Under consideration for publication in Knowledge and Information Systems

Ranking in Signed Social Networks: Model and Algorithms

Jinhong Jung*, Woojeong Jin+, U Kang*[1], and Lee Sael†[2]

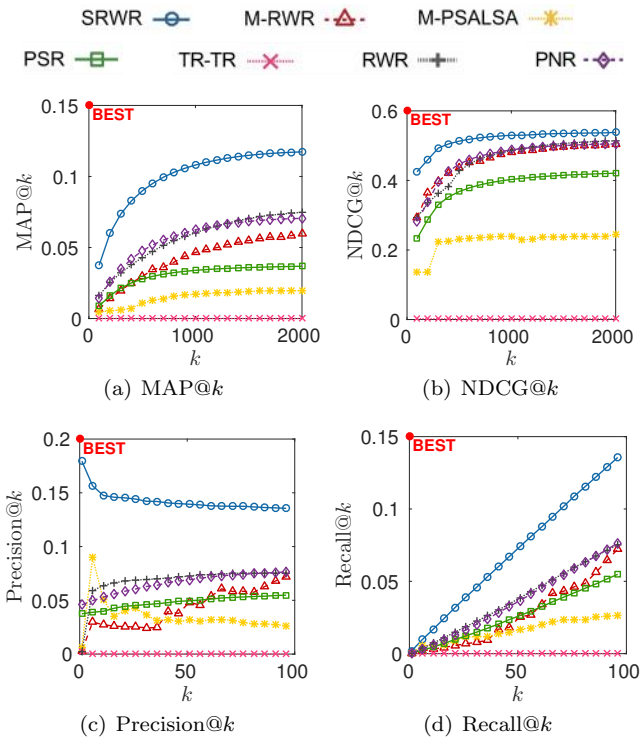
*Department of Computer Science and Engineering, Seoul National University
+Department of Electrical and Computer Engineering, Seoul National University
†Department of Computer Science, The State University of New York (SUNY) Korea

Abstract. How can we rank nodes in signed social networks? Relationships between nodes in a signed network are represented as positive (trust) or negative (distrust) edges. Many social networks have adopted signed networks to express trust between users. Consequently, ranking friends or enemies in signed networks has received much attention from the data mining community. The ranking problem, however, is challenging because it is difficult to interpret negative edges. Traditional random walk based methods such as PageRank and Random Walk with Restart cannot provide effective rankings in signed networks since they assume only positive edges. Although several methods have been proposed by modifying traditional ranking models, they also fail to account for proper rankings due to the lack of ability to consider complex edge relations. In this paper, we propose Signed Random Walk with Restart (SRWR), a novel model for personalized ranking in signed networks. We introduce a signed random surfer so that she considers negative edges by changing her sign for walking. Our model provides proper rankings reflecting signed edges based on the signed random walk. We develop two methods for computing SRWR: SRWR-Iter and SRWR-Pre which are iterative and preprocessing methods, respectively. SRWR-Iter naturally follows the definition of SRWR, and iteratively updates SRWR scores until convergence. SRWR-Pre enables fast ranking computation which is important for the performance of applications of SRWR. Through extensive experiments, we demonstrate that SRWR achieves the best accuracy (up to 87%) for sign prediction, and predicts trolls 4× more accurately than other ranking models. In terms of efficiency, SRWR-Pre preprocesses a signed network 4.5× faster, and requires 11× less memory space than other preprocessing methods; furthermore, SRWR-Pre computes SRWR scores up to 14× faster than other methods in the query phase.

Introduction

How can we obtain personalized rankings for users in signed networks? In many social networks, users are allowed to make opinions to indicate their trust or distrust to other's opinions. The users are represented as nodes, and the expressions are represented as positive and negative edges in graphs which are

[1] Co-corresponding Author
[2] Co-corresponding Author



(a) MAP@k (b) NDCG@k

(c) Precision@k (d) Recall@k

Fig. 1. The performance of ranking models for the troll identification task through various measurements: MAP@k (1(a)), NDCG@k (1(b)), Precision@k (1(c)), and Recall@k (1(d)). SRWR shows the best performance for all the measurements compared to other competitors.

called *signed networks* (Szell, Lambiotte and Thurner, 2010). Ranking nodes in signed networks has received much interest from data mining community to reveal trust and distrust between users (Kunegis, Lommatzsch and Bauckhage, 2009) inducing many useful applications such as link prediction (Leskovec, Huttenlocher and Kleinberg, 2010) and community detection (Yang, Cheung and Liu, 2007) in signed networks.

Traditional ranking models, however, do not provide satisfactory node rankings in signed networks. Existing random walk based ranking models such as PageRank (Page, Brin, Motwani and Winograd, 1999) and Random Walk with Restart (Tong, Faloutsos, Gallagher and Eliassi-Rad, 2007; Shin, Jung, Lee and Kang, 2015) assume only positive edges; thus, they are inappropriate in the networks containing negative edges. Many researchers have proposed heuristics on the classical methods to make them computable in signed networks (Kunegis et al., 2009; Shahriari and Jalili, 2014). However, the approaches are unsatisfactory because they cannot consider complex relationships of consecutive edges such as friend-of-enemy or enemy-of-enemy. This problem results from the non-interpretable negative edges in traditional random walks. In addition, most existing ranking models in signed networks focus only on global rankings, although personalized rankings are more useful for individuals in many contexts such as recommendation. Also, the fast ranking computation is important for the performance of applications in SRWR.

Table 1. Table of symbols. Boldface capital letters, such as $\mathbf{A}$, represent matrices. Boldface small letters, such as $\mathbf{r}$, represent vectors.

Symbol	Definition
G	signed input graph
n	number of nodes in G
n_1	number of spokes in G
n_2	number of hubs in G
m	number of edges in G
s	seed node (= query node, source node)
c	restart probability
ϵ	error tolerance
$\overleftarrow{\mathbf{N}}_u$	set of in-neighbors to nodes u
$\overrightarrow{\mathbf{N}}_u$	set of out-neighbors from nodes u
$\mathbf{A}$	$(n \times n)$ signed adjacency matrix of G
$\lvert\mathbf{A}\rvert$	$(n \times n)$ absolute adjacency matrix of G
$\mathbf{D}$	$(n \times n)$ out-degree matrix of $\lvert\mathbf{A}\rvert$, $\mathbf{D}_{ii} = \sum_j \lvert\mathbf{A}\rvert_{ij}$
$\tilde{\mathbf{A}}$	$(n \times n)$ semi-row normalized matrix of $\mathbf{A}$
$\tilde{\mathbf{A}}_+$	$(n \times n)$ positive semi-row normalized matrix of $\mathbf{A}$
$\tilde{\mathbf{A}}_-$	$(n \times n)$ negative semi-row normalized matrix of $\mathbf{A}$
$\lvert\tilde{\mathbf{A}}\rvert$	$(n \times n)$ absolute row-normalized matrix of $\lvert\mathbf{A}\rvert$
$\mathbf{q}$	$(n \times 1)$ starting vector (= s-th unit vector)
$\mathbf{r}^+$	$(n \times 1)$ trust SRWR score vector
$\mathbf{r}^-$	$(n \times 1)$ distrust SRWR score vector
$\mathbf{r}^d$	$(n \times 1)$ relative trustworthy vectors, $\mathbf{r}^d = \mathbf{r}^+ - \mathbf{r}^-$
$\mathbf{p}$	$(n \times 1)$ $\mathbf{p} = \mathbf{r}^+ + \mathbf{r}^-$
$\lvert\mathbf{H}\rvert$	$(n \times n)$ $\lvert\mathbf{H}\rvert = \mathbf{I} - (1-c)\lvert\tilde{\mathbf{A}}\rvert^\top$
$\mathbf{T}$	$(n \times n)$ $\mathbf{T} = \mathbf{I} - (1-c)(\gamma\tilde{\mathbf{A}}_+^\top - \beta\tilde{\mathbf{A}}_-^\top)$
$\lvert\mathbf{H}\rvert_{ij}$, $\mathbf{T}_{ij}$	$(n_i \times n_j)$ (i,j)-th partition of $\lvert\mathbf{H}\rvert$ or $\mathbf{T}$
$\mathbf{S}_{\lvert\mathbf{H}\rvert}$, $\mathbf{S}_\mathbf{T}$	$(n_2 \times n_2)$ Schur complement of $\lvert\mathbf{H}\rvert_{11}$ or $\mathbf{T}_{11}$
$\mathbf{q}_i$, $\mathbf{p}_i$, $\mathbf{r}_i^-$	$(n_i \times 1)$ i-th partition of $\mathbf{q}$, $\mathbf{p}$ or $\mathbf{r}^-$

