Personalized PageRank in Uncertain Graphs with Mutually Exclusive Edges*

Jung Hyun Kim
Arizona State University
Tempe, AZ 85287
jkim294@asu.edu

Mao-Lin Li
Arizona State University
Tempe, AZ, USA 85287
maolinli@asu.edu

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

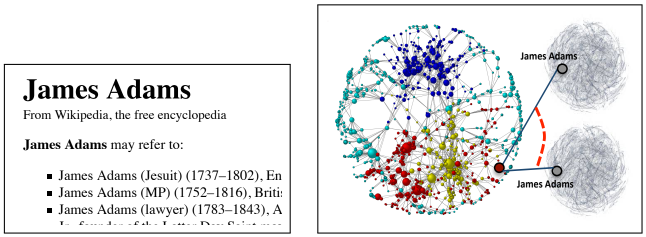
Maria Luisa Sapino
University of Torino
I-10149 Torino, Italy
mlsapino@di.unito.it

ABSTRACT

Measures of node ranking, such as personalized PageRank, are utilized in many web and social-network based prediction and recommendation applications. Despite their effectiveness when the underlying graph is certain, however, these measures become difficult to apply in the presence of uncertainties, as they are not designed for graphs that include uncertain information, such as edges that mutually exclude each other. While there are several ways to naively extend existing techniques (such as trying to encode uncertainties as edge weights or computing all possible scenarios), as we discuss in this paper, these either lead to large degrees of errors or are very expensive to compute, as the number of possible worlds can grow exponentially with the amount of uncertainty. To tackle with this challenge, in this paper, we propose an efficient *Uncertain Personalized PageRank (UPPR)* algorithm to approximately compute personalized PageRank values on an uncertain graph with edge uncertainties. *UPPR* avoids enumeration of all possible worlds, yet it is able to achieve comparable accuracy by carefully encoding edge uncertainties in a data structure that leads to fast approximations. Experimental results show that *UPPR* is very efficient in terms of execution time and its accuracy is comparable or better than more costly alternatives.

1 INTRODUCTION

Measures of node ranking are used in many web and social media based prediction and recommendation applications [6, 24, 27]. There are several ways to rank nodes in a graph ranking, including the well known personalized PageRank (PPR) measure [9, 18], which weights the nodes in a given graph based on their positions relative to a given seed set of nodes (Section 2).



James Adams
From Wikipedia, the free encyclopedia

James Adams may refer to:

- James Adams (Jesuit) (1737–1802), En
- James Adams (MP) (1752–1816), Britis
- James Adams (lawyer) (1783–1843), A

(a) Ambiguity in data

(b) Uncertain edge

Figure 1: Ambiguity in Wikipedia and its potential impact on the proximity/cluster analysis

Despite their effectiveness when the underlying graph is certain, these measures become difficult to apply in the presence of graph uncertainties, as they are not designed for graphs that include uncertain information. Unfortunately, in many real world web and social-network based applications, it may not be possible to obtain a perfect and complete structure of the underlying knowledge graph for various reasons: This may be due to lack of information, noise in data collection, or privacy concerns [17].

Most existing works on graph uncertainty consider *existence uncertainty*, where a given edge exists probabilistically and the existence probabilities of the individual edges are assumed to be independent from each other [2, 10, 16, 20, 26, 29]. In practice, however, this assumption does not always hold: we may be aware of the existence of an edge, but we may not know between which pairs of nodes the edge exists. For example, we may be able to deduce that one of the several friends of an individual in a social network may be his/her father, but we may not know which friend. As another example, we may know that a name referred to in a web document is one of the many named entities in a knowledge base, but we may not know which one is the correct entity (Figure 1(a)).

In this paper, we propose an *uncertain edge model with mutual exclusion* that can handle such general forms of uncertainty[1] and consider the node ranking problem in the presence of such edges. Obtaining node rankings in such a graph is difficult because addition or removal of one single edge can have a drastic effect on proximity [11]: e.g., addition of just one edge may be sufficient to

[*] This work has been funded by NSF grants #1339835, #1318788, #1610282, #1633381, and the EU-H2020 grant #690817.

[1] For relational data, this type of uncertainty is also known as "partial maybe null", where one is not sure if the attribute has a value or not, but if the value exists, then it must be within a specified set[1, 7]

SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan
© 2017 ACM. 978-1-4503-5022-8/17/08...$15.00
DOI: 10.1145/3077136.3080794

link two otherwise distant node clusters, thereby significantly altering the proximities of a large number of pairs of nodes in the graph (Figure 1(b)). A naive way to deal with this would be to measure *expected node proximities* by taking into account the likelihoods of different interpretations and the node proximity measurements corresponding to each interpretation: one can

(1) first enumerate all possible interpretations (or possible worlds) of the uncertain graph, where each interpretation is a possible certain graph;
(2) compute node proximity under each possible world; and
(3) finally, combine all these node proximity measurements into a single *expected proximity* value.

It is, however, easy to see that an exhaustive enumeration based approach will quickly become intractable since (as we see in Section 3) the number of possible worlds can grow exponentially with the amount of uncertainty in the graph. To tackle this challenge, in this paper, we propose an efficient *Uncertain Personalized PageRank (UPPR)* algorithm to approximately compute personalized PageRank values on an uncertain graph with edge uncertainties. *UPPR* avoids enumeration of all possible worlds, yet it is able to achieve comparable accuracy by carefully encoding edge uncertainties in a data structure that leads to fast approximations. Experiment results show that *UPPR* is very efficient in terms of execution time (multiple orders faster than other algorithms with similar accuracy) and its accuracy is close to perfect.

In the next section, we discuss the related literature. In Section 3, we introduce the uncertain graph model. In Section 4, we discuss alternative "naive" techniques and discuss their individual shortcomings. Then, in Section 5, we present the proposed efficient and effective *uncertain personalized PageRank (UPPR)* technique. We evaluate the various techniques discussed in the paper in Section 6 using several data sets and conclude in Section 8.

2 RELATED WORKS

2.1 Graphs with Uncertainty

Uncertain graphs are common in many applications. For example, in biological protein interaction networks, uncertainty may be introduced when the existence of certain interactions are often only statistically probable [16, 20]. In communication networks, possibility of link failure needs to be accounted for in finding stable and reliable paths for packet delivery with minimum cost: this involves taking into account several forms of uncertainty, including *existence uncertainty*, *ambiguity*, and *confusion* on edges [10].

In web-based applications, such as social networks, uncertainties may exist due to inherent lack of prior knowledge regarding the existence of friendship or influence flow among the users in the underlying network [17] and it may be critical to take into account such forms of uncertainty in predicting which nodes are likely to be connected to which other nodes [24]. Other graph analysis operations that are affected from graph uncertainty include shortest paths, reachability analysis, and subgraph searching. A common challenge is that, in the presence of uncertainty, (already expensive) graph operations becomes more expensive. [8] presented an interval labeled edge model and discussed efficient computation of minimum paths and trees on such uncertain graphs without having to enumerate all possible worlds. [26] and [29] also focused on shortest paths, but on graphs where edges have probabilistic interpretations for existence in uncertain graphs. Given edges that are accompanied with the probability of existence, [16, 20] propose ways to compute reliability and reachability efficiently through Monte-Carlo sampling. [30] proposed pruning techniques to reduce the complexity of subgraph searching and subgraph pattern mining in uncertain graphs by avoiding enumeration of all possible worlds of the uncertain graph.

2.2 Node Ranking in Uncertain Graphs

PageRank is a widely-used measure to compute node importance / significance in a graph [5]. It takes into account the connectivity of nodes in the graph by defining the score of the node $v_i \in V$ as the amount of time spent on v_i in a sufficiently long random walk on the graph. The personalized PageRank (PPR) [9, 18] technique extends this in a way that takes into account the context defined by a given set of *important* nodes: given a set of seed nodes $S \subseteq V$, the PPR scores can be represented as a vector $\overrightarrow{r}$, where $\overrightarrow{r} = \alpha \mathbf{T} \overrightarrow{r} + (1 - \alpha)\overrightarrow{s}$, where $\overrightarrow{s}[i] = \frac{1}{\|S\|}$ if $v_i \in S$ and $\overrightarrow{s}[i] = 0$, otherwise. Intuitively, given a set of nodes $S \subseteq V$, instead of jumping to a random node in V with probability $(1 - \alpha)$, the random walk jumps to one of the nodes in the seed set, S. Since we constrain the teleportation jumps from any node in the graph to only the given set of important seed nodes, then the random-walk spends more time on nodes that are close to the seeds and, thus, those nodes are declared more significant based on the context defined by the seed nodes. Due to the cost of obtaining exact PPR scores, non-exact solutions (based on low rank decomposition [28] or Monte Carlo methods [22]) have been proposed.

