seeds through non-uniform teleportation to the seed nodes requires a mechanism to distinguish among the nodes in the seed set, S.

A. PPR-G: PPR with Global Seed Ranking

A *first (and as we see later, mostly ineffective) attempt* to differentiate among the seeds might be to consider the global properties of the nodes in the seed set. One relatively straightforward way to achieve this is to first measure the significance of the individual seed nodes in the overall graph, G, (using for example PageRank) and, then, modulate the teleportation rates onto the seed nodes based on the relative significance values of the seeds. In other words, we would compute the *PPR scores with global seed ranking* (also referred to as PPR-G scores) as follows: Given a graph, $G(V, E)$, and a seed node set S, let $\vec{p}$ be the PageRank scores computed by solving Equation 1. Then, the PPR-G scores, $\vec{g}$, are obtained by solving

$$\vec{g} = (1 - \beta)\mathbf{T}_G\, \vec{g} + \beta \vec{s_2},$$

[1] Otherwise the seed set would have been constructed differently



(a) A is under-accounted relative to B and C if we discount teleportations

(b) A is not under-accounted when teleportation scores are discounted

Fig. 2. (a) Discounting teleportation scores would cause under-accounting of A relative to B and C. (b) If we add self-loops to nodes, when A is selected as a re-start point, A will not be under-accounted relative to its neighbors

where $\vec{s_2}$ is a re-seeding vector, such that for each $v_i \notin S$, $\vec{s_2}[i] = 0$, and for each $v_i \in S$, $\vec{s_2}[i] = \frac{\vec{p}[i]}{\sum_{v_j \in S} \vec{p}[j]}$.

As we later see in Section IV, however, in many cases, simply modifying the teleportation rates of the seed nodes based on the global significances of the nodes in the seed set does not properly eliminate seed-bias.

B. RPR-1: Teleportation-Discounted PPR

In the PPR formulation, the seed and non-seed nodes have different contributors to their final scores. *For the non-seed nodes*, the only contributor to their PPR scores is the number of times they are visited during the regular random-walk process. On the other hand, *for the seed nodes*, both (a) the number of times they are visited during the regular random-walk process and (b) the number of times they are selected as a teleportation destination for random-walk restart contribute to the PPR scores; we refer to these as the *random-walk contribution* (rw-PPR) and *teleportation contribution* (t-PPR), respectively. As described above, for non-seed nodes, the value of t-PPR score is 0.0.

Our first observation is that the t-PPR score of a seed node is exactly $\beta/|S|$ for each seed and, thus, a seed node will have at least $\beta/|S|$ overall PPR score, even if it is an outlier in the overall context defined by the seed set, S. Therefore, the first proposal to increase robustness of the the PPR scores against noise in the the seed set S is to discount the teleportation contributions from the PPR scores.

1) Teleportation-Discounting: Based on this observation, we define *teleportation-discounted PPR* scores (also referred to as RPR-1 scores) as follows: Given a graph, G, and a seed node set S, let $\vec{\pi}$ be the PPR scores as defined earlier:

- for each $v_i \notin S$, the corresponding RPR-1 score, $\vec{\rho_1}[i]$, is defined as $\vec{\rho_1}[i] = \frac{\vec{\pi}[i]}{1-\beta}$;
- in contrast, for each $v_i \in S$, the corresponding RPR-1 score, $\vec{\rho_1}[i]$, is defined as $\vec{\rho_1}[i] = \frac{\vec{\pi}[i] - \frac{\beta}{|S|}}{1-\beta}$.

PPR-1 scores as defined above do not alter the relative ordering of the non-seed nodes; instead (as discussed above) they aim to allow us to discover the significance of the seed nodes themselves relative to the non-seed nodes within the overall context defined by the seed set, S.

2) Preventing Under-Accounting of Seeds: One potential problem with the teleportation-discounting is that this time the seed nodes may in fact be *under-accounted*: neighbors of a seed node may get higher amounts of random-walk traffic (i.e, rw-PPR) than the seed node itself, simply because once you teleport to a seed node, you need to visit one of its neighbors but not vice versa. In order to prevent this *under-accounting*, we modify the input graph G by inserting a self-loop to each node in the network (Figure 2). In the resulting graph G', every node is a neighbor of itself and thus a seed node, v, will not get a lesser amount of random-walk traffic than its neighbors when v is selected as a re-start point.

C. RPR-2: Seed-Set Maximal PPR

RPR-1 reduces the teleportation bias, but the seed nodes are still being teleported to with the same, undifferentiated rate.

1) Seed-Set Maximality Principle and RPR-2: An alternative to fixing the teleportation significance of the individual seed nodes *a priori*, without considering the context provided by the other seed nodes (as in the PPR-G scheme we discussed in Section III-A), is to discover the teleportation significance of the individual seed nodes, relying on a novel *seed-set maximality* principle that would tie the teleportation rates of the seeds to their contributions to the overall personalized PageRank score of the seed set.