In this paper, we propose Signed Random Walk with Restart (SRWR), a novel model for effective personalized rankings in signed networks. The main idea of SRWR is to introduce a sign into a random surfer in order to let the surfer consider negative edges. Consequently, our model considers complex edge relationships, and makes random walks interpretable in signed networks. We devise SRWR-Iter, an iterative method which naturally follows the definition of SRWR, and iteratively update SRWR scores until convergence. Furthermore, we propose SRWR-Pre, a preprocessing method for computing SRWR scores quickly. Through extensive experiments, we demonstrate the effectiveness and the efficiency of our proposed approaches. Our main contributions are as follows:

- **Novel ranking model.** We propose Signed Random Walk with Restart (SRWR), a novel model for personalized rankings in signed networks (Definition 1). We show that our model is a generalized version of RWR working on both signed and unsigned networks (Lemma 2).
- **Algorithm.** We propose SRWR-Iter and SRWR-Pre for computing SRWR scores. SRWR-Iter is an iterative algorithm which naturally follows the

definition of SRWR (Algorithm 2). SRWR-Pre is a preprocessing method which employs a node reordering technique and block elimination to accelerate SRWR computation speed (Algorithms 3 and 4).

– **Experiment.** We show that SRWR achieves the best accuracy (up to 87%) for sign prediction, and predicts trolls 4× more accurately than other ranking models (Figure 1). In terms of efficiency, SRWR-Pre preprocesses signed networks up to 4.5× faster, and requires 11× less memory space than baseline preprocessing methods. Furthermore, SRWR-Pre computes SRWR scores up to 14× faster than other methods including SRWR-Iter (Figure 10).

The code of our method and datasets used in this paper are available at http://datalab.snu.ac.kr/srwrpre. The rest of this paper is organized as follows. We discuss problems of ranking in signed networks and motivations of our model in Section 2. We describe our proposed model and algorithms for computing personalized rankings in Section 3. After presenting our experimental results in Section 4, we provide a review of related works in Section 5. Lastly, we conclude in Section 6. Table 1 lists the symbols used in this paper.

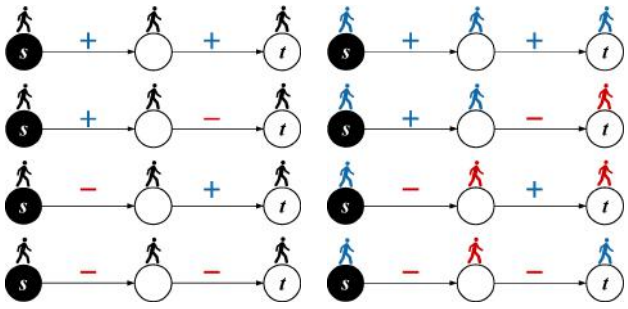
Motivation

Ranking nodes using the traditional ranking models in signed networks would be wrong or misleading because the algorithms assume only positive weights on edges in graphs. For example, random walk based ranking models such as PageRank or Random Walk with Restart (RWR) consider the weight of an edge as the likelihood that a random surfer chooses the edge to move. The higher weight the edge has, the greater chance the surfer has to choose the edge. In this context, a negative weight of an edge is not interpretable.

The existence of negative edges also makes the existing models unable to consider complex relationships between edges although several modifications of PageRank have been proposed in signed networks. For example, Shahriari et al. proposed Modified PageRank (Shahriari and Jalili, 2014) which computes PageRank scores separately on both positive and negative subgraphs. The PageRank score vector from the positive subgraph indicates a trust ranking, and that from the negative subgraph is a distrust ranking. However, it could provide misleading ranking results because the original graph becomes disconnected by splitting it into two subgraphs, thereby ignoring relationships between positive and negative edges.

As discussed in (Cartwright and Harary, 1956; Leskovec et al., 2010), structural balance theory provides how to interpret relationships of consecutive signed edges as follows: "the friend of my friend is my friend," "the friend of my enemy is my enemy," "the enemy of my friend is my enemy," and "the enemy of my enemy is my friend." This theory motivates us to design a new ranking model which has the capability of interpreting complicated edge relationships without dividing a signed network into positive and negative subgraphs for computing rankings.

On top of that, most existing ranking models in signed networks compute only a global ranking similarly to PageRank or HITS, even if a personalized ranking is more appropriate for recommending information reflecting tastes of a specific user. We are also concerned in efficiently computing personalized rankings since it is crucial to suggest personalized information within a limited time. Hence, our goal is to design an effective and efficient personalized ranking method for applications in signed networks.



(a) Traditional random walks (b) Signed random walks

Fig. 2. Examples of traditional random walks and signed random walks. Each case represents 1) friend's friend, 2) friend's enemy, 3) enemy's friend, or 4) enemy's enemy from the top. A random surfer has either a positive (blue) or a negative (red) sign on each node in Figure 2(b). When the signed random surfer traverses a negative edge, she changes her sign from positive to negative or vice versa.

Proposed Methods

We propose Signed Random Walk with Restart (SRWR), a novel ranking model in signed networks from Section 3.1 to Section 3.3. Then we first propose an iterative algorithm SRWR-Iter for computing SRWR scores w.r.t. a seed node in Section 3.4, and then propose a preprocessing algorithm SRWR-Pre to accelerate SRWR computation speed in Section 3.5.

Signed Random Walk with Restart Model

As discussed in Section 1, complicated relationships of signed edges are the main obstacles for providing effective rankings in signed networks. Most existing works on signed networks have not focused on personalized rankings. In this work, our goal is to design a novel ranking model which resolves those problems in signed networks. The main ideas of our model are as follows:

- We introduce a signed random surfer. The sign of the surfer is either positive or negative, which means favorable or adversarial to a node, respectively.
- When the random surfer encounters a negative edge, she changes her sign from positive to negative, or vice versa. Otherwise, she keeps her sign.
- We introduce balance attenuation factors into the surfer to consider the uncertainty for friendship of enemies.

There are four cases according to the signs of edges as shown in Figure 2: 1) friend's friend, 2) friend's enemy, 3) enemy's friend, and 4) enemy's enemy. Suppose a random surfer starts at node s toward node t. A traditional surfer just moves along the edges without considering signs as seen in Figure 2(a) since there is no way to consider the signs on the edges. Hence, classical models cannot distinguish those edge relationships during her walks. For instance, the model considers that node s and node t are friends for the second case (friend's enemy), even though node t are more likely to be an enemy w.r.t. node s.

On the contrary, our model in Figure 2(b) has a signed random surfer who considers those complex edge relationships. If the random surfer starting at node s with a positive sign encounters a negative edge, she flips her sign from positive

to negative, or vice versa. Our model distinguishes whether node t is the friend of node s or not according to her sign at node t. As shown in Figure 2(b), the results for all cases from our model are consistent with structural balance theory (Cartwright and Harary, 1956). Thus, introducing a signed random surfer enables our model to discriminate those edge relationships.