Several works considered the problem of ranking on graphs with different forms of uncertainties. [13] considered PageRank when web graphs contain erroneous link information and proposed an approximate solution using interval matrices – the proposed approach captures the PageRank scores of the nodes affected by fragile links in terms of lower and upper bounds of PageRank values. A different node-centric uncertain graph model and node ranking approach are presented in [23]: in particular, [23] collapses the uncertain parts of a graph into a *cloud* graph, where the end of every undetected link is connected to this cloud graph and computes PageRank scores on this transformed graph. [12] considered uncertain graphs, where edges are annotated with existence probabilities and extended the SimRank measure [14] under probabilistic interpretations of edge existence and transition matrices.

In this paper, we propose a more general uncertainty model (of which the *existence* uncertainty considered by the works listed above is special case) and discuss efficient ways to compute PPR under this more powerful model.

3 PROBLEM FORMULATION

Let $G = (V, E)$ be a directed graph with a set, V, of nodes and a set, E, of edges. Conventionally, each edge $e \in E$ is defined using two nodes in the graph: a source node $source(e) \in V$ and a target node $target(e) \in V$. In this paper, on the other hand, we divide the graph edges into *certain* and *uncertain* edges.

Definition 3.1 (Certain edges). A certain edge $e_+ \in E$ has well defined source and target nodes, v_{source} and v_{dest}. We denote

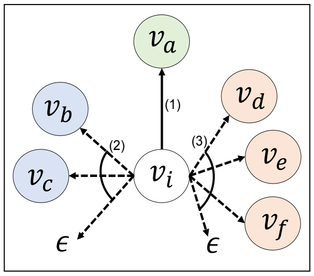


Figure 2: A graph with certain and uncertain edges

this with $source(e_+) = \{v_{source}\}$ and $target(e_+) = \{v_{dest}\}$. We denote the subset of E consisting of E's certain edges as E_+. ◇

In Figure 2, $e_{+_{(1)}} = \{\langle v_i, v_a \rangle\}$ is a certain edge from v_i to v_a. Note that, since $\|source(e_+)\| = \|target(e_+)\| = 1$, this edge type does not include any uncertain information. In this paper, we refer to this certainty as having a unique possible world. Each uncertain edge, on the other hand, can represent multiple possible worlds:

Definition 3.2 (Uncertain Edges). An uncertain edge $e_- \in E$ has a well defined source node but does not have a well defined target node.[2] More specifically, we have

- $source(e_-) \subseteq V$,
- $target(e_-) \subseteq V \cup \{\epsilon\}$ and $target(e_-) \neq \{\epsilon\}$, and
- $\|source(e_-)\| = 1$ and $\|target(e_-)\| > 1$.

Above ϵ denotes a non-existing node. We denote the subset of E consisting of all of E's uncertain edges as E_-. ◇

Figure 2 includes two uncertain edges, $e_{-_{(2)}}$ and $e_{-_{(3)}}$ with different degrees. The uncertain edge $e_{-_{(3)}}$ captures a form of *uncertainty with mutual exclusion* among the edges from v_i to v_d, v_e, or v_f. This uncertainty, however, is independent from the existence uncertainty of $e_{-_{(2)}}$. Therefore, the proposed model allows as a special case the *independent existence uncertainty* model considered by many of the existing works [2, 10, 16, 20, 26, 29].

3.1 Possible Worlds of an Uncertain Edge

Each uncertain edge implicitly defines multiple possible worlds in which different interpretations are valid:

Definition 3.3 (Possible Worlds of an Edge under Mutual Exclusion Semantics). Let $e \in E$ be an edge. Let $source(e)$ denote a source node of the edge and let $target(e) \subseteq V \cup \{\epsilon\}$ denote the potential targets of the edge. Given this edge, we define *all possible worlds covered by this edge under mutual exclusion semantics* as

$$pw_{unique}(e) \quad = \quad \Big\{ \langle v_i, v_j \rangle \,|\, (v_i = source(e)) \wedge (v_j \in target(e)) \Big\}$$

The possible worlds covered by an uncertain edge consist of all combinations of target nodes; if a target node is potentially non-existent, then it is also a possible world. $\|pw_{unique}(e)\| = \|target(e)\|$ is the number of possible worlds on the edge, e ◇

In the example visualized in Figure 2, there are three possible worlds defined by $e_{-_{(2)}}$ ($= \{\langle v_i, v_b \rangle, \langle v_i, v_c \rangle, \langle v_i, \epsilon \rangle\}$ – the last one implying that this edge does not exist) and four possible worlds defined by $e_{-_{(3)}}$ ($= \{\langle v_i, v_d \rangle, \langle v_i, v_e \rangle, \langle v_i, v_f \rangle, \langle v_i, \epsilon \rangle\}$ – again the last one implying that this edge does not exist).

Note that under a more general interpretation, more than one of the potential combinations, implied by the uncertainty encoded in the edge, may be possible in the real world.

Definition 3.4 (Possible Worlds of an Edge under Multiple Edge Semantics). Let $e \in E$ be a certain or uncertain edge and $pw_{unique}(e)$ be the corresponding possible worlds covered by this edge under mutual exclusion semantics. Given this edge, we define *all possible worlds covered by this edge under multiple edge semantics* as all possible non-empty subsets of its target set[3]. Note that, since a possible world containing ϵ is equivalent to the world where ϵ has been removed, we have

$$\|pw_{multiple}(e)\| = \begin{cases} 2^{(\|pw_{unique}(e)\|-1)}, & \epsilon \in \text{target(e)} \\ 2^{\|(pw_{unique}(e)\|)} - 1, & \text{otherwise} \end{cases} \quad \diamond$$

Under these semantics, in the example in Figure 2, there would be $2^{(3-1)} = 4$ possible worlds defined by the uncertain edge $e_{-_{(2)}}$ and $2^{(4-1)} = 8$ possible worlds defined by $e_{-_{(3)}}$. For the certain edge $e_{(1)}$, this gives $2^{(1-1)} = 1$ possible world.

3.2 Possible Worlds of a Graph

Given the above definitions, we can now define the possible worlds of a graph with uncertainty:

Definition 3.5 (Possible Worlds of a Graph). Let $G = (V, E)$ be a directed graph which has a set of nodes V and a set of edges E. For all $e \in E$, let $pw(e)$ denote the possible worlds (under mutual exclusion or multiple edge semantics) of the edge e. We define *all possible worlds covered by this graph* as the Cartesian product of the possible worlds of edges: $pw(G) = \times_{e \in E} pw(e)$. ◇

If we reconsider Figure 2, under mutual exclusion semantics, this graph would have 1 × 3 × 4 = 12 possible worlds. In contrast, under the multiple edge semantics, the graph would have 1 × 4 × 8 = 32 possible worlds. Since uncertain edges have ≥ 2 possible worlds, it is easy to see that the size of the $pw(G)$ grows exponentially in the number of uncertain edges; i.e., $\|pw(G)\|$ is $O(2^{\|E_-\|})$.

3.3 PPR under Uncertainty

We now define personalized PageRank under uncertainty.