Principle 1 (Seed-Set Maximal PPR Scores): Let $G(V, E)$ be a graph and let $S \subseteq V$ be a set of seed nodes. Given an overall teleportation rate β, the re-seeding/re-start vector, $\vec{s}$, should be selected such that the overall PageRank scores of the nodes in S will be maximal. ◇

Intuitively, this principle requires that the seed set as a whole must score highly, but does not require that the individual seed nodes themselves have high scores. Based on this principle, the *seed-set maximal PPR scores* (also referred to as the RPR-2 scores) are computed as follows: Given a graph $G(V, E)$, a teleportation probability, β, and a seed set, S, the re-start vector $\vec{s}$ should be such that

$$\vec{\rho_2} = (1-\beta)\mathbf{T}_G \, \vec{\rho_2} + \beta\vec{s},$$

$$\sum_{v_i \in V} \vec{s}[i] = 1, \forall_{v_i \in V} 0 \leq \vec{s}[i] \leq 1, \sum_{v_i \in V} \vec{\rho_2}[i] = 1, 0 \leq \vec{\rho_2}[i] \leq 1,$$

and the following term is *maximized*:

$$seed\_set\_significance = \sum_{v_i \in S} \vec{\rho_2}[i].$$

Naturally, one possible concern with this new formulation is that it might potentially increase the computation cost. We next show that RPR-2 scores can be cheaply computed.

2) Seed only Re-Starts for RPR-2: According to the above formulation of seed-set maximal PPR scores, the seed nodes in S are not necessarily the only targets for re-starts. However, we can show that, for any traversal that re-starts at a non-seed node, there is a traversal that starts only at the seed nodes, but has a higher seed node traversal rate.

Proof 1: Given a graph $G(V, E)$, let $\vec{s}$ be an optimal re-start vector, such that $\exists v_i \notin S$ such that $\vec{s}[i] > 0$. Now consider the traversals that start only from a $v_i \notin S$ and let $\vec{\alpha_i}$ be a vector describing the average portion of time the random walk spends on the graph nodes in V before the next teleportation. We can then define two quantities (that add up to 1.0):

$$seed\_ratio_i = \sum_{v_j \in S} \vec{\alpha_i}[j], \text{ and } non\_seed\_ratio_i = \sum_{v_j \notin S} \vec{\alpha_i}[j].$$

Note that, since the traversal starts at a non-seed node, we have $non\_seed\_ratio_i > 0$; moreover, we can split this term into two, $non\_seed\_ratio_{i,before}$ and $non\_seed\_ratio_{i,after}$; i.e., the amount of time spent on non-seed nodes before and after a seed node is met during the random-walk, respectively. Let us also define a vector, $\vec{f_i}$, where for a given $v_j \in S$, the value of $\vec{f_i}[j]$ is the likelihood of v_j being the first seed met during the random-walk starting at node v_i.

Now consider an alternative re-start vector $\vec{\sigma}$ such that (a) $\vec{\sigma}[i] = 0$, (b) $\forall_{v_j \notin S}$, if $j \neq i$, then $\vec{\sigma}[j] = \vec{s}[j]$, and (c) $\forall_{v_j \in S}$, $\vec{\sigma}[j] = \vec{s}[j] + \vec{s}[i]\ \vec{f_i}[j]$. It is easy to see that the random-walks resulting when using the restart vector $\vec{\sigma}$ are similar to the random-walks resulting when using $\vec{s}$, except that the value of $non\_seed\_ratio_{i,before}$ is equal to 0. This means some of this time will be spent on the seed nodes contradicting the initial premise that $\vec{s}$ was an optimal re-start vector, maximizing the total amount of time spent on seed nodes. □

Based on the above proof, we can reformulate the equation set for seed-set maximal PPR scores in a way that limits the transportation targets, as follows:

$$\vec{\rho_2} = (1-\beta)\mathbf{T}_G\ \vec{\rho_2} + \beta\vec{s}, \quad \sum_{v_i \in V} \vec{\rho_2}[i] = 1, \quad 0 \le \vec{\rho_2}[i] \le 1,$$

$$\sum_{v_i \in S} \vec{s}[i] = 1, \quad \forall_{v_i \in s} 0 \le \vec{s}[i] \le 1, \quad \forall_{v_i \notin s} \vec{s}[i] = 0,$$

and $\quad seed\_set\_significance = \sum_{v_i \in S} \vec{\rho_2}[i] \quad$ is *maximum.*

3) Constraining the Size of the Re-Start Set: We can further show that in practice the restart set, S_{crit}, is a singleton; i.e., with very high likelihood, $|S_{crit}| = 1$.

Proof 2: Given a graph $G(V, E)$, let $\vec{s}$ be an optimal re-start vector, such that $\exists v_i, v_j \in S$ such that $\vec{s}[i] > 0$ and $\vec{s}[j] > 0$. Now consider all the traversals that start only from v_i and let $\vec{\alpha_i}$ be a vector describing the average portion of time the random walk that starts at v_i spends on the nodes in V before the next teleportation. Similarly, let $\vec{\alpha_j}$ be a vector describing the average portion of time a random walk that starts at v_j spends on the nodes in V before the next teleportation. Given $\vec{\alpha_i}$ and $\vec{\alpha_j}$, let us define two quantities,

$$seed\_ratio_i = \sum_{v_k \in S} \vec{\alpha_i}[k] \ \text{ and } \ seed\_ratio_j = \sum_{v_k \in S} \vec{\alpha_j}[k],$$

and let us assume that $seed\_ratio_i > seed\_ratio_j$ (a similar argument holds when $seed\_ratio_j > seed\_ratio_i$).