Trust or distrust relationships between a specific node s and other nodes are revealed as the surfer is allowed to move around a signed network starting from node s. If the positive surfer visits a certain node u many times, then node u is trustable for node s. On the other hand, if the negative surfer visits node u many times, then node s is not likely to trust node u. Thus, rankings are obtained by revealing a degree of trust or distrust between people based on the signed random walks. Here, we formally define our model on signed networks in Definition 1. Note that Definition 1 involves the concept of restart which provides personalized rankings w.r.t. a user.

Definition 1 (Signed Random Walk with Restart)**.** *A signed random surfer has a sign, which is either positive or negative. At the beginning, the surfer starts with + sign from a seed node s because she trusts s. Suppose the surfer is currently at node u, and c is the restart probability of the surfer. Then, she takes one of the following actions:*

- ***Action 1: Signed Random Walk.*** *The surfer randomly moves to one of the neighbors from node u with probability $1 - c$. The surfer flips her sign if she encounters a negative edge. Otherwise, she keeps her sign.*
- ***Action 2: Restart.*** *The surfer goes back to the seed node s with probability c. Her sign should become + at the seed node s because she trusts s.* ■

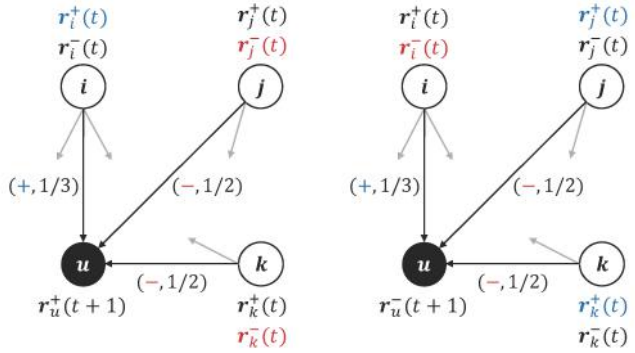
We measure two probabilities on each node through Signed Random Walk with Restart (SRWR) starting from the seed node s. The two probabilities are represented as follows:

- $\mathbf{r}_u^+ = \Pr_s(u, +)$: the probability that the positive surfer is at node u after SRWR from the seed node s.
- $\mathbf{r}_u^- = \Pr_s(u, -)$: the probability that the negative surfer is at node u after SRWR from the seed node s.

If $\mathbf{r}_u^+$ is high, then node u is trustable for node s. On the other hand, if $\mathbf{r}_u^-$ is high, node u is not reliable for node s. $\mathbf{r}^+$ is a trust SRWR score vector and $\mathbf{r}^-$ is a distrust SRWR score vector for all nodes. Both are used for personalized rankings w.r.t. the seed node s. If we regard $\mathbf{r}^+$ and $\mathbf{r}^-$ as score vectors, $\mathbf{r}^d = \mathbf{r}^+ - \mathbf{r}^-$ is considered as a relative trustworthiness vector of nodes w.r.t. s, which is also used as a personalized ranking. For instance, if $\mathbf{r}_u^d$ is positive, then node u is trustable for node s. Otherwise, node u is not trustable for node s.

Formulation for Signed Random Walk with Restart

We formulate the probability vectors, $\mathbf{r}^+$ and $\mathbf{r}^-$, following Signed Random Walk with Restart. First, we explain how to define $\mathbf{r}^+$ and $\mathbf{r}^-$ using the example shown in Figure 3. In the example, we label a (sign, transition probability) pair on each edge. For instance, the transition probability for the positive edge from node i to node u is $1/3$ because node i has 3 outgoing edges. This edge is denoted by $(+, 1/3)$. Other pairs of signs and transition probabilities are also similarly defined. In order that the random surfer has a positive sign on node u at time $t+1$, a positive surfer on one of u's neighbor at time t must move to node



(a) An example of a positive probability, $\mathbf{r}_u^+(t+1)$

(b) An example of a negative probability, $\mathbf{r}_u^-(t+1)$

Fig. 3. Examples of how $\mathbf{r}_u^+$ and $\mathbf{r}_u^-$ are defined in SRWR.

u through a positive edge, or a negative surfer must move through a negative edge according to the signed random walk action in Definition 1. Considering the restart action of the surfer with the probability c, $\mathbf{r}_u^+(t+1)$ in Figure 3(a) is represented as follows:

$$\mathbf{r}_u^+(t+1) = (1-c)\left(\frac{\mathbf{r}_i^+(t)}{3} + \frac{\mathbf{r}_j^-(t)}{2} + \frac{\mathbf{r}_k^-(t)}{2}\right) + c\mathbf{1}(u=s)$$

where $\mathbf{1}(u=s)$ is 1 if u is the seed node s and 0 otherwise. In Figure 3(b), $\mathbf{r}_u^-(t+1)$ is defined similarly as follows:

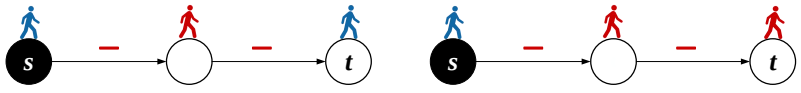
$$\mathbf{r}_u^-(t+1) = (1-c)\left(\frac{\mathbf{r}_i^-(t)}{3} + \frac{\mathbf{r}_j^+(t)}{2} + \frac{\mathbf{r}_k^+(t)}{2}\right)$$

Note that we do not add the restarting score $c\mathbf{1}(u=s)$ to $\mathbf{r}_u^-(t+1)$ in this case because the surfer's sign must become positive when she goes back to the seed node s. The recursive equations of our model are defined as follows:

$$\begin{aligned}\mathbf{r}_u^+ &= (1-c)\left(\sum_{v\in\overleftarrow{\mathbf{N}}_u^+}\frac{\mathbf{r}_v^+}{|\overrightarrow{\mathbf{N}}_v|} + \sum_{v\in\overleftarrow{\mathbf{N}}_u^-}\frac{\mathbf{r}_v^-}{|\overrightarrow{\mathbf{N}}_v|}\right) + c\mathbf{1}(u=s)\\ \mathbf{r}_u^- &= (1-c)\left(\sum_{v\in\overleftarrow{\mathbf{N}}_u^-}\frac{\mathbf{r}_v^+}{|\overrightarrow{\mathbf{N}}_v|} + \sum_{v\in\overleftarrow{\mathbf{N}}_u^+}\frac{\mathbf{r}_v^-}{|\overrightarrow{\mathbf{N}}_v|}\right)\end{aligned} \tag{1}$$

where $\overleftarrow{\mathbf{N}}_i$ is the set of in-neighbors of node i, and $\overrightarrow{\mathbf{N}}_i$ is the set of out-neighbors of node i. Superscripts of $\overleftarrow{\mathbf{N}}_i$ or $\overrightarrow{\mathbf{N}}_i$ indicate signs of edges between node i and its neighbors (e.g., $\overleftarrow{\mathbf{N}}_i^+$ indicates the set of positively connected in-neighbors of node i). We need to introduce several symbols related to an adjacency matrix $\mathbf{A}$ to vectorize Equation (1).

Definition 2 (Signed adjacency matrix)**.** *The signed adjacency matrix* $\mathbf{A}$ *of* G *is a matrix such that* $\mathbf{A}_{uv}$ *is positive or negative when there is a positive or a negative edge from node* u *to node* v *respectively, and zero otherwise.* ■



(a) The surfer's sign at node t is positive with β (b) The surfer's sign at node t is negative with $1-\beta$

(c) The surfer's sign at node t is negative with γ (d) The surfer's sign at node t is positive with $1-\gamma$

Fig. 4. Examples of balance attenuation factors. (a) and (b) represent the uncertainty of enemy-of-enemy with probability β, and (c) and (d) represent the uncertainty of friend-of-enemy with probability γ.