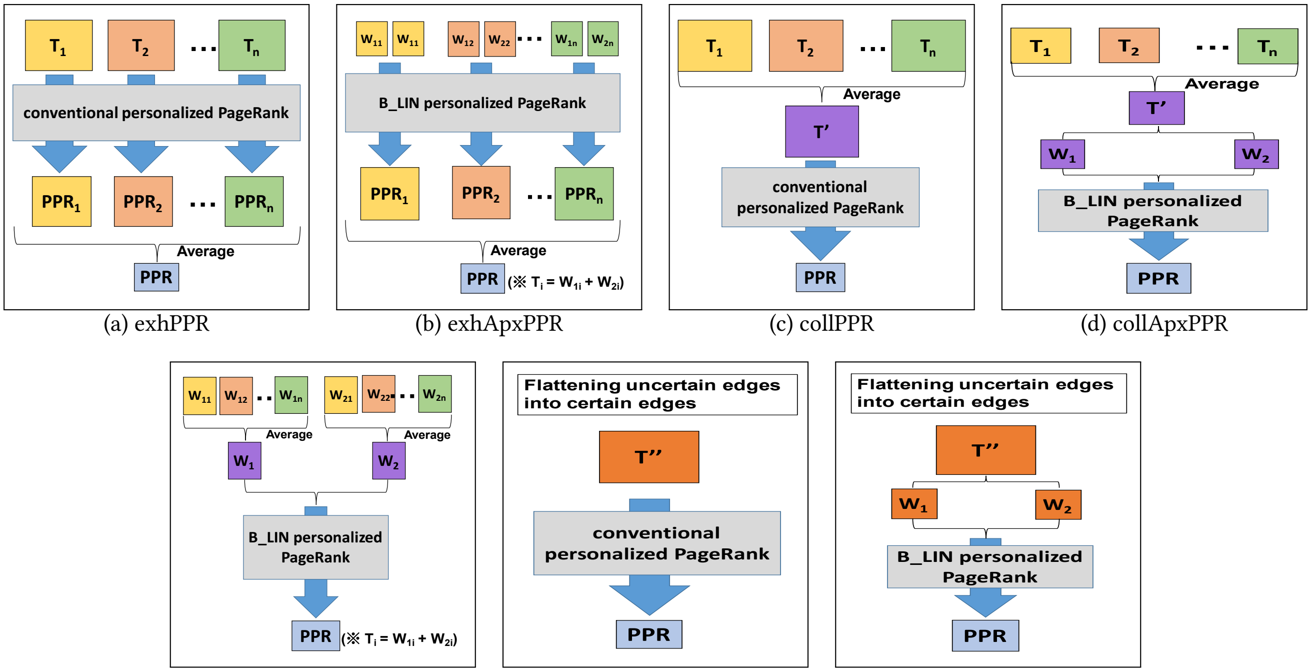
Definition 3.6 (Personalized PageRank under Uncertainty). Let $G(V, E)$ be an uncertain graph. Given a seed set, S, of nodes we can define the personalized PageRank vector, $\overrightarrow{r}$, for G as follows:

$$\overrightarrow{r} = \underset{G_i \in pw(G)}{AVG} \; PPR(G_i, S),$$

where G_i denotes a possible world implied by the uncertain graph G and $PPR(G_i, S)$ returns a personalized PageRank vector, $\overrightarrow{r}_i$, corresponding to G_i and seed set S. ◇

[2] Due to space constraints, in this paper we only deal with the case of uncertainty in the target nodes, while we consider the edges' source nodes as given.

[3] This can be extended to the case where there is a constraint in the number of real edges an edge can potentially represent.


T1
T2
Tn
conventional personalized PageRank
PPR1
PPR2
PPRn
Average
PPR
W11
W11
W12
W22
W1n
W2n
B_LIN personalized PageRank
(※ Ti = W1i + W2i)
T'
W1
W2
W21
Flattening uncertain edges into certain edges
T''

(a) exhPPR (b) exhApxPPR (c) collPPR (d) collApxPPR

(e) collApx2PPR (f) flatPPR (g) flatApxPPR

Figure 3: Alternative (naive) approaches for computing PPR values on an uncertain graph

Intuitively, under the assumption that all possible worlds are equally likely, the above definition of personalized PageRank corresponds to the *expected*[4] values of the node scores.

4 "NAIVE" APPROACHES

In this section, we present several (naive) approaches for computing PPR values on an uncertain graph (Figure 3):

4.1 Exhaustive Approaches

The most straightforward way to obtain the PPR values on an uncertain graph is to exhaustively enumerate all possible worlds, compute the PPRs for each possible world, and combine (i.e., average) the results. Obviously this exhaustive approach (exhPPR), visualized in Figure 3(a), is likely to be very expensive as it involves potentially exponential number of PPR computations.

One way to alleviate this cost is to rely on a fast approximate PPR technique (such as *B_LIN* [28], which partitions the given graph into subgraphs and pre-processes intra-partition edges, W_1, and inter-partition edges, W_2, on these subgraphs in a post-processing phase) to obtain PPR scores for each possible world (Figure 3(b)). Note that, while this exhaustive approximate approach, which we refer to as exhApxPPR, is likely to be faster than the basic approach, since it involves exponential number of (approximate) PPR computations, it is still likely to be prohibitively expensive.

4.2 Collapsing-based Approaches

Since the major cost of the exhaustive approach is the number of exhaustive PPR computations, one way to reduce the cost would be to enumerate all possible transition matrices corresponding to all possible worlds of the uncertain graph and then *collapse* these transition matrices into a single transition matrix by taking their average. After this, we can obtain the final PPR scores either by solving an exact PPR (collPPR, Figure 3(c)) or approximate PPR (collApxPPR, Figure 3(d)) problem.

Another alternative is to first partition each individual transition matrix of each possible world, G_i, and then *collapse* the intra-partition, W_{1i}, and inter-partition, W_{2i}, transition matrices for all possible worlds into an inter-partition and an intra-partition matrix to be processed using *B_LIN*[28] and combined in a post-processing phase. In Figure 3(e), we refer to this pre-partitioning based alternative approach as collApx2PPR.

Accuracy Problem with Collapsing: The collapsing based approach can lead to relatively large errors when uncertainty is concentrated around nodes with large PPR scores: Let G be an uncertain graph with two possible worlds with transition matrices, T_1 and T_2, respectively. Given these, we can compute the expected PPR scores as defined in the previous section as

$$\vec{r} = (\vec{r}_1 + \vec{r}_2)/2 = \left(\alpha\left(T_1\vec{r}_1 + T_2\vec{r}_2\right)\right)/2 + (1-\alpha)\vec{s},$$

where $\vec{s}$ is the teleportation vector for the seeds. In contrast, when using the collapsing based approach we instead compute

$$\vec{r}\,' = \alpha\left((T_1 + T_2)/2\right)\vec{r}\,' + (1-\alpha)\vec{s}.$$

Given these, the error term, $\vec{e} = \vec{r} - \vec{r}\,'$ can be obtained as

$$\vec{e} = \left(\alpha\left(T_1\vec{r}_1 + T_2\vec{r}_2\right)\right)/2 - \alpha\left((T_1 + T_2)/2\right)\vec{r}\,'.$$

[4] This can be extended to cases where each possible world has a different likelihood.

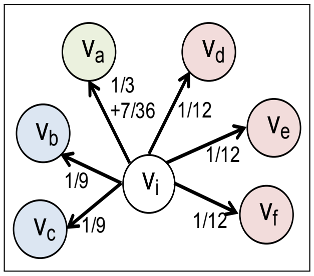


Figure 4: Flattening of the uncertain graph in Figure 2 into an (approximate) certain graph

Assuming that this error term is relatively small; i.e., $\vec{r} \sim \vec{r}'$, we can replace $\vec{r}'$ with $\vec{r} = (\vec{r}_1 + \vec{r}_2)/2$, to obtain

$$\begin{aligned}\vec{e} \quad &\sim \quad \left(\alpha\left(T_1\vec{r}_1 + T_2\vec{r}_2\right)\right)/2 - \alpha\left((T_1 + T_2)/2\right)\left(\left(\vec{r}_1 + \vec{r}_2\right)/2\right)\\ &\sim \quad ((T1 - T2)/4)\,\vec{r}_1 + ((T2 - T1)/4)\,\vec{r}_2.\end{aligned}$$

In other words, the error term is especially large when the uncertainties (i.e., differences between the transition matrices of the possible worlds) are concentrated around nodes with large PPR scores.

Execution Time Problem with Collapsing: Since they reduce the number of PPR computations to just one, the collapsing based approaches are likely to be much faster than the exhaustive approach. Nevertheless, since it involves the enumeration of all possible worlds before obtaining the collapsed transition matrix, the cost is still exponential in the number of uncertain edges.

4.3 Flattening-based Approaches

An alternative approach to avoid the enumeration cost of collapsing is to approximate the collapsed transition matrix by constructing it directly from the uncertain graph G by flattening each uncertain edge into certain edges. Let v_i be a node with c outgoing certain edges and u outgoing uncertain edges. To flatten the outgoing edges of a node v_i, we do the following:

1. Each outgoing certain edge is associated with $1/(c+u)$ transition probability.
2. Let e_- be an outgoing uncertain edge, with t targets
   (a) each non-ϵ target of e_- is given a transition probability of $(1/t) \times (1/(c+u))$
   (b) if ϵ is a target for e_-, then the corresponding $(1/t) \times 1/(c+u)$ transition probability is distributed among the c certain edges of v_i; if the vertex does not have any outgoing certain edges, then the probability is re-distributed among all the nodes in the graph.