Now consider an alternative re-start vector $\vec{\sigma}$ such that (a) $\forall_{v_k \notin \{v_i, v_j\}}$, $\vec{\sigma}[k] = \vec{s}[k]$, (b) $\vec{\sigma}[j] = 0$, and (c) $\vec{\sigma}[i] = \vec{s}[i] + \vec{s}[j]$. It is easy to see that in the random-walks resulting when using the restart vector $\vec{\sigma}$ are similar to the random-walks resulting when using $\vec{s}$, except that all transportations to v_j are replaced with transportations to v_i (with the higher $seed\_ratio$ value among the two); thus, overall, more time will be spent on seed nodes when using $\vec{\sigma}$ instead of $\vec{s}$. It follows that when $seed\_ratio_i > seed\_ratio_j$, an optimal re-start vector cannot contain both v_i and v_j, contradicting the initial premise that $\vec{s}$ is an optimal re-start vector, such that both $\vec{s}[i] > 0$ and $\vec{s}[j] > 0$. □

In other words, since in practice it is highly unlikely that $seed\_ratio$ values will be equivalent for different seed nodes, the subset S_{crit} of S is likely to contain the one and only node, v_i, which has the highest $seed\_ratio_i$.

4) Efficient and Re-Use Promoting Computation: Given a graph, $G(V, E)$, a seed set, S, and a teleportation probability, β, one way to obtain the RPR-2 scores is to solve the linear optimization problem

$$\texttt{maximize} \quad \sum_{v_i \in S} \vec{\rho_2}[i]$$

subject to the constraints

$$\vec{\rho_2} = (1-\beta)\mathbf{T}_G\ \vec{\rho_2} + \beta\vec{s}, \sum_{v_i \in V} \vec{\rho_2}[i] = 1, \quad 0 \le \vec{\rho_2}[i] \le 1,$$

$$\sum_{v_i \in S} \vec{s}[i] = 1, \quad \forall_{v_i \in s} 0 \le \vec{s}[i] \le 1, \quad \forall_{v_i \notin s} \vec{s}[i] = 0.$$

While there are many efficient linear solvers that one can use to obtain a solution to the above optimization problem, there are two issues to consider: (a) in general, solving the optimization problem is more expensive than simply solving the linear equations for a given re-start vector $\vec{s}$, and (b) when the seed set S changes (even if the change is small, say one new seed node is considered or one of the seed nodes is dropped) the linear optimization problem needs to be reformulated and solved anew. In this subsection, we note that we can avoid treating the problem as an optimization problem (thereby reducing its cost) and, in the meantime, also support the re-use of existing solutions, by converting the problem into a set of single-seed PPR computations.

Converting the Problem into a Set of Linear Equations. Given a graph, $G(V, E)$, a seed set, S, and an overall teleportation probability, β, we reformulate the problem (relying on the observation in Section III-C3) as follows:

- *Step 1.* for each $v_i \in S$, solve the linear equation $\vec{\pi_i} = (1-\beta)\mathbf{T}_G\ \vec{\pi_i} + \beta\vec{s_i}$, where $\vec{s_i}$ is a re-start vector such that $\vec{s_i}[i] = 1$ and $\forall_{j \neq i}\ \vec{s_i}[j] = 0$;

- *Step 2.* Next, for each $v_i \in S$, compute $\Pi(v_i) = \sum_{v_j \in S} \vec{\pi_i}[j]$;

- *Step 3.* Let S_{crit} be the (small) subset of S, where $S_{crit} = argmax_{v_i}(\Pi(v_i))$;

- *Step 4.* If S_{crit} is singleton (i.e., $S_{crit} = \{v_i\}$) then $\vec{\pi_i}$ gives the RPR-2 scores; i.e., $\rho_2 = \vec{\pi_i}$; else (i.e., if S_{crit} is not a singleton), then $\rho_2 = \frac{1}{|S_{crit}|}\sum_{v_i \in S_{crit}} \vec{\pi_i}$.

Note that, since in general the seed set S includes relatively few nodes, the above formulation requires the solution of a small number of single-seed PPR problems. This is especially advantageous when G is large as we can leverage any of the highly effective approximation algorithms [2], [4], [11], [13], [15], [17], [23] or parallelized implementations [3], [20] for computing these PPR scores. Most importantly, the first step of the algorithm (where we solve a linear equation independently for each seed node) can be trivially parallelized by assigning each node to a different computation unit.

Solution Re-Use for Incremental Computation. Given a graph, $G(V, E)$, a seed set, S, and an overall teleportation probability, β, assume that we have already computed $\vec{\pi_i}$ and $\Pi(v_i)$ for all $v_i \in S$. Let S_{new} be a new seed set, let $\Delta S^+ = S_{new} \setminus S$ denote the new nodes in the seed set and $\Delta S^- = S \setminus S_{new}$ denote the set of nodes dropped from the seed set. We can incrementally compute the RPR-2 scores as follows:

- *Step 1.* For each $v_i \in \Delta S^+$, solve the linear equation $\vec{\pi_i} = (1-\beta)\mathbf{T}_G\ \vec{\pi_i} + \beta\vec{s_i}$, where $\vec{s_i}$ is a re-start vector such that $\vec{s_i}[i] = 1$ and $\forall_{j \neq i}\ \vec{s_i}[j] = 0$;