Definition 3 (Semi-row normalized matrix)**.** *Let* $|\mathbf{A}|$ *be the absolute adjacency matrix of* $\mathbf{A}$*, and* $\mathbf{D}$ *be the out-degree diagonal matrix of* $|\mathbf{A}|$ *(i.e.,* $\mathbf{D}_{ii} = \sum_j |\mathbf{A}|_{ij}$*). Then semi-row normalized matrix of* $\mathbf{A}$ *is* $\tilde{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A}$*.* ■

Definition 4 (Positive or negative semi-row normalized matrix)**.** *The positive semi-row normalized matrix* $\tilde{\mathbf{A}}_+$ *contains only positive values in the semi-row normalized matrix* $\tilde{\mathbf{A}}$*. The negative semi-row normalized matrix* $\tilde{\mathbf{A}}_-$ *contains absolute values of negative elements in* $\tilde{\mathbf{A}}$*. In other words,* $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}_+ - \tilde{\mathbf{A}}_-$*, and* $|\tilde{\mathbf{A}}| = \tilde{\mathbf{A}}_+ + \tilde{\mathbf{A}}_-$*.* ■

Based on Definitions 3 and 4, Equation (1) is represented as follows:

$$\begin{aligned} \mathbf{r}^+ &= (1-c)\left(\tilde{\mathbf{A}}_+^\top \mathbf{r}^+ + \tilde{\mathbf{A}}_-^\top \mathbf{r}^-\right) + c\mathbf{q} \\ \mathbf{r}^- &= (1-c)\left(\tilde{\mathbf{A}}_-^\top \mathbf{r}^+ + \tilde{\mathbf{A}}_+^\top \mathbf{r}^-\right) \end{aligned} \tag{2}$$

where $\mathbf{q}$ is a vector whose sth element is 1 and all other elements are 0.

Balance Attenuation Factors

The signed surfer measures trust and distrust of nodes w.r.t. a seed node according to edge relationships as discussed in Section 3.1. Our model strongly supports balance theory describing the four cases between nodes as shown in Figure 2(b). However, the naive balance theory would not hold for explaining behaviors of people in real-world signed networks, since unbalanced relationships frequently appear (e.g., the enemy of my friend could be my friend) due to the uncertainty in trusting the friendship of enemies.

We reflect the uncertainty of the friendship of an enemy into our ranking model by introducing stochastic parameters, β and γ, called *balance attenuation factors*. Note that we assume the friendship of a friendly user is reliable. β is a parameter for the uncertainty of "the enemy of my enemy is my friend", and γ is for "the friend of my enemy is my enemy." We first explain β using the fourth case (enemy's enemy) in Figure 2(b). Suppose a surfer with a positive sign starts at node s toward node t and encounters two consecutive negative edges. Based on balance theory, her sign becomes negative at the intermediate node m and

Algorithm 1: Normalization phase of SRWR-Iter

Input: signed adjacency matrix: $\mathbf{A}$
Output: positive semi-row normalized matrix: $\tilde{\mathbf{A}}_+$, and negative semi-row normalized matrix: $\tilde{\mathbf{A}}_-$

1: compute out-degree matrix $\mathbf{D}$ of $|\mathbf{A}|$, $\mathbf{D}_{ii} = \sum_j |\mathbf{A}|_{ij}$
2: compute semi-row normalized matrix, $\tilde{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A}$.
3: split $\tilde{\mathbf{A}}$ into $\tilde{\mathbf{A}}_+$ and $\tilde{\mathbf{A}}_-$ such that $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}_+ - \tilde{\mathbf{A}}_-$
4: **return** $\tilde{\mathbf{A}}_+$ and $\tilde{\mathbf{A}}_-$

Algorithm 2: Iteration phase of SRWR-Iter

Input: positive semi-row normalized matrix: $\tilde{\mathbf{A}}_+$, and negative semi-row normalized matrix: $\tilde{\mathbf{A}}_-$, and seed node: s, restart probability: c, balance attenuation factors: β and γ, and error tolerance: ϵ.
Output: positive SRWR scores: $\mathbf{r}^+$ and negative SRWR scores: $\mathbf{r}^-$

1: set the starting vector $\mathbf{q}$ from the seed node s
2: set $\mathbf{r}^+ = \mathbf{q}$, $\mathbf{r}^- = \mathbf{0}$, and $\mathbf{r}' = [\mathbf{r}^+; \mathbf{r}^-]$
3: **repeat**
4: $\quad \mathbf{r}^+ \leftarrow (1-c)(\tilde{\mathbf{A}}_+^\top \mathbf{r}^+ + \beta \tilde{\mathbf{A}}_-^\top \mathbf{r}^- + (1-\gamma)\tilde{\mathbf{A}}_+^\top \mathbf{r}^-) + c\mathbf{q}$
5: $\quad \mathbf{r}^- \leftarrow (1-c)(\tilde{\mathbf{A}}_-^\top \mathbf{r}^+ + \gamma \tilde{\mathbf{A}}_+^\top \mathbf{r}^- + (1-\beta)\tilde{\mathbf{A}}_-^\top \mathbf{r}^-)$
6: $\quad$ concatenate $\mathbf{r}^+$ and $\mathbf{r}^-$ into $\mathbf{r} = [\mathbf{r}^+; \mathbf{r}^-]^\top$
7: $\quad$ compute the error between $\mathbf{r}$ and $\mathbf{r}'$, $\delta = \|\mathbf{r} - \mathbf{r}'\|$
8: $\quad$ update $\mathbf{r}' \leftarrow \mathbf{r}$ for the next iteration
9: **until** $\delta < \epsilon$
10: **return** $\mathbf{r}^+$ and $\mathbf{r}^-$

positive at node t in Figure 4(a). However, some people might think that the enemy of my enemy is my enemy as shown in Figure 4(b). In this case, her sign will be negative at nodes m and t. To consider this uncertainty, we introduce a parameter β so that if the negative surfer at node m encounters a negative edge, her sign becomes positive with probability β or negative with $1-\beta$ at node t. The other parameter γ is also interpreted similarly to β. When the negative surfer at node m encounters a positive edge, her sign will be negative with probability γ or positive with $1-\gamma$ at node t as in Figures 4(c) and 4(d). SRWR with the balance attenuation factors is represented as follows:

$$\begin{aligned} \mathbf{r}^+ &= (1-c)\left(\tilde{\mathbf{A}}_+^\top \mathbf{r}^+ + \beta \tilde{\mathbf{A}}_-^\top \mathbf{r}^- + (1-\gamma)\tilde{\mathbf{A}}_+^\top \mathbf{r}^-\right) + c\mathbf{q} \\ \mathbf{r}^- &= (1-c)\left(\tilde{\mathbf{A}}_-^\top \mathbf{r}^+ + \gamma \tilde{\mathbf{A}}_+^\top \mathbf{r}^- + (1-\beta)\tilde{\mathbf{A}}_-^\top \mathbf{r}^-\right) \end{aligned} \tag{3}$$

Note that the uncertainty of a friend's friendship could be considered by adding other factors similarly to the proposed approach, but in this work, we only reflect the uncertainty of an enemy's friendship for simplicity.

SRWR-Iter: Iterative Algorithm for Signed Random Walk with Restart

We present an iterative algorithm SRWR-Iter for computing SRWR scores efficiently and accurately based on Equation (3).

Normalization phase (Algorithm 1). Our proposed algorithm first com-

putes the out-degree diagonal matrix $\mathbf{D}$ of $|\mathbf{A}|$, which is the absolute adjacency matrix of $\mathbf{A}$ (line 1). Then, the algorithm computes the semi-row normalized matrix $\tilde{\mathbf{A}}$ using $\mathbf{D}$ (line 2). We split $\tilde{\mathbf{A}}$ into two matrices: the positive semi-row normalized matrix ($\tilde{\mathbf{A}}_+$) and the negative semi-row normalized matrix ($\tilde{\mathbf{A}}_-$) (line 3) satisfying $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}_+ - \tilde{\mathbf{A}}_-$.