See [19] for details. For instance, in the example visualized in Figure 2, since there are three outgoing edges, the probabilities of outgoing edges for v_i would be set as $\frac{1}{3}$ on the edge going to v_a, $\frac{1}{3} \times \frac{1}{3} = \frac{1}{9}$ on the edge going to v_b and v_c, and $\frac{1}{3} \times \frac{1}{4} = \frac{1}{12}$ on the edge going to v_d, v_e, and v_f. Note that, when ϵ is selected for any of the outgoing edges, the only available traversal direction is towards v_a. Therefore, this would lead to an additional transition probability of $\frac{1}{9} + \frac{1}{12}(= \frac{7}{36})$ towards v_a. This is visualized in Figure 4.

Once the flattened transition matrix is obtained, we can solve the final PPR scores either using an exact PPR (flatPPR, Figure 3(f)) or an approximate PPR (flatApxPPR, Figure 3(g)) technique. Note that, while they are likely to be faster than both exhaustive and collapsing-based approaches, flattening-based solutions further compound the accuracy problems.

5 UPPR: PROPOSED APPROACH

We propose an efficient and effective *Uncertain Personalized PageRank (UPPR)* algorithm to approximately compute personalized PageRank values on an uncertain graph with edge uncertainties. In particular, *UPPR* avoids enumeration of all possible worlds, yet is able to achieve high accuracy by carefully encoding edge uncertainties in a data structure that leads to good approximations.

5.1 Special Case: Two Possible Worlds

Let $G(V,E)$ be an edge uncertain graph, Let us split $G(V,E)$ into two subgraphs: a subgraph, $G_c(V,E_c)$, consisting of certain edges, and a subgraph, $G_u(V,E_u)$, consisting of uncertain edges. Let us first consider the special case where $G_u(V,E_u)$ defines only two possible worlds. In Section 5.2, we will generalize this to the case where there may be more than two possible worlds.

Let T_1 and T_2 be transition matrices corresponding to two possible worlds of G. The personalized PageRank values $\vec{r_1}$ and $\vec{r_2}$ for T_1 and T_2 for seed set, S, are defined in Section 2.2 as

$$\vec{r_1} = \alpha\mathbf{T_1}\vec{r_1} + (1-\alpha)\vec{s}, \quad \text{and} \quad \vec{r_2} = \alpha\mathbf{T_2}\vec{r_2} + (1-\alpha)\vec{s},$$

where α is a *residual probability parameter* and $\vec{s}$ is a *re-seeding* vector such that if a node $v_i \in S$, then $\vec{s}[i] = \frac{1}{\|S\|}$ and $\vec{s}[i] = 0$, otherwise. It is easy to see that these two equations can be rewritten as follows to solve for $\vec{r_1}$ and $\vec{r_2}$:

$$\vec{r_1} = (1-\alpha)(I - \alpha T_1)^{-1}\vec{s} \quad \text{and} \quad \vec{r_2} = (1-\alpha)(I - \alpha T_2)^{-1}\vec{s}.$$

Given these, as defined in Section 3.3, we can compute the *expected* PPR values for the edge uncertain graph as

$$\vec{r} = \frac{1}{2}(\vec{r_1} + \vec{r_2}) = \frac{1-\alpha}{2}\left((I - \alpha T_1)^{-1} + (I - \alpha T_2)^{-1}\right)\vec{s}.$$

Let us split both T_1 and T_2 into three parts:

$$T_1 = T_{BL} + T_X + P_1 \text{ and } T_2 = T_{BL} + T_X + P_2,$$

where $T_{BL} + T_X$ corresponds to the certain parts of the graph and P_1 and P_2 correspond to the uncertain edges in the two possible worlds. Let T_{BL} be the block-diagonal matrix, obtained by partitioning the graph into blocks (for example using *METIS* [15]), and T_X represent (certain) transitions across these partitions.

Note that, in general, we have $|T_{BL}| \gg |T_X|$. As we will see shortly, in this section, we further assume[5] that $|T_X| \gg |P_1|$ and $|T_X| \gg |P_2|$. As proposed in [28], assuming that the blocks are sufficiently small, we can efficiently compute $Q_{BL}^{-1} = (I - \alpha T_{BL})^{-1}$ by first computing the inverse matrices of each block and then combining these inverse matrices to obtain Q_{BL}^{-1}, which itself is in block-diagonal form. Moreover, since T_X, P_1, and P_2 are all sparse, we can also efficiently decompose the $T_X + P_1$ and $T_X + P_2$ into

$$T_X + P_1 \simeq U_1 S_1 V_1 \text{ and } T_X + P_2 \simeq U_2 S_2 V_2, \qquad (1)$$

[5] While this is a common assumption in related work [2], in Section 5.5, we discuss how to relax this assumption in cases where the number of uncertain edges involved in each possible world is large.

using a sparse approximate decomposition algorithm, such as [3]. Given these, we can rewrite $\overrightarrow{r} = \overrightarrow{r} = \frac{1}{2}(\overrightarrow{r_1} + \overrightarrow{r_2})$ as

$$\simeq \frac{1-\alpha}{2}\left(\Big(I-\alpha(T_{BL}+U_1S_1V_1)\Big)^{-1} + \Big(I-\alpha(T_{BL}+U_2S_2V_2)\Big)^{-1}\right)\overrightarrow{s}.$$

Then, by applying the well-known Sherman-Morrison lemma [25] on the term $(I-\alpha(T_{BL}+U_iS_iV_i))^{-1}$, we can reformulate the above equation to obtain[6]

$$\overrightarrow{r} \simeq \frac{1-\alpha}{2}\Big(Q_{BL}^{-1} + \alpha Q_{BL}^{-1}U_1(S_1^{-1} - \alpha V_1 Q_{BL}^{-1}U_1)^{-1}V_1Q_{BL}^{-1} + \\ Q_{BL}^{-1} + \alpha Q_{BL}^{-1}U_2(S_2^{-1} - \alpha V_2 Q_{BL}^{-1}U_2)^{-1}V_2Q_{BL}^{-1}\Big)\overrightarrow{s}.$$

When we further apply the Sherman-Morrison lemma on the term $(S_1^{-1} - \alpha V_1 Q_{BL}^{-1} U_1)^{-1}$ in the above equation, we obtain

$$(1-\alpha)Q_{BL}^{-1}\overrightarrow{s} \\ + \frac{\alpha(1-\alpha)}{2}Q_{BL}^{-1}\Big(U_1(S_1 + \alpha S_1V_1(Q_{BL} - \alpha U_1S_1V_1)^{-1}U_1S_1)V_1 \\ + U_2(S_2 + \alpha S_2V_2(Q_{BL} - \alpha U_2S_2V_2)^{-1}U_2S_2)V_2\Big)Q_{BL}^{-1}\overrightarrow{s}.$$

This equation can be simplified by introducing the terms $M_1 = U_1S_1V_1$ and $M_2 = U_2S_2V_2$ (where $M_1 \simeq T_X + P_1$ and $M_2 \simeq T_X + P_2$):

$$\overrightarrow{r} \simeq (1-\alpha)\left(I + \frac{\alpha}{2}Q_{BL}^{-1}\Big((M_1+M_2) + \alpha\big(M_1(Q_{BL} - \alpha M_1)^{-1}M_1 \\ + M_2(Q_{BL} - \alpha M_2)^{-1}M_2\big)\Big)\right)Q_{BL}^{-1}\overrightarrow{s}. \tag{2}$$

Relying on the assumption that $|T_{BL}| \gg |T_X| + |P_1|$ and $|T_{BL}| \gg |T_X| + |P_2|$, we can ignore the terms αM_1 and αM_2 in $(Q_{BL} - \alpha M_1)^{-1}$ and $(Q_{BL} - \alpha M_2)^{-1}$ in the above equation and rewrite the rest as

$$\overrightarrow{r} \simeq (1-\alpha)\left(I + \frac{\alpha}{2}Q_{BL}^{-1}\Big((2T_X + P_1 + P_2) + \alpha\big(2T_XQ_{BL}^{-1}T_X \\ + (P_1+P_2)Q_{BL}^{-1}T_X + T_XQ_{BL}^{-1}(P_1+P_2) \\ + P_1Q_{BL}^{-1}P_1 + P_2Q_{BL}^{-1}P_2\big)\Big)\right)Q_{BL}^{-1}\overrightarrow{s}. \tag{3}$$