- *Step 2.* Next, for each $v_i \in \Delta S^+$, compute $\Pi_{new}(v_i) = \sum_{v_j \in S_{new}} \vec{\pi_i}[j]$;
- *Step 3.* Also, for each $v_i \in S_{new} \cap S$, compute $\Pi_{new}(v_i) = \Pi_{new}(v_i) + \sum_{v_j \in \Delta S^+} \vec{\pi_i}[j] - \sum_{v_j \in \Delta S^-} \vec{\pi_i}[j]$;
- *Step 4.* Given these, once again, let S_{crit} be the (small) subset of S, where $S_{crit,new} = argmax_{v_i}(\Pi_{new}(v_i))$, and compute the ρ_2 scores as $\rho_2 = \frac{1}{|S_{crit,new}|} \sum_{v_i \in S_{crit,new}} \vec{\pi_i}$.

It is easy to see that, when ΔS^+ and ΔS^- are small, the RPR-2 computations can be done very fast (if necessary, leveraging approximation algorithms [2], [4], [11], [13], [15], [17], [23] and/or parallel implementations [3], [20] for computing the new PPR scores). Once again, the first step of the algorithm (where we solve a linear equation independently for each new seed node) can be trivially parallelized by assigning each node to a different computation unit.

D. RPR-3: Teleportation-Discounted, Seed-Set Maximal PPR

As we have seen in Section III-B, one disadvantage of the use of standard PPR scores is that the teleportation contribution `t-PPR` score of a seed node (which is exactly $\frac{\beta}{|S|}$ for each seed node) may not capture how significant the node is within the context defined by the entire seed set S. This is especially true in the case of RPR-2 scores, where the set, S_{crit}, of seed nodes selected for re-start is very small. In fact, when S_{crit} is singleton (as most likely), the only seed node in S_{crit} will have at least β overall RPR-2 score. Therefore, when computing the *teleportation-discounted, seed-set maximal PPR scores* (or RPR-3 scores, also denoted as $\vec{\rho_3}$), we replace the use of $\vec{\pi_i}$ vectors for each $v_i \in S$, with the RPR-1 vectors $\vec{\rho_{1,i}}$ as introduced in Section III-B.

IV. Experimental Evaluation

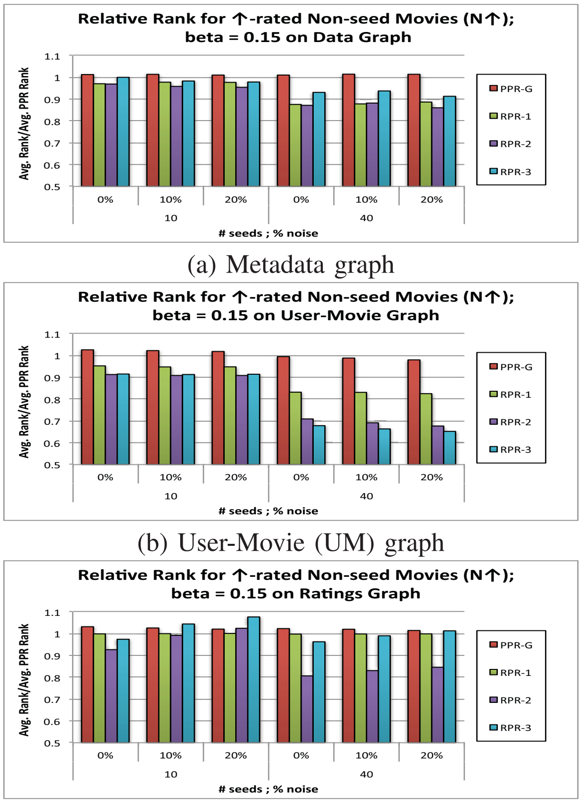
We ran the experiments on a quad-core Intel(R) Core(TM)i5-2400 CPU @ 3.10GHz machine with 8.00GB RAM. All codes are implemented in Matlab and run using Matlab 7.11.0 (2010b).

A. Data Sets

For comparing the different RPR alternatives' performances against conventional PPR, we used the IMDB and MovieLens datasets available from [25], [26]. This dataset contains metadata (e.g. actors, directors) about 1681 movies as well as a total of 100K ratings (between 1, for "dislike" (↓), to 5, for "like" (↑)) provided by 943 users. From this graph, we have constructed three data and social graphs, with distinct semantics and topological properties: (a) Metadata Graph: In the metadata graph, nodes represent the data elements (such as movies) and the edges represent relationships between these data elements (such as an actor playing in a movie). The data graph contains 1272 nodes and 60K relationship edges (with average node degree of ∼47). (b) User-Movie (UM) Graph: In the UM graph, nodes represent users and movies. There is an edge between a user-movie pair if the user has watched the movie (indicated by the existence of a rating). This graph has 1682 movie nodes, 943 user nodes, and 200K directional (user to movie) edges (with average node degree of ∼76). (c) Ratings Graph: The ratings graph consists of the same nodes and edges as the user-movie (UM) graph. However, each ratings edge $u_i \rightarrow m_j$ has an associated numeric weight between 1 and 5, reflecting user u_i's preference for movie m_j.