Iteration phase (Algorithm 2). Our algorithm computes the SRWR score vectors $\mathbf{r}^+$ and $\mathbf{r}^-$ for the seed node s with the balance attenuation factors (β and γ) in the iteration phase. We set $\mathbf{q}$ to s-th unit vector, and initialize $\mathbf{r}^+$ to $\mathbf{q}$ and $\mathbf{r}^-$ to $\mathbf{0}$ (lines 1 and 2). Our algorithm iteratively computes Equation (3) (lines 4 and 5). We concatenate $\mathbf{r}^+$ and $\mathbf{r}^-$ vertically (line 6) into $\mathbf{r}$. We then compute the error δ between $\mathbf{r}$ and $\mathbf{r}'$ which is the result in the previous iteration (line 7). We update $\mathbf{r}$ into $\mathbf{r}'$ for the next iteration (line 8). The iteration stops when the error δ is smaller than a threshold ϵ (line 9).

Theoretical Analysis of Iterative Algorithm and Signed Random Walk with Restart

We theoretically analyze the iterative algorithm SRWR-Iter and the properties of Signed Random Walk with Restart.

Convergence Analysis of SRWR-Iter. We show that the iteration in Algorithm 2 converges to the solution of a linear system as described in the following theorem.

Theorem 1 (Convergence of SRWR-Iter)**.** *Suppose* $\mathbf{r} = [\mathbf{r}^+; \mathbf{r}^-]^\top$ *and* $\mathbf{q}_s = [\mathbf{q}; \mathbf{0}]^\top$*. Then the iteration for* $\mathbf{r}$ *in Algorithm 2 converges to the solution* $\mathbf{r} = c(\mathbf{I} - (1-c)\tilde{\mathbf{B}}^\top)^{-1}\mathbf{q}_s$ *where* $\tilde{\mathbf{B}}^\top = \begin{bmatrix} \tilde{\mathbf{A}}_+^\top & \beta\tilde{\mathbf{A}}_-^\top + (1-\gamma)\tilde{\mathbf{A}}_+^\top \\ \tilde{\mathbf{A}}_-^\top & (1-\beta)\tilde{\mathbf{A}}_-^\top + \gamma\tilde{\mathbf{A}}_+^\top \end{bmatrix}$.

Proof. Equation (3) is represented as follows:

$$\begin{bmatrix} \mathbf{r}^+ \\ \mathbf{r}^- \end{bmatrix} = (1-c) \begin{bmatrix} \tilde{\mathbf{A}}_+^\top & \beta\tilde{\mathbf{A}}_-^\top + (1-\gamma)\tilde{\mathbf{A}}_+^\top \\ \tilde{\mathbf{A}}_-^\top & (1-\beta)\tilde{\mathbf{A}}_-^\top + \gamma\tilde{\mathbf{A}}_+^\top \end{bmatrix} \begin{bmatrix} \mathbf{r}^+ \\ \mathbf{r}^- \end{bmatrix} + c \begin{bmatrix} \mathbf{q} \\ \mathbf{0} \end{bmatrix} \Leftrightarrow \mathbf{r} = (1-c)\tilde{\mathbf{B}}^\top\mathbf{r} + c\mathbf{q}_s$$

where $\tilde{\mathbf{B}}^\top = \begin{bmatrix} \tilde{\mathbf{A}}_+^\top & \beta\tilde{\mathbf{A}}_-^\top + (1-\gamma)\tilde{\mathbf{A}}_+^\top \\ \tilde{\mathbf{A}}_-^\top & (1-\beta)\tilde{\mathbf{A}}_-^\top + \gamma\tilde{\mathbf{A}}_+^\top \end{bmatrix}$, $\mathbf{r} = \begin{bmatrix} \mathbf{r}^+ \\ \mathbf{r}^- \end{bmatrix}$, and $\mathbf{q}_s = \begin{bmatrix} \mathbf{q} \\ \mathbf{0} \end{bmatrix}$. Thus, the iteration in Algorithm 2 is written as in the following equation:

$$\begin{aligned} \mathbf{r}^{(k)} &= (1-c)\tilde{\mathbf{B}}^\top\mathbf{r}^{(k-1)} + c\mathbf{q}_s \\ &= \left((1-c)\tilde{\mathbf{B}}^\top\right)^2 \mathbf{r}^{(k-2)} + \left((1-c)\tilde{\mathbf{B}}^\top + \mathbf{I}\right) c\mathbf{q}_s \\ &= \cdots \\ &= \left((1-c)\tilde{\mathbf{B}}^\top\right)^k \mathbf{r}^{(0)} + \left(\sum_{j=0}^{k-1} \left((1-c)\tilde{\mathbf{B}}^\top\right)^j\right) c\mathbf{q}_s \end{aligned}$$

The spectral radius $\rho((1-c)\tilde{\mathbf{B}}^\top) = (1-c) < 1$ when $0 < c < 1$ since $\tilde{\mathbf{B}}^\top$ is a column stochastic matrix and its largest eigenvalue is 1 (Strang, 2006). Therefore, $\lim_{k\to\infty}((1-c)\tilde{\mathbf{B}}^\top)^k\mathbf{r}^{(0)} = \mathbf{0}$ and $\lim_{k\to\infty}\mathbf{r}^{(k)}$ converges as follows:

$$\lim_{k\to\infty} \mathbf{r}^{(k)} = \mathbf{0} + \lim_{k\to\infty} \left(\sum_{j=0}^{k-1} \left((1-c)\tilde{\mathbf{B}}^\top\right)^j\right) c\mathbf{q}_s = c\left(\mathbf{I} - (1-c)\tilde{\mathbf{B}}^\top\right)^{-1}\mathbf{q}_s.$$

In the above equation, $\sum_{j=0}^{\infty}((1-c)\tilde{\mathbf{B}}^\top)^j$ is a geometric series of the matrix $(1-c)\tilde{\mathbf{B}}^\top$, and the series converges to $(\mathbf{I}-(1-c)\tilde{\mathbf{B}}^\top)^{-1}$ since the spectral radius of $(1-c)\tilde{\mathbf{B}}^\top$ is less than one. Note that $(\mathbf{I}-(1-c)\tilde{\mathbf{B}}^\top)^{-1}$ is a non-negative matrix whose entries are positive or zero because the matrix is the sum of non-negative matrices (i.e., $\sum_{j=0}^{\infty}((1-c)\tilde{\mathbf{B}}^\top)^j$). Hence, each entry of $\mathbf{r}$ is non-negative (i.e., $\mathbf{r}_u \geq 0$ for any node u). □

Complexity Analysis of SRWR-Iter. We analyze the complexity of SRWR-Iter in terms of space and time.

Theorem 2 (Space and Time Complexities of SRWR-Iter)**.** *Let n and m denote the number of nodes and edges of a signed network, respectively. Then the space complexity of Algorithm 2 is $O(m+n)$. The time complexity of Algorithm 2 is $O(T(m+n))$ where T is the number of iterations.*

Proof. In Algorithm 2, the space complexity for $\tilde{\mathbf{A}}_+$ and $\tilde{\mathbf{A}}_-$ is $O(m)$ if we exploit a sparse matrix format such as compressed column storage to save the matrices. We also need $O(n)$ for SRWR score vectors $\mathbf{r}^+$ and $\mathbf{r}^-$. Thus, the space complexity is $O(m+n)$. One iteration in Algorithm 2 takes $O(m+n)$ time due to sparse matrix vector multiplications and vector additions where the time complexity of a sparse matrix vector multiplication is linearly proportional to the number of non-zeros of a matrix (Duff, Grimes and Lewis, 1989). Hence, the total time complexity is $O(T(m+n))$ where T is the number of iterations. □

Theorem 2 indicates that the space and time complexities of SRWR-Iter are linear with respect to the number of edges and nodes.