Furthermore, again relying on the assumption that $|T_{BL}| \gg |T_X| \gg |P_1|, |P_2|$, the term $P_1Q_{BL}^{-1}P_1 + P_2Q_{BL}^{-1}P_2$ will be negligible next to $(P_1+P_2)Q_{BL}^{-1}T_X + T_XQ_{BL}^{-1}(P_1+P_2)$ and thus can be ignored and $\overrightarrow{r}$ can be approximately computed as

$$(1-\alpha)\left(I + \frac{\alpha}{2}Q_{BL}^{-1}\Big((2T_X + (P_1+P_2)) + \alpha\big(2T_XQ_{BL}^{-1}T_X + \\ (P_1+P_2)Q_{BL}^{-1}T_X + T_XQ_{BL}^{-1}(P_1+P_2)\big)\Big)\right)Q_{BL}^{-1}\overrightarrow{s}. \tag{4}$$

Summary and Key Advantages: First of all, assuming that the blocks are sufficiently small and Q_{BL}^{-1} can be efficiently computed, once Q_{BL}^{-1} is at hand, solving for $\overrightarrow{r}$ using the above equation involves very sparse matrix multiplications (involving T_X and $P_1 + P_2$) and thus can be processed very efficiently (see Section 6). A second advantage of the above formulation is that it can be easily extended to any number of possible worlds.

5.2 General Case: > 2 Possible Worlds

When we have n possible worlds (i.e., $\overrightarrow{r} = \frac{1}{n}(\overrightarrow{r_1} + \ldots + \overrightarrow{r_n})$), the UPPR equation (Equation 4) can be generalized as

$$\simeq (1-\alpha)\left(I + \frac{\alpha}{n}Q_{BL}^{-1}\Big((nT_X + (P_1 + \ldots + P_n)) + \alpha\big(nT_XQ_{BL}^{-1}T_X \\ + (P_1 + \ldots + P_n)Q_{BL}^{-1}T_X + T_XQ_{BL}^{-1}(P_1 + \ldots + P_n)\big)\Big)\right)Q_{BL}^{-1}\overrightarrow{s}. \tag{5}$$

As we see in Section 6, this formulation leads to efficient execution plans, especially because the term $\frac{1}{n}(P_1 + \ldots + P_n)$ in Equation 5 can be obtained (without having to enumerate all possible worlds) directly by computing the ratio of the number of possible worlds in which a given edge exists.

Under *mutual exclusion* semantics: As we have seen in Section 3.1, the possible worlds covered by an uncertain edge consist of all combinations of its target nodes. Under mutual exclusion semantics, only one of the edges implied by the uncertain edge can be valid in the real world. Let v_i be a node which has c outgoing certain edges and u outgoing uncertain edges. If, in a given possible world, some of the u outgoing uncertain edges map to ϵ, then in that possible world, the transition probabilities for the remaining certain and uncertain edges will be higher. We can use this observation to compute $P_{avg} = \frac{1}{n}(P_1 + \ldots + P_n)$ as follows:

Let v_j be a target node of an uncertain edge, e_-, with $\|target(e_-)\| = k$. The value of $P_{avg}(j,i)$ can be computed as[7]

$$\frac{1}{k}\times\Big(\sum_{h=0}^{u-1}\Big(\frac{1}{c+u-h}\Big)(\text{ratio of worlds s.t. } h \text{ of other unc.edges are } \epsilon)\Big).$$

Here, $p()$ denotes the probability of a given event.

Note that, if e_- has ϵ as a target, then the corresponding transition probability has to be redistributed among the outgoing certain edges of the node and, if none exists, then it needs to be redistributed among all nodes in the graph. Let e_+ be an outgoing certain edge from v_i and let us denote its target as v_j. The transition probability, for e_+, taking into account ϵ transition for the uncertain edges, can be computed as

$$\sum_{h=0}^{u}\Big(\frac{1}{c+u-h}\Big)(\text{ratio of worlds s.t. } h \text{ of unc.edges are } \epsilon).$$

However, since e_+ is a certain edge, it belongs to either intra-partition or cross-partition certain edges. Therefore, when we compute the $P_{avg}(j,i)$, we need to compensate for the portion of the transition probability already accounted in T_{BL} or T_X. Let $C(j,i)$ denote $T_{BL}(j,i) + T_X(j,i)$; then, the cell $[j,i]$ in P_{avg} has the compensated value

$$\left(\sum_{h=0}^{u}\Big(\frac{1}{c+u-h}\Big)(\text{ratio of worlds s.t. } h \text{ of unc.edges are } \epsilon)\right) - C(j,i).$$

If v_i does not have any certain edges, the transition probability is distributed among all nodes in the graph. See [19] for details.

In both cases, to compute, P_{avg}, we need to compute the probability that for h out of a given number of uncertain edges, ϵ will

[6] For space considerations, we do not provide the complete sequence of algebraic operations in each step.

[7] Note that, if v_j is a target for multiple outgoing edges from v_i, all transition probabilities to v_j need to be aggregated (see [19]).

be selected as the target. Let us be given $m = (m_0 + m_1)$ uncertain edges, such that m_0 many do not contain ϵ in the target set and m_1 many do. Let the maximum target size for this latter set of nodes be max_target. Then, we can group the m_1 uncertain edges to max_target many groups where, each group, g_l, consists of uncertain edges with target size l; i.e., $\|g_1\| + \|g_2\| + \ldots + \|g_{max\_target}\| = m_1$. Note that, by definition, any uncertain edge which contains ϵ as a target must also have at least one other node in its target set, $\|g_1\| = 0$.

Given this, we can compute the probability that h out of m uncertain edges will be ϵ as

$$p(h_2 + h_3 + \ldots + h_{max\_target} = h \quad s.t. \quad \forall_{2 \leq l \leq max\_target} \; h_l \text{ in } \|g_l\| \text{ edges select } \epsilon).$$

The probability $p(h_l$ in $\|g_l\|$ edges select $\epsilon)$ is binomially distributed with $B(\|g_l\|, 1/l)$ – i.e., there are $\|g_l\|$ uncertain edges, each serving as an independent trial with $1/l$ success rate for the selection of ϵ among the available targets. Consequently, the probability that h out of m uncertain edges select ϵ as their targets is distributed as a summation of the binomial distributions $B(\|g_2\|, 1/2) + \ldots + B(\|g_{max\_target}\|, 1/max\_target)$. Algorithms to efficiently compute summation of binomial distributions are presented in [4]. See [19] for details.

Under *multiple edge* semantics: In this case, several of the edges implied by a given uncertain edge can be simultaneously valid. Let v_i be a node with c outgoing certain edges and u outgoing uncertain edges. Let v_j be a target node of an outgoing edge, e, from v_i. The value of $P_{avg}(j, i)$ can be computed as[8]

$$\sum_{h=0}^{total\_out} \left(\frac{1}{c+h}\right) \times p\left(\sum_{e \in U} num\_selected\_target\_nodes(e) = h\right),$$

where $total\_out = \sum_{e \in U} \|target(e)/\{\epsilon\}\|$ and $num\_selected\_target\_nodes(e)$ is the number of nodes selected as outgoing targets for e in a given possible world (if ϵ is the only target selected, then $num\_selected\_target\_nodes(e) = 0$).

Note that, similarly with the case of mutual exclusion semantics, for certain edges, we need to compensate for transition probabilities already accounted in T_{BL} or T_X. Also, if v_i does not have any certain edges, the transition probability for the case where all uncertain edges select ϵ as target needs to be distributed among all nodes. See [19] for details.