Parameter	Range	Default Value
# of seeds	{10, 40}	10
% of noise in the seed set	{0%, 10%, 20%}	10%

TABLE III. Personalized PageRank evaluation parameters



(a) Metadata graph

(b) User-Movie (UM) graph

(c) Ratings graph

Fig. 3. Average ranks of non-seed ↑-rated movies relative to their ranks returned by conventional PPR (the ratio for PPR itself is 1.0): the lower is the ratio, the better is the measure

Note that the *metadata graph* captures no knowledge about users and their preferences. The *user-movie (UM) graph*, which can be seen as a rich social network of users and movies, captures which users judged (rated) which movies, but does not capture the value of the rating. The *ratings graph* also captures users' declared preferences in the form of edge weights.

In these experiments, we set the default value of β to 0.15 as is commonly done. Results for other β values are similar.

B. Evaluation Strategies

1) Effectiveness: In order to measure effectiveness of the scores computed for ranking movies, we rely on the following criteria: **Relevance:** If we select a subset of a user's ↑-rated movies as seeds, the scores should be such that the user's remaining ↑-rated movies should rank well, whereas user's ↓-rated movies should rank poorly. **Robustness:** Moreover, if the scores are robust, then even if the seed set contains some small number of ↓-rated movies, this should not negatively affect the rankings significantly.

As shown in Table III, we consider seed sets of different sizes (10 and 40). For each configuration, we select those users who have sufficient ↑-rated movies (e.g., if the target seed set size is 10, then we pick those users with at least 10 ↑-rated movies): For the UM and Ratings graphs, there are 313 users for seed set size 10 and 64 users for seed set size 40. For the Metadata graph, there are 139 users when the seed set size is 10 and 42 users when the seed set size is 40. For each user, we created random seed sets with different degrees of noise (see Table III for the experiment parameters). Each seed set consists of a number of movies rated "↑" by the user (for measuring relevance) and a smaller number of movies rated "↓" by the same user. The "↓" movies included in the seed set act as noise (and serve for measuring robustness). For each configuration, we have considered 10 different (randomly picked) seed sets. For each seed set, we treated the rest of the movie ratings by this user as the **ground truth** to help measure the following effectiveness criteria based on the transition probabilities implied by the underlying graph:

- *Recommendation Effectiveness – Average rank for non-seed ↑-rated movies ($AvgRank_{(N\uparrow)}$):* Movies that we know (from the ground truth) that the user would like, but are not included in the seed set are *expected to rank well; i.e., have small average rank values*.
- *Seed Differentiation Power – Average rank for ↓-rated seed movies ($AvgRank_{(S\downarrow)}$):* "Noise" in the seed set (i.e., movies we know from the ground truth that the user does not like, but nevertheless included in the seed set) are *expected to have large average rank values*, even though they are in the seed set. Note that, since a well ranked ↓-rated seed would imply that *the movies neighboring this noisy seed* would also be well ranked, the average rankings of the seed nodes help us (indirectly) quantify the impact of noise on the neighbors of the seeds.
- *Seed-Bias Elimination – Average rank for ↑-rated seed movies ($AvgRank_{(S\uparrow)}$):* Movies that the user likes and are included in the seed set are *expected to rank well and have small average rank values*.

2) Efficiency: We compute matrix algebra based formulations of PPR and RPR using Matlab. The RPR-2 and RPR-3 schemes, which seek seed-maximal solutions, are implemented by converting the maximization problem to a set of linear equations (as discussed in Sections III-C2 through III-C4): these enable the same evaluation mechanism for PPR, PPR-G, and RPR-1 to be applicable also for RPR-2 and RPR-3, making the accuracy and execution time comparisons straightforward.

C. Results

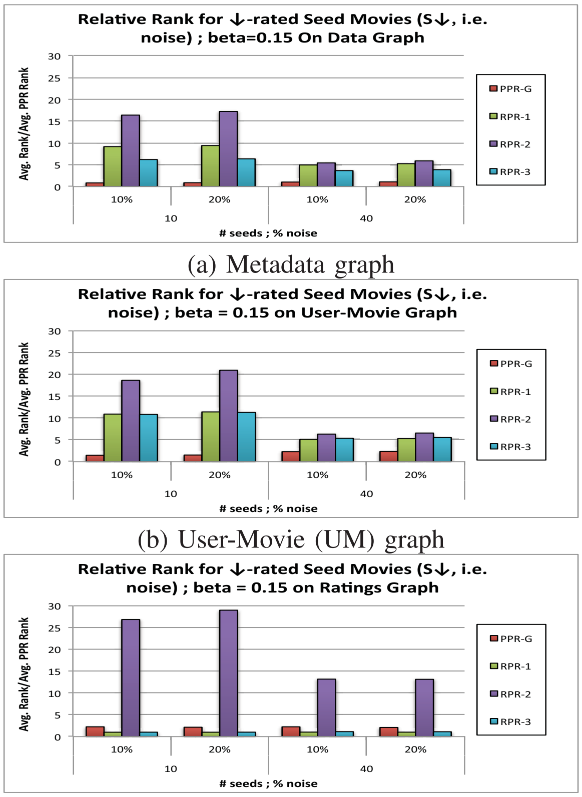
1) Effectiveness Evaluations: Figures 3 through 5 compare and evaluate the effectiveness of the different ranking algorithms described in this paper, based on the three effectiveness criteria $C = \{N\uparrow, S\uparrow, S\downarrow\}$ listed above. For each of these three criteria, we compare the rankings returned by algorithm A to the rankings returned by the conventional PPR (i.e., PPR with uniform teleportation probabilities) using the measure

$$relative\_rank(A,C) = (AvgRank_C\ by\ A)/(AvgRank_C\ by\ PPR).$$

This measure helps us to observe how algorithm A handles seed noise relative to PPR with no seed differentiation.