Properties of SRWR. We discuss the properties of our ranking model SRWR to answer the following questions: 1) Is the signed random surfer able to visit all nodes in a network which is strongly connected? and 2) Does SRWR work on unsigned networks as well? The first question is answered in Lemma 1, and the second one is answered in Lemma 2.

Lemma 1. *Suppose a signed network is strongly connected. Then, all entries of $\mathbf{r}^+ + \mathbf{r}^-$ are positive (i.e., $\mathbf{r}^+ + \mathbf{r}^- > 0$).*

Proof. Let $\mathbf{r}^+ + \mathbf{r}^-$ be $\mathbf{p}$. By summing the recursive equations on $\mathbf{r}^+$ and $\mathbf{r}^-$ in Equation (3), $\mathbf{p}$ is represented as follows:

$$\mathbf{p} = (1-c)\left(\tilde{\mathbf{A}}_+^\top \mathbf{p} + \tilde{\mathbf{A}}_-^\top \mathbf{p}\right) + c\mathbf{q} \Leftrightarrow \mathbf{p} = (1-c)|\tilde{\mathbf{A}}|^\top \mathbf{p} + c\mathbf{q} \Leftrightarrow \mathbf{p} = \mathbf{G}\mathbf{p}$$

where $|\tilde{\mathbf{A}}| = \tilde{\mathbf{A}}_+ + \tilde{\mathbf{A}}_-$ by Definition 3, $\mathbf{G} = (1-c)|\tilde{\mathbf{A}}|^\top + c\mathbf{q}\mathbf{1}^\top$, and $\mathbf{1}^\top\mathbf{p} = \sum_i \mathbf{p}_i = 1$ by Lemma 5. Note that the graph represented by $\mathbf{G}$ is also strongly connected since the graph of $|\tilde{\mathbf{A}}|$ has the same topology with the original graph which is strongly connected. Moreover, the graph represented by $\mathbf{G}$ has a self-loop at the seed node s due to $c\mathbf{q}\mathbf{1}^\top$. Thus, $\mathbf{G}$ is irreducible and aperiodic. Hence, all entries of $\mathbf{p} = \mathbf{r}^+ + \mathbf{r}^-$ are positive according to Perron-Frobenius theorem (Langville, Meyer and Fernández, 2008). □

Note that $\mathbf{r}_u^+$ (or $\mathbf{r}_u^-$) indicates that the stationary probability of the positive (or negative) surfer visits node u after performing SRWR starting from a seed node. According to Lemma 1, $\mathbf{r}_u^+ + \mathbf{r}_u^-$ for an arbitrary node u is always positive if a given signed network is strongly connected. That is, the signed random surfer is able to visit node u with probability $\mathbf{r}_u^+ + \mathbf{r}_u^-$ which is always greater than zero.

Next, we prove that our model SRWR is a generalized version of RWR working on both unsigned and signed networks in the following lemma.

Lemma 2. *The result of SRWR on networks containing only positive edges is the same as that of RWR.*

Proof. $\tilde{\mathbf{A}}_+ = \tilde{\mathbf{A}}$ and $\tilde{\mathbf{A}}_- = \mathbf{0}_{n\times n}$ because the adjacency matrix $\mathbf{A}$ only contains positive edges. Also, $\mathbf{r}^- = \mathbf{0}_{n\times 1}$ at the beginning time of Algorithm 2. Equation (3) is represented as follows:

$$\mathbf{r}^+ = (1-c)\left(\tilde{\mathbf{A}}^\top \mathbf{r}^+ + \beta \mathbf{0}_{n\times n} \times \mathbf{0}_{n\times 1} + (1-\gamma)\tilde{\mathbf{A}}^\top \mathbf{0}_{n\times 1}\right) + c\mathbf{q}$$

$$\mathbf{r}^- = (1-c)\left(\mathbf{0}_{n\times n} \times \mathbf{r}^+ + \gamma \tilde{\mathbf{A}}^\top \mathbf{0}_{n\times 1} + (1-\beta)\mathbf{0}_{n\times n} \times \mathbf{0}_{n\times 1}\right)$$

Therefore, $\mathbf{r}^- = \mathbf{0}_{n\times 1}$ and $\mathbf{r}^+ = (1-c)\tilde{\mathbf{A}}^\top \mathbf{r}^+ + c\mathbf{q}$. The equation of $\mathbf{r}^+$ is exactly the same as that of RWR. □

SRWR-Pre: Preprocessing Algorithm for Signed Random Walk with Restart

We propose SRWR-Pre, a preprocessing algorithm to quickly compute SRWR scores. The iterative approach SRWR-Iter in Algorithm 2 requires multiple matrix-vector multiplications to compute SRWR scores whenever seed node s changes; thus the iterative method is not fast enough when we require SRWR scores for any pair of nodes in large-scale signed networks. Our goal is to directly compute SRWR scores from precomputed intermediate data without iterations. We exploit the following ideas for our preprocessing method:

- The positive and negative SRWR score vectors $\mathbf{r}^+$ and $\mathbf{r}^-$ are obtained by solving linear systems (Section 3.5.1).
- The adjacency matrix of real-world graphs is permuted so that it contains a large but easy-to-invert block diagonal matrix as shown in Figure 5 (Section 3.5.2).
- The block elimination approach efficiently solves a linear system on a matrix having an easy-to-invert sub-matrix (Section 3.5.3).

Our preprocessing method comprises two phases: preprocessing phase (Algorithm 3) and query phase (Algorithm 4). The preprocessing phase preprocesses a given signed adjacency matrix into several sub-matrices required in the query phase to compute SRWR scores w.r.t. seed node s. Note that the preprocessing phase is performed once, and the query phase is run for each seed node. The starting vector $\mathbf{q}$ in Equation (3) is called an SRWR query, and $\mathbf{r}^+$ and $\mathbf{r}^-$ are the results corresponding to the query $\mathbf{q}$. The query vector $\mathbf{q}$ is determined by the seed node s, and $\mathbf{r}^+$ and $\mathbf{r}^-$ are distinct for each SRWR query. To exploit sparsity of graphs, we save all matrices in a sparse matrix format such as compressed column storage (Duff et al., 1989) which stores only non-zero entries and their locations.

Formulation of Signed Random Walk with Restart as Linear Systems

We first represent linear systems related to $\mathbf{r}^+$ and $\mathbf{r}^-$. Let $\mathbf{p}$ be the sum of $\mathbf{r}^+$ and $\mathbf{r}^-$ (i.e., $\mathbf{p} = \mathbf{r}^+ + \mathbf{r}^-$). Then, $\mathbf{p}$ is the solution of the following linear system:

$$|\mathbf{H}|\mathbf{p} = c\mathbf{q} \Leftrightarrow \mathbf{p} = c|\mathbf{H}|^{-1}\mathbf{q} \tag{4}$$

where $|\mathbf{H}| = \mathbf{I} - (1-c)|\tilde{\mathbf{A}}|^\top$ and $|\tilde{\mathbf{A}}| = \tilde{\mathbf{A}}_+ + \tilde{\mathbf{A}}_-$. The proof of Equation (4) is presented in Lemma 3. The linear system for $\mathbf{r}^-$ is given by the following equation:

$$\mathbf{T}\mathbf{r}^- = (1-c)\tilde{\mathbf{A}}_-^\top \mathbf{p} \Leftrightarrow \mathbf{r}^- = (1-c)\left(\mathbf{T}^{-1}(\tilde{\mathbf{A}}_-^\top \mathbf{p})\right) \tag{5}$$

where $\mathbf{T} = \mathbf{I} - (1-c)(\gamma\tilde{\mathbf{A}}_+^\top - \beta\tilde{\mathbf{A}}_-^\top)$, and γ and β are balance attenuation factors. Theorem 3 shows the proof of Equation (5). Based on the aforementioned linear systems in Equations (4) and (5), $\mathbf{r}^-$ and $\mathbf{r}^+$ for a given seed node s are computed as follows:

1. Set a query vector $\mathbf{q}$ whose s-th element is 1 and all other elements are 0.
2. Solve the linear system in Equation (4) to obtain the solution $\mathbf{p}$.
3. Compute $\mathbf{r}^-$ by solving the linear system in Equation (5).
4. Compute $\mathbf{r}^+ = \mathbf{p} - \mathbf{r}^-$.