To compute P_{avg} using the above equation, we need to compute the probability $p\left(\sum_{e \in U} num\_selected\_target\_nodes(e) = h\right)$. Once again, this can be achieved by representing the distribution as a sum of binomial-like distributions: intuitively, if e is an uncertain edge with ϵ, then the probability that t many non-ϵ targets are selected can be represented in the form of a binomial with $\|target(e)\| - 1$ many trials and $1/2$ success rate. If, on the other hand, e is an uncertain edge without ϵ, the probability that t many targets are selected can be represented in the form of a binomial with $target(e)$ many trials and $1/2$ success rate. In the latter case, however, we need to correct for the situation where $t = 0$. This is because, under multiple edge semantics, for an uncertain edge without ϵ, the selected target nodes must include at least one node in the graph; thus, t cannot take the value of 0. See [19] for details.

5.3 Accuracy of UPPR

The UPPR equation (Equation 5) captures the underlying uncertainty in a way that leads to minimal approximation errors under the assumption $|T_{BL}| \gg |T_X| \gg |P_*|$. In particular, the UPPR process has three specific sources for potential errors, each of which is minimized under these, generally valid, assumptions:

The first source of error is the decomposition of $T_X + P_*$ into $U_* S_* V_*$ using an approximate algorithm, such as [3], that relies on the sparsity of the edges that cross partitions and of the uncertain edges (see Equation 1). The second source of error is the assumption that the terms αM_1 and αM_2 are negligible relative to the rest of the terms in Equation 2; this relies on the assumption that T_X and P_* that contribute to M_* are both sparse matrices. The third source of error is the assumption that the term $P_1 Q_{BL}^{-1} P_1 + P_2 Q_{BL}^{-1} P_2$ in Equation 3 is negligible relative to $(P_1 + P_2) Q_{BL}^{-1} T_X + T_X Q_{BL}^{-1} (P_1 + P_2)$.

Note that all three potential sources of error are minimized when $|T_{BL}| \gg |T_X| \gg |P_*|$. While the fact that whether $|T_{BL}| \gg |T_X|$ holds or not depends on the type of graph and the partitioning algorithm used, whether $|T_X| \gg |P_*|$ or not depends on the amount of uncertain edges in the graph. In Section 5.5, we discuss how to relax the assumption, $|T_X| \gg |P_*|$, in cases where there are significant number of uncertain edges in the graph rendering $|P_*|$ relatively dense, using a hybrid strategy.

5.4 Efficient Computation of UPPR Scores

Let us partition Equation 5 into 6 subcomponents:

$$\begin{aligned}
\vec{r} = \frac{1}{n}(\vec{r_1} + \ldots + \vec{r_n}) \simeq\ & \underbrace{(1-\alpha) Q_{BL}^{-1} \vec{s}}_{(1)} + \underbrace{\alpha(1-\alpha) Q_{BL}^{-1} T_X Q_{BL}^{-1} \vec{s}}_{(2)} \\
& + \underbrace{\frac{\alpha(1-\alpha)}{n} Q_{BL}^{-1} (P_1 + \ldots + P_n) Q_{BL}^{-1} \vec{s}}_{(3)} \\
& + \underbrace{\alpha^2(1-\alpha) Q_{BL}^{-1} T_X Q_{BL}^{-1} T_X Q_{BL}^{-1} \vec{s}}_{(4)} \\
& + \underbrace{\frac{\alpha^2(1-\alpha)}{n} Q_{BL}^{-1} (P_1 + \ldots + P_n) Q_{BL}^{-1} T_X Q_{BL}^{-1} \vec{s}}_{(5)} \\
& + \underbrace{\frac{\alpha^2(1-\alpha)}{n} Q_{BL}^{-1} T_X Q_{BL}^{-1} (P_1 + \ldots + P_n) Q_{BL}^{-1} \vec{s}}_{(6)}.
\end{aligned}$$

Each of the six subcomponents above contains an extremely sparse re-seeding vector $\vec{s}$. Moreover, Q_{BL}^{-1} is a block diagonal matrix and T_X and P_* are all sparse. Consequently, each of the terms can be computed, right to left, through efficient vector-matrix multiplications. For example, the subcomponent (2) can be computed from right to left with the following sequence of efficient operations:

$$\underbrace{Q_{BL}^{-1}}_{|V|\times|V|} \underbrace{\vec{s}}_{|V|\times 1} \rightarrow \underbrace{T_X}_{|V|\times|V|} \underbrace{Q_{BL}^{-1}\vec{s}}_{|V|\times 1} \rightarrow \underbrace{Q_{BL}^{-1}}_{|V|\times|V|} \underbrace{T_X Q_{BL}^{-1}\vec{s}}_{|V|\times 1}$$

$$\rightarrow \alpha(1-\alpha) \underbrace{Q_{BL}^{-1} T_X Q_{BL}^{-1} \vec{s}}_{|V|\times 1}.$$

[8] Again, all v_i to v_j transitions need to be aggregated.

Data	# of nodes	# of edges	# of partitions
ego-Facebook	4,039	88,234	3
Wiki-Vote	7,115	103,689	3
web-NotreDame	325,729	1,497,134	50
web-BerkStan	685,230	7,600,595	500

Table 1: Data sets

	# of uncertain edges	degree of edge uncertainty	edge semantics	# of possible worlds
different # of uncertain edges	2	4	mut.excl. (multiple)	16-64
	4			256-4,096
	6			4,096-262,144
	8(7)			65,536-2,097,152
different degree of edge uncertainty	4	2	mut.excl. (multiple)	16-16
		4		256-4,096
		6(5)		1,296-65,536
		8(6)		4,096-1,048,576
		10		10,000

Table 2: Uncertainty scenarios

Moreover, since the terms $(P_1 + ... + P_n)$, $Q_{BL}^{-1}\overrightarrow{s}$, $T_X Q_{BL}^{-1}\overrightarrow{s}$, and $Q_{BL}^{-1} T_X Q_{BL}^{-1}\overrightarrow{s}$ occur in multiple subcomponents, they can be cached and reused – once these terms are cached, the rest of the computations for the six subcomponents can be executed in parallel. Note further that several of the terms above can be cached and reused for the same uncertain graph with different seed vectors or even graphs with the same certain, but different uncertain components (to carry out hypothetical, if-then analyses).

5.5 Hybrid Computation in the Presence of Large Numbers of Uncertain Edges

As we have discussed in the previous section, the accuracy of the proposed UPPR technique relies on the assumption that $|T_{BL}| \gg |T_X| \gg |P_*|$. In particular, whether $|T_X| \gg |P_*|$ or not depends on the amount of uncertain edges in the graph: UPPR is likely to be highly effective and efficient if the number of uncertain edges in the graph is relatively small. In contrast, as we have seen in Section 4.2, the collapsing (and similarly flattening) based techniques may lead to large errors if the uncertain edges are concentrated around nodes with large PPR scores. We can leverage these two observations to deal with graphs with large numbers of uncertain edges: The idea is to eliminate uncertain edges in the graph, relying on the highly efficient *flattening* technique, away from the seed nodes of the graph (which are likely to have large PPR scores) and only maintain uncertain edges in the neighborhoods of the seed nodes. Consequently, errors due to flattening are minimized as this technique is utilized only in regions with less likelihood of producing high PPR scores; UPPR errors are also minimized, especially in large graphs, as the numbers ($|P_*|$) of uncertain edges in possible worlds that UPPR has to deal with have been reduced relative to the rest of the graph.

6 EXPERIMENTS

6.1 Datasets and Setup

We ran experiments on a 16-core CPU Nehalem Node with 64 GB RAM. All codes were implemented in Matlab and run using Matlab R2013b. Table 1 provides an overview of the four data sets [21], with different numbers of nodes and edges, and graph partitions, considered in the experiments (the partitions are obtained using METIS [15]). Table 2 details the volumes of uncertainty we have experimented with for the results reported in this section. Here, the "*degree of uncertainty*" refers to the number of target nodes on each uncertain edge it represents and the "*edge semantics*" describes "mutual exclusion" and "multiple edge" semantics. These together define the number of possible worlds corresponding to a given uncertain edge. To obtain uncertain graphs with the specifications in the table, we select random edges in the original graph and render them uncertain by augmenting destinations with random nodes. We further assume that the uncertain edges are located on the seeds (as discussed in Sections 4.2 and 5.5, uncertain edges away from the seeds can be flattened into certain edges).