Recommendation Effectiveness: Firstly, let us consider Figure 3 which compares the average user rankings of ↑-rated movies that were not included in the seed set. Since the primary goal of a movie ranking system is to locate non-seed movies that the user would enjoy ($N\uparrow$) and rank them earlier than the other movies, *the smaller the value of the $relative\_rank(A, N\uparrow)$, the better would be the recommendations returned by the algorithm A.*

- As we see in the Figure 3, PPR-G does not provide any significant advantages over pure PPR, unless the teleportation rate is very small (i.e., not much significance is given to the seed set) or the seed set contains large amounts of noise.



(a) Metadata graph

(b) User-Movie (UM) graph

(c) Ratings graph

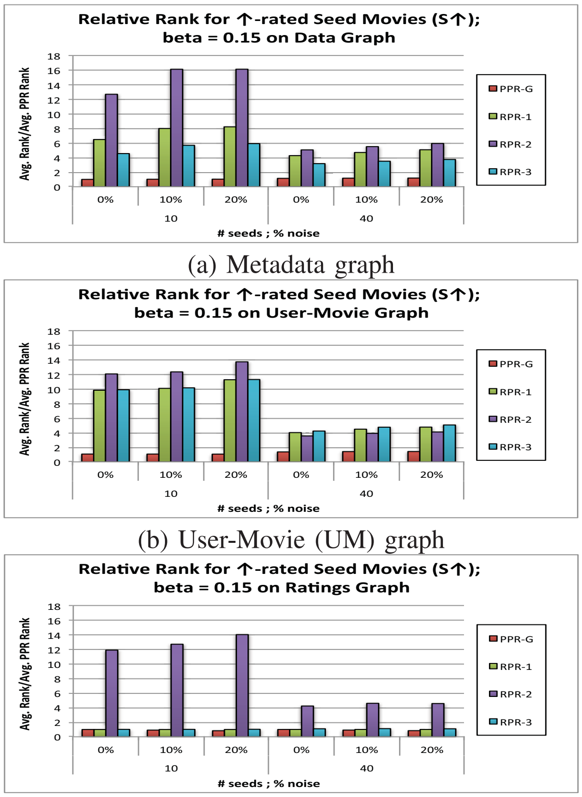
Fig. 4. Average ranks of ↓-rated movies included in the seed set relative to their ranks returned by conventional PPR (the ratio for PPR itself is 1.0): the higher is the ratio, the better is the measure

- Figures 3 through 5 show that the proposed RPR schemes lead to significant improvements in the rankings, with RPR-2 providing the most consistent improvements.
- *Teleportation-discounting* (used in RPR-1 and RPR-3) is effective in (metadata and UM) graphs, which do not properly capture user preferences. *Seed-set maximization* (used in RPR-2), however, provides benefits for all graphs, including the ratings graph, which reflects the user preferences in the transition probabilities.
- It is important to note that the RPR techniques provide better recommendation rankings even in situations where the seed set contains 0% artificially introduced noise, confirming that RPR provides better personalization given user history.

In addition to the above key observations, we also note that on the UM graph (with higher average node degree), RPR-3 provides the highest improvements. On the metadata graph (with lower average node degree), on the other hand, RPR-3 is less advantageous than RPR-2.

Seed Differentiation Power: Figure 4 compares the average user rankings of ↓-rated movies that were included in the seed set. In this case, the *higher the relative rank, the better is the algorithm in differentiating the noise in the seed set.*

- As we seen in the figure, once again, PPR-G does not provide any significant advantages over pure PPR;
- The proposed RPR algorithms, on the other hand push the ↓rated seed nodes (i.e., noise) significantly further down in the



(a) Metadata graph

(b) User-Movie (UM) graph

(c) Ratings graph

Fig. 5. Average ranks of ↑-rated movies included in the seed set relative to their ranks returned by conventional PPR (the ratio for PPR itself is 1.0): the lower is the ratio, the better is the measure

overall ranking relative to conventional PPR, indicating that RPR algorithms are effective in eliminating seed-bias; and

- As before, *teleportation-discounting* (used in RPR-1 and RPR-3) is effective in (metadata and UM) graphs, which do not capture user preferences; but *seed-set maximization* (used in RPR-2), provides benefits for all graphs.
- In fact, comparing the UM and ratings graphs, we see that RPR-2 provides higher average rankings for $S\downarrow$movies in the ratings graph, indicating that *seed-set maximization* leverages the user preference information embedded in the transition matrix well.

Seed-Bias Elimination: As we discussed in Section III, PPR assumes that all the nodes in the seed set are very important and thus they tend to rank better than most (if not all) non-seed nodes. However, *we expect that a seed-bias eliminating ranking system would push some of the highly rated seeds further down in the rankings to bring up those movies that are good, but not used as seeds*.