Lemma 3. *Suppose that* $\mathbf{p} = \mathbf{r}^+ + \mathbf{r}^-$, $|\mathbf{H}| = \mathbf{I} - (1-c)|\tilde{\mathbf{A}}|^\top$ *and* $|\tilde{\mathbf{A}}| = \tilde{\mathbf{A}}_+ + \tilde{\mathbf{A}}_-$. *Then,* $\mathbf{p}$ *is the solution of the following linear system:*

$$|\mathbf{H}|\mathbf{p} = c\mathbf{q} \Leftrightarrow \mathbf{p} = c|\mathbf{H}|^{-1}\mathbf{q}$$

Proof. According to the result in Lemma 1, the recursive equation for $\mathbf{p}$ is represented as follows:

$$\mathbf{p} = (1-c)|\tilde{\mathbf{A}}|^\top \mathbf{p} + c\mathbf{q}$$

where $|\tilde{\mathbf{A}}| = \tilde{\mathbf{A}}_+ + \tilde{\mathbf{A}}_-$ is the row-normalized matrix of $|\mathbf{A}|$. The linear system for $\mathbf{p}$ is represented by moving $(1-c)|\tilde{\mathbf{A}}|^\top \mathbf{p}$ to the left side as follows:

$$\left(\mathbf{I} - (1-c)|\tilde{\mathbf{A}}|^\top\right)\mathbf{p} = c\mathbf{q} \Leftrightarrow |\mathbf{H}|\mathbf{p} = c\mathbf{q}$$

where $|\mathbf{H}|$ is $\mathbf{I} - (1-c)|\tilde{\mathbf{A}}|^\top$. Note that $|\mathbf{H}|$ is invertible when $0 < c < 1$ because it is strictly diagonally dominant (Van Loan, 1996). Hence, $\mathbf{p} = c|\mathbf{H}|^{-1}\mathbf{q}$. □

Theorem 3. *The SRWR score vectors* $\mathbf{r}^+$ *and* $\mathbf{r}^-$ *from Equation* (3) *are represented as follows:*

$$\mathbf{r}^+ = \mathbf{p} - \mathbf{r}^-$$
$$\mathbf{r}^- = (1-c)\left(\mathbf{T}^{-1}(\tilde{\mathbf{A}}_-^\top \mathbf{p})\right)$$

where $\mathbf{p} = c|\mathbf{H}|^{-1}\mathbf{q}$, $\mathbf{T} = \mathbf{I} - (1-c)(\gamma\tilde{\mathbf{A}}_+^\top - \beta\tilde{\mathbf{A}}_-^\top)$, *and* γ *and* β *are balance attenuation factors which are between* 0 *and* 1 *(i.e.,* $0 < \gamma, \beta < 1$*).*

Proof. Note that $\mathbf{r}^- = (1-c)(\tilde{\mathbf{A}}_-^\top \mathbf{r}^+ + \gamma\tilde{\mathbf{A}}_+^\top \mathbf{r}^- + (1-\beta)\tilde{\mathbf{A}}_-^\top \mathbf{r}^-)$ by Equation (3), and $\mathbf{r}^+ = \mathbf{p} - \mathbf{r}^-$ according to Lemma 3. The equation for $\mathbf{r}^-$ is represented by plugging $\mathbf{r}^+ = \mathbf{p} - \mathbf{r}^-$ as follows:

$$\mathbf{r}^- = (1-c)\left(\tilde{\mathbf{A}}_-^\top \mathbf{p} - \tilde{\mathbf{A}}_-^\top \mathbf{r}^- + \gamma\tilde{\mathbf{A}}_+^\top \mathbf{r}^- + (1-\beta)\tilde{\mathbf{A}}_-^\top \mathbf{r}^-\right) \Leftrightarrow$$
$$\mathbf{r}^- = (1-c)\left(\gamma\tilde{\mathbf{A}}_+^\top - \beta\tilde{\mathbf{A}}_-^\top\right)\mathbf{r}^- + (1-c)\tilde{\mathbf{A}}_-^\top \mathbf{p}$$

We move $(1-c)(\gamma\tilde{\mathbf{A}}_+^\top - \beta\tilde{\mathbf{A}}_-^\top)\mathbf{r}^-$ to the left side; then, the above equation is represented as follows:

$$\left(\mathbf{I} - (1-c)(\gamma\tilde{\mathbf{A}}_+^\top - \beta\tilde{\mathbf{A}}_-^\top)\right)\mathbf{r}^- = (1-c)\tilde{\mathbf{A}}_-^\top\mathbf{p} \Leftrightarrow \mathbf{T}\mathbf{r}^- = (1-c)\tilde{\mathbf{A}}_-^\top\mathbf{p}$$

where $\mathbf{T}$ is $\mathbf{I}-(1-c)(\gamma\tilde{\mathbf{A}}_+^\top-\beta\tilde{\mathbf{A}}_-^\top)$. Note that the matrix $\mathbf{T}$ is strictly diagonally dominant when $0 < c < 1$ and $0 < \gamma, \beta < 1$; thus, $\mathbf{T}$ is invertible. Hence, $\mathbf{r}^- = (1-c)(\mathbf{T}^{-1}(\tilde{\mathbf{A}}_-^\top\mathbf{p}))$. $\mathbf{r}^+$ is obtained by computing $\mathbf{r}^+ = \mathbf{p} - \mathbf{r}^-$. □

One naive approach (Inversion) for SRWR score vectors $\mathbf{r}^+$ and $\mathbf{r}^-$ based on the linear systems in Equations (4) and (5) is to precompute the inverse of the matrices $|\mathbf{H}|$ and $\mathbf{T}$. However, this approach is impractical for large-scale graphs since inverting a matrix requires $O(n^3)$ time and $O(n^2)$ space where n is the dimensions of the matrix. Another approach (LU) is to obtain the inverse of LU factors of $|\mathbf{H}|$ and $\mathbf{T}$ after reordering the matrices in the order of node degrees as suggested in (Fujiwara, Nakatsuji, Onizuka and Kitsuregawa, 2012) (i.e., $\mathbf{p} = c(\mathbf{U}_\mathbf{p}^{-1}(\mathbf{L}_\mathbf{p}^{-1}\mathbf{q}))$; $\mathbf{r}^- = (1-c)(\mathbf{U}_{\mathbf{r}^-}^{-1}(\mathbf{L}_{\mathbf{r}^-}^{-1}(\tilde{\mathbf{A}}_-^\top\mathbf{p})))$ where $|\mathbf{H}|^{-1} = \mathbf{U}_\mathbf{p}^{-1}\mathbf{L}_\mathbf{p}^{-1}$ and $\mathbf{T}^{-1} = \mathbf{U}_{\mathbf{r}^-}^{-1}\mathbf{L}_{\mathbf{r}^-}^{-1}$). Although LU is more efficient than Inversion in terms of time and space as shown in Figure 10, LU still has a performance issue due to $O(n^3)$ time and $O(n^2)$ space complexities. On the other hand, our preprocessing method SRWR-Pre is faster and more memory efficient than Inversion and LU as we will see in Section 4.5.