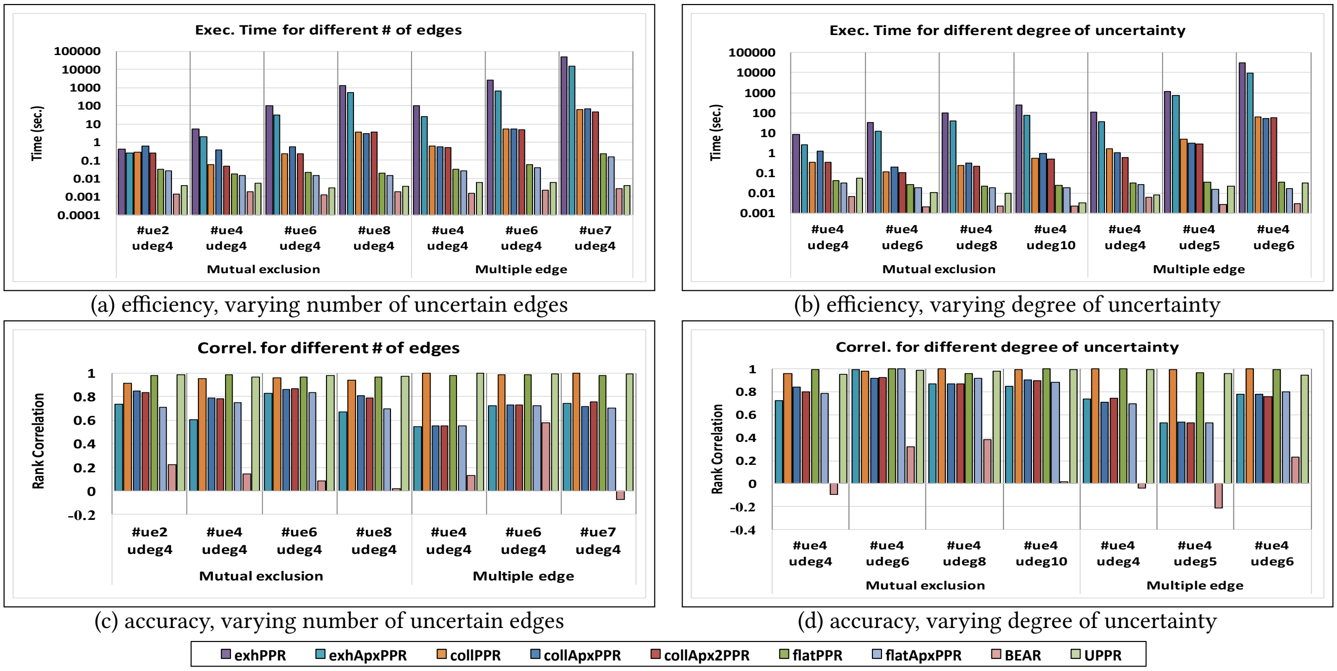
6.2 Alternative Approaches

In this section, in addition to UPPR (presented in Section 5), we considered all alternative approaches discussed in Section 4. As a further baseline, we also consider a Monte Carlo-based solution (which starts from the seed nodes, and samples random walks of a given length) and *BEAR* [27], a recent PPR computation algorithm, which originally does not take uncertainty into account. For uncertainty, we use the flattened transition matrix for the transition matrix and compute PPR values. In the experiments, without loss of generality, we set the residual probability parameter, α to 0.85. To compare different algorithms, we consider both efficiency (i.e., PPR computation time) and accuracy (in terms of the correlations of PPR rankings for the nodes that are ranked top-50 by the exhaustive technique, exhPPR).

7 RESULTS AND DISCUSSIONS

We start the discussion of the results by considering efficiency and accuracy of the various algorithms on the Facebook data set, for different degrees of uncertainty in the graph.

Impact of the Degree of Uncertainty. Figures 5(a) and (b) show the execution times of different algorithms, as the overall number of uncertain edges and degree of uncertainty in the graph are increased. As we see in the figure 5, exhaustive and collapsing-based approaches (which need to enumerate the possible worlds) quickly become infeasible as the number of possible worlds increases. While flattening-based approaches are reasonably fast and scale better than the exhaustive and collapsing-based approaches, they are 1 or 2 order slower than UPPR. *BEAR* takes less time than *UPPR* for PPR computation but the difference between them is negligible. Figures 5(c) and (d) confirm that execution time savings on *UPPR* do not come with any drop in accuracy – *UPPR* provides similar (or in some cases better) accuracy to the two collapsing- and flattening-based approaches, collPPR and flatPPR, that rely on direct computation of PPR from the transition matrix, even though it uses an approximate solution for PPR. As expected, the accuracy of *BEAR* is very poor compared to *UPPR* and the accuracy is not stable and affected by the amount of uncertainty. Other techniques such as collApxPPR, collApx2PPR, and flatApxPPR that similarly solve PPR approximately, relying on a sparse approximation method, all have significantly degraded accuracies. This indicates that, by carefully accounting for the sources of errors, UPPR is able to achieve high



(a) efficiency, varying number of uncertain edges

(b) efficiency, varying degree of uncertainty

(c) accuracy, varying number of uncertain edges

(d) accuracy, varying degree of uncertainty

Figure 5: Results on the Facebook data set, for different amount of uncertainty with different edge semantics: UPPR provides almost perfect accuracy and its execution time is not affected by the amount of uncertainty

accuracies (~1.0) efficiently (~0.01 seconds) and avoids accuracy pitfalls that other schemes are not able to handle effectively.

UPPR vs. Monte Carlo Method. Additionally, we consider a Monte Carlo (MC) based alternative to UPPR. [22] notes that (in regular graphs) for estimating PPR values close to a desired threshold δ (where δ is the expected PPR score; i.e., $1/|V|$, where $|V|$ is the number of nodes), a Monte Carlo based algorithm would need $O(1/(\delta \times \rho^2)) = O(|V|/\rho^2)$, samples of length, $geometric(\frac{1}{1-\alpha})$, where ρ is the relative error and $1-\alpha$ is the teleportation rate. This means that, when we seek high accuracy, Monte Carlo based solutions may be *prohibitive* [22]. Indeed, for the Facebook data set, with ~ 4000 nodes, to have 95% accuracy, we would need $4000/0.05^2 = 1,600,000$ random walk samples (of length $\geq \lceil \frac{1}{0.15} \rceil = 7$, since we set α to 0.85).

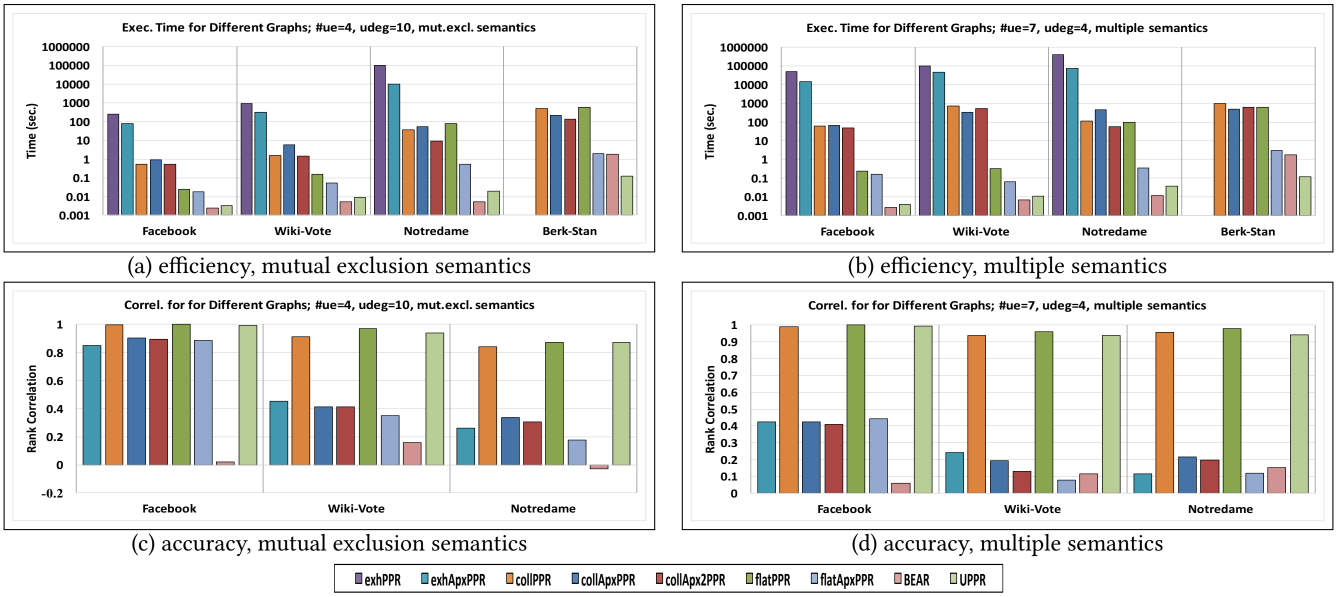
In Table 3, we report the accuracy comparison for a more modest target error rate of 0.15, which leads to ~ $150K$, random walks – note that, even in this modest case, taking $150K$ random walk samples is more expensive (65 seconds in Matlab) to compute than UPPR (~0.01 seconds). In the table, we see that for top-100 to top-500 results, Monte Carlo, is able to match the target accuracy in the presence if mutual exclusion semantics; but fails to do so when all nodes are considered. In the presence of multiple edge semantics, MC is able to match the target error rate only when top-500 results are considered and the results are very poor for top-100 nodes, even with larger number of samples, with longer lengths. Note that UPPR is able to achieve significantly higher accuracy (for top-100, top-500, as well as for all nodes), very cheaply (~ 0.01 seconds for this data set as shown in Figure 5).