- In Figure 5, we see that this is indeed true for the proposed RPR schemes: as we would expect from a good seed bias eliminating algorithm, expected, the *teleportation-discounting* (for metadata and UM graphs) and *seed-set maximization* (for all graphs) increase the relative rankings of the ↑-rated seed nodes, for accommodating the better rankings of good, but not seed nodes – as we have already seen in Figure 3.

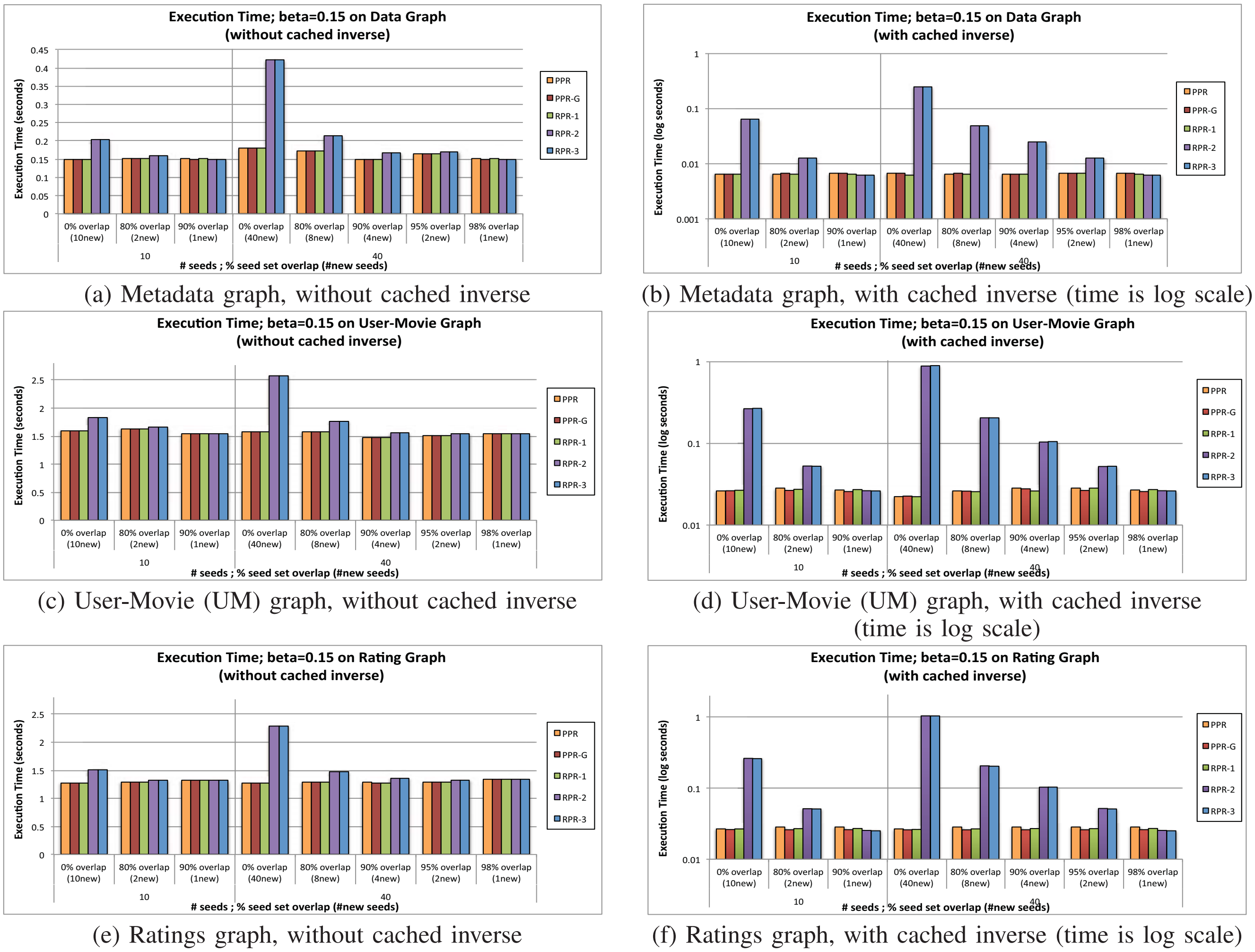
Effectiveness Summary: The above experiments have shown that *teleportation-discounting* is an effective technique in improving ranking effectiveness in graphs which are preference-neutral (like the metadata and user-movie, UM, graphs). The *seed-set maximization* technique, on the other hand, performs well for all graphs (including those that already capture user preferences) and noise scenarios and thus should be the preferred ranking technique (according to the results, even in situations where the noise is 0%).

2) Efficiency Evaluations: In Figure 6, we consider the execution times of the different personalized PageRank algorithms considered in this paper (without explicit parallelization). In particular, we consider three scenarios: Fresh graph, fresh seed set: In the first scenario, we are given a graph and a fresh set of seeds and the computation starts from scratch (Figures 6(a), (c) and (e), columns corresponding to 0% overlap). Cached graph, fresh seed set: In the second, the graph is fixed and the matrix inverse has already been computed and cached; given a fresh seed set, this cached inverse is used for computing the scores and rankings (Figures 6(b), (d) and (f), columns corresponding to 0% overlap). Fresh/cached graph, overlapping (i.e., cached) seed set: In the third scenario (Figures 6(a) through (f), columns corresponding to more than 0% overlap), the new seed set overlaps with seed sets considered in the past and this overlap is leveraged as described in Section III-C4.