Different Data Sets and the Impact of the Graph Size. In the experiments reported in Figure 6, we compare the efficiency and effectiveness of the various algorithms we presented in the paper for graphs of different sizes. The figure reports results for two sample uncertainty complexities: Figures 6(a) and (c) report execution time and rank correlation for a scenario with mutual exclusion semantics, whereas Figures 6(b) and (d) consider a scenario with multiple edge semantics. As we see in this figure, the proposed UPPR method is scalable, not only in terms of the possible worlds of the graph, but also the graph size. While the closest algorithms to UPPR in terms of efficiency and scalability, flatApxPPR and BEAR, suffer significantly from accuracy degradations, UPPR provides very high (mostly close to perfect) accuracy in all cases considered in this section.

Here, we do not present the accuracy results for the largest Berk-Stan data set as the cost of performing the exhaustive enumeration needed to obtain the accuracy ground-truth is prohibitive on this data set. However, the results show that *UPPR* provides very good accuracy, while its execution time is minimally effected by graph size. In fact, on the largest data set, *UPPR* is even faster than the *BEAR* baseline, while providing significantly better accuracy.

8 CONCLUSIONS

In this paper, we presented an uncertain edge model with mutual exclusion and shown that, while there are several ways to naively extend existing personalized PageRank computation techniques to graphs with uncertain edges, these either lead to large degrees of errors or are very expensive to compute in practice. We therefore proposed a novel *Uncertain Personalized PageRank (UPPR)* algorithm to approximately compute personalized PageRank values on such graphs. Experiments confirmed that the proposed technique has very high accuracy and is multiple-orders faster than available algorithms that can provide comparable accuracy.



(a) efficiency, mutual exclusion semantics

(b) efficiency, multiple semantics

(c) accuracy, mutual exclusion semantics

(d) accuracy, multiple semantics

Figure 6: Results in graphs of different sizes: as the figures show, UPPR provides good accuracy and its execution time is minimally effected by the graph size

Edge type	# of random walks	Length of random walks	Top 100 acc.	Top 500 acc.	All nodes acc.
Mutual exclusion semantics (#ue=4, #udeg=10)	UPPR		**0.952**	**0.981**	**0.997**
	150K Monte Carlo	8	0.782	0.881	0.525
		10	0.816	0.919	0.583
		20	0.841	0.920	0.519
		30	0.834	0.908	0.533
	300K Monte Carlo	8	0.814	0.911	0.584
		10	0.845	0.927	0.545
		20	0.858	0.924	0.588
		30	0.813	0.913	0.571
Multiple edge semantics (#ue=7, #udeg=4)	UPPR		**0.998**	**0.989**	**0.997**
	150K Monte Carlo	8	0.193	0.878	0.571
		10	0.145	0.900	0.656
		20	0.258	0.969	0.658
		30	0.269	0.937	0.696
	300K Monte Carlo	8	0.193	0.945	0.649
		10	0.163	0.912	0.667
		20	0.148	0.905	0.660
		30	0.155	0.901	0.670

Table 3: UPPR vs. MC method on the Facebook graph

REFERENCES

[1] S. Adali, M.L. Sapino, and B. Marshall, A rank algebra to support multimedia mining applications, MDM'07, 2007

[2] P. Boldi, F. Bonchi, A. Gionis, and T. Tassa. Injecting Uncertainty in Graphs for Identity Obfuscation, PVLDB'12, 2012.

[3] M. Brand. Fast online svd revisions for lightweight recommender systems. SDM, pp 37–46. SIAM, 2003.

[4] K. Butler and M. Stephens. The distribution of a sum of binomial random variables, Stanford University CA Dept of Statistics, 1993.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. WWW, 1998.

[6] K.S. Candan and W.S. Li, Using random walks for mining web document associations. PAKDD'00, pp.37–46, 2000.

[7] K.S. Candan, J. Grant, and V.S.Subrahmanian, A Unified Treatment of Null Values Using Constraints, Information Sciences, 98 (1-4) (1997), pp. 99–156

[8] H. Cao, K.S. Candan, and M.L. Sapino. Skynets: Searching for Minimum Trees in Graphs with Incomparable Edge Weights. CIKM'11, 2011.

[9] S. Chakrabarti. Dynamic personalized pagerank in entity-relation graphs. WWW'07, 571–580, 2007.

[10] J.B. Collins and S.T. Smith. Network Discovery For Uncertain Graphs. Fusion'14, 2014.

[11] C. De Kerchove, L. Ninove, and P. Van Dooren. Maximizing pagerank via outlinks. Linear Algebra and its Applications, 429(5):1254–1276, 2008.

[12] L. Du, C. Li, H. Chen, L. Tan, and Y. Zhang. Probabilistic simrank computation over uncertain graphs. Information Sciences. 295:521–535, 2015.

[13] H. Ishii and R. Tempo. Computing the pagerank variation for fragile web data. SICE Journal of Control, Measurement, and System Integration, 2(1):1–9, 2009.

[14] G. Jeh and J. Widom. SimRank: A Measure of Structural-Context Similarity. SIGKDD'02, pages 538–543, 2002.

[15] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing, 1998.

[16] A. Khan, F. Bonchi, A. Gionis, and F. Gullo. Fast Reliability Search in Uncertain Graphs. EDBT'14, 2014.

[17] A. Khan and L. Chen. On Uncertain Graphs Modeling and Queries. VLDB'15, pp. 2042–2043, 2015.

[18] J.H. Kim, K.S. Candan, M.Luisa. Sapino. Locality-sensitive and Re-use Promoting Personalized PageRank computations. Knowl. Inf. Syst. 47(2): 261-299 (2016)

[19] J.H. Kim. Efficient Node Proximity and Node Significance Computations in Graphs. PhD Thesis. Arizona State University, 2017.

[20] R.-H. Li, J.X. Yu, R. Mao, and T. Jin. Efficient and Accurate Query Evaluation on Uncertain Graphs via Recursive Stratified Sampling. ICDE'14, 2014.

[21] J. Leskovec and A. Krevl. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. 2014.

[22] P. Lofgren. Efficient Algorithms for Personalized PageRank. PhD Thesis, Stanford University. 2015.

[23] X. Niu, L. Li, and K. Xu. Digrank: Using global degree to facilitate ranking in an incomplete graph. CIKM'11, pp 2297–2300. 2011.

[24] D.L. Nowell and J. Kleinberg. The Link Prediction Problem for Social Networks. CIKM'03, 2003.

[25] W. W. Piegorsch and G. Casella. Erratum: inverting a sum of matrices. SIAM review, 32(3):470, 1990.

[26] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. K-nearest neighbors in uncertain graphs. VLDB'10, 3(1-2):997–1008, 2010.

[27] K. Shin, J. Jung, L. Sael, and U Kang. BEAR: Block Elimination Approach for Random Walk with Restart on Large Graphs. SIGMOD'15, 2015.

[28] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. ICDM, pp. 613–622, 2006.

[29] Y. Yuan, L. Chen, G. Wang. Efficiently Answering Probability Threshold-Based Shortest Path Queries over Uncertain Graphs, DASFAA'10, 2010.

[30] Y. Yuan, G. Wang, H. Wang, and L. Chen. Efficient Subgraph Search over Large Uncertain Graphs. PVLDB, 4(11), 2011.