- As we see in Figures 6(b) and (d), when a cached inverse of the transition matrix is available and the seed set overlap is low, RPR-2 and RPR-3 schemes (which need to solve multiple linear equations per Section III-C4) are slower than PPR, PPR-G, and RPR-1 schemes. Thus, when cached inverse is available and seed overlaps are small, we recommend using the RPR-1 scheme which (as we have seen earlier) is more robust than PPR and PPR-G and, also, as fast.
- The figure also shows that the difference between the various strategies is small or non-existent (a) when the graph itself is fresh (i.e., no cached inverse is available), (b) the seed set is small, or (c) the overlap between the current seed set and the seeds considered in the past is large (i.e., cached solutions for individual seeds can be reused per Section III-C4). In these cases, RPR-2 and RPR-3 are competitive in execution times and should be used where the accuracy provided by the *seed-set maximization strategy* is critical.
- Also, we would like to remind that, as discussed in Section III-C4, RPR is trivially parallelizable by mapping new seeds such that each computation unit processes only one (or few) new seeds (see the cases marked "**1new**" in Figure 6 for the impact of processing *only one seed per processing unit*).

V. Conclusions

In this paper, we have shown that conventional personalized PageRank (PPR) algorithms associate *unnecessarily high bias* to the seed nodes and this negatively affects the node rankings when the seed set is *incomplete* and/or *noisy*. To deal with this problem, we propose three alternative *robust personalized PageRank* (RPR) algorithms that eliminate the potential noise in the seed set. We have shown that a novel *teleportation discounting* technique ensures that rankings are not overly biased towards the seed nodes and a novel *seed-set maximal PPR principle* helps differentiate among the seeds by considering the overall context defined by the seed set.



(a) Metadata graph, without cached inverse

(b) Metadata graph, with cached inverse (time is log scale)

(c) User-Movie (UM) graph, without cached inverse

(d) User-Movie (UM) graph, with cached inverse (time is log scale)

(e) Ratings graph, without cached inverse

(f) Ratings graph, with cached inverse (time is log scale)

Fig. 6. Execution times for different measures (w/o explicit parallelization): we consider situations where (a,c,e) personalized PageRank computation starts from scratch and (b,d,f) where the cached matrix inverses are leveraged. For each configuration, we consider different rates of updates to the seed set.

REFERENCES

[1] Andersen R, et al. (2008) Trust-based recommendation systems: an axiomatic approach. WWW, pages 199-208, 2008.

[2] Avrachenkov K, et al. (2011) Quick Detection of Top-k Personalized PageRank Lists. WAW, pages 50-61, 2011,

[3] B. Bahmani., K. Chakrabarti, D. Xin, Fast personalized PageRank on MapReduce. In SIGMOD, pages 973-984, 2011.

[4] Bahmani B., et al. Fast incremental and personalized PageRank. PVLDB. 4, 3, pages 173-184, 2010.

[5] Balmin A., et al. ObjectRank: Authority-based keyword search in databases. VLDB, pages 564-575, 2004.

[6] Borgatti MG., et al. Network measures of social capital. Connections 21(2):27-36, 1998.

[7] Brin S., et al. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems 30: 107-117, 1998.

[8] Buckley C., Voorhees E.M. Retrieval evaluation with incomplete information. SIGIR, pages 25-32, 2004.

[9] Candan K.S. and Li W.S. Using random walks for mining web document associations. PAKDD, pages 294-305, 2000.

[10] Candan K.S., et al. Reasoning for Web document associations and its applications in site map construction. Data Knowl. Eng. 43(2): 121-150, 2002.

[11] Chakrabarti S. Dynamic personalized pagerank in entity-relation graphs. WWW, pages 571-580, 2007.

[12] Chen M., et al. Clustering via random walk hitting time on directed graphs. AAAI, pages 616-621, 2008.

[13] Csalogany K., et al. Towards Scaling Fully Personalized PageRank: Algorithms, Lower Bounds, and Experiments Internet Math. 2,3, pages 333-358, 2005.

[14] Fouss F., et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. TKDE, pages 1041-4347, 2007.

[15] Fujiwara Y., et al. Fast and exact top-k search for random walk with restart. PVLDB. 5, 5, pages 442-453, 2012.

[16] Z. Guan, et al. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. SIGIR, 540-547, 2009

[17] Gupta M., et al. Fast algorithms for Top-k Personalized PageRank Queries. WWW, pages 1225-1226, 2008.

[18] Haveliwala T.H. Topic-sensitive PageRank. WWW, 517-526, 2002.

[19] Kim H.N, El-Saddik A. Personalized PageRank vectors for tag recommendations: inside FolkRank. RecSys, 45-52, 2011.

[20] Malewicz G., et al. Pregel: a system for large-scale graph processing. SIGMOD, pages 135-146, 2010.

[21] Mei Q., et al. Query suggestion using hitting time. CIKM, 2008.

[22] Sarkar P., et al. Fast incremental proximity search in large graphs. ICML, pages 896-903, 2008.

[23] Tong H., et al. Fast Random Walk with Restart and Its Applications. ICDM, pages 613-622, 2006.

[24] White D.R., et al. Betweenness centrality measures for directed graphs. Social Networks, 16, pages 335-346,1994.

[25] http://www.imdb.com/

[26] http://www.grouplens.org/