Node Reordering based on Hub-and-Spoke Structure

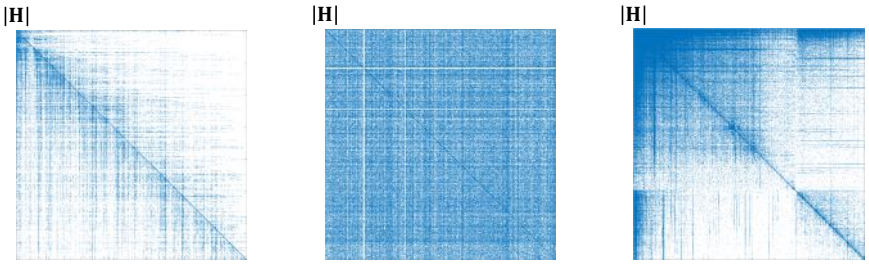
SRWR-Pre permutes the matrices $|\mathbf{H}|$ and $\mathbf{T}$ using a node reordering technique based on the hub-and-spoke structure. Previous works (Shin et al., 2015) have exploited the node reordering technique to reduce computational cost of graph operations in real-world graphs. We also adopt the node reordering based on the hub-and-spoke structure to efficiently solve the linear systems in Equations (4) and (5).

The hub-and-spoke structure indicates that most real-world graphs follow power-law degree distribution with few hubs (very high degree nodes) and majority of spokes (low degree nodes). The structure has been utilized to concentrate entries of an adjacency matrix by reordering nodes as shown in Figure 5. Any reordering method based on the hub-and-spoke structure can be utilized for the purpose; in this paper, we use SlashBurn (Kang and Faloutsos, 2011; Lim, Kang and Faloutsos, 2014) as a hub-and-spoke reordering method because it shows the best performance in concentrating entries of an adjacency matrix (see the details in Appendix 7.1).

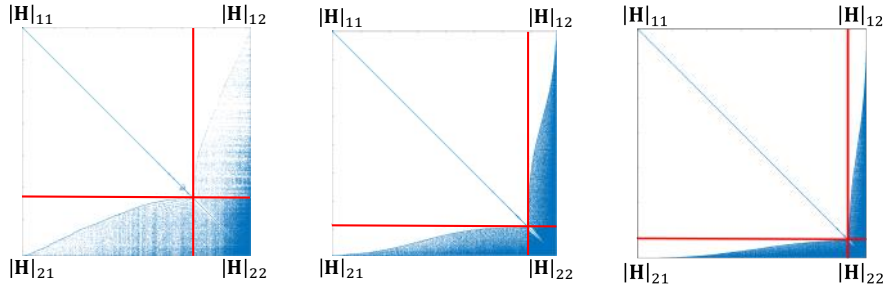
We reorder nodes of the signed adjacency matrix $\mathbf{A}$ so that reordered matrix contains a large but easy-to-invert submatrix such as block diagonal matrix as shown in Figure 5. We then compute $|\mathbf{H}| = \mathbf{I} - (1-c)(\tilde{\mathbf{A}}_+^\top + \tilde{\mathbf{A}}_-^\top)$ and $\mathbf{T} = \mathbf{I} - (1-c)(\gamma\tilde{\mathbf{A}}_+^\top - \beta\tilde{\mathbf{A}}_-^\top)$. Note that $|\mathbf{H}|$ and $\mathbf{T}$ have the same sparsity pattern as the reordered adjacency matrix $\mathbf{A}^\top$ except for the diagonal part. Hence, $|\mathbf{H}|$ and $\mathbf{T}$ are partitioned as follows:

$$|\mathbf{H}| = \begin{bmatrix} |\mathbf{H}|_{11} & |\mathbf{H}|_{12} \\ |\mathbf{H}|_{21} & |\mathbf{H}|_{22} \end{bmatrix}, \mathbf{T} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix}. \quad (6)$$

Let n_1 and n_2 denote the number of spokes and hubs, respectively (see the details in Appendix 7.1). Then $|\mathbf{H}|_{11}$ and $\mathbf{T}_{11}$ are $n_1 \times n_1$ matrices, $|\mathbf{H}|_{12}$ and



(a) Original matrix $|\mathbf{H}|$ in the Wikipedia dataset (b) Original matrix $|\mathbf{H}|$ in the Slashdot dataset (c) Original matrix $|\mathbf{H}|$ in the Epinions dataset



(d) Reordered matrix $|\mathbf{H}|$ in the Wikipedia dataset (e) Reordered matrix $|\mathbf{H}|$ in the Slashdot dataset (f) Reordered matrix $|\mathbf{H}|$ in the Epinions dataset

Fig. 5. The result of node reordering on each signed network. (a), (b), and (c) are the original matrix $|\mathbf{H}|$ before node reordering in the Wikipedia, the Slashdot, and the Epinions datasets, respectively. (d), (e) and (f) present $|\mathbf{H}|$ reordered by the hub-and-spoke method. Note that $\mathbf{T}$ is also reordered equivalently to $|\mathbf{H}|$ since they have the same sparsity pattern. $|\mathbf{H}|_{11}$ and $\mathbf{T}_{11}$ are block diagonal.

$\mathbf{T}_{12}$ are $n_1 \times n_2$ matrices, $|\mathbf{H}|_{21}$ and $\mathbf{T}_{21}$ are $n_2 \times n_1$ matrices, and $|\mathbf{H}|_{22}$ and $\mathbf{T}_{22}$ are $n_2 \times n_2$ matrices. The linear systems for $|\mathbf{H}|$ and $\mathbf{T}$ in Equations (4) and (5) are represented as follows:

$$|\mathbf{H}|\mathbf{p} = c\mathbf{q} \Leftrightarrow \begin{bmatrix} |\mathbf{H}|_{11} & |\mathbf{H}|_{12} \\ |\mathbf{H}|_{21} & |\mathbf{H}|_{22} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix} = c \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix} \tag{7}$$

$$\mathbf{T}\mathbf{r}^- = (1-c)\mathbf{t} \Leftrightarrow \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^- \\ \mathbf{r}_2^- \end{bmatrix} = (1-c) \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{bmatrix} \tag{8}$$

where $\mathbf{t} = \tilde{\mathbf{A}}_-^\top \mathbf{p}$ is an $n \times 1$ vector.

Block Elimination for Solving Linear Systems

The solutions of the partitioned linear systems in Equations (7) and (8) are obtained by the following equations:

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix} = \begin{bmatrix} |\mathbf{H}|_{11}^{-1}(c\mathbf{q}_1 - |\mathbf{H}|_{12}\mathbf{p}_2) \\ c(\mathbf{S}_{|\mathbf{H}|}^{-1}(\mathbf{q}_2 - |\mathbf{H}|_{21}(|\mathbf{H}|_{11}^{-1}(\mathbf{q}_1)))) \end{bmatrix} \tag{9}$$

$$\mathbf{r}^- = \begin{bmatrix} \mathbf{r}_1^- \\ \mathbf{r}_2^- \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{11}^{-1}((1-c)\mathbf{t}_1 - \mathbf{T}_{12}\mathbf{r}_2^-) \\ (1-c)(\mathbf{S}_{\mathbf{T}}^{-1}(\mathbf{t}_2 - \mathbf{T}_{21}(\mathbf{T}_{11}^{-1}(\mathbf{t}_1)))) \end{bmatrix} \tag{10}$$

where $\mathbf{S}_{|\mathbf{H}|} = |\mathbf{H}|_{22} - |\mathbf{H}|_{21}|\mathbf{H}|_{11}^{-1}|\mathbf{H}|_{12}$ is the Schur complement of $|\mathbf{H}|_{11}$ and $\mathbf{S}_{\mathbf{T}} = \mathbf{T}_{22} - \mathbf{T}_{21}\mathbf{T}_{11}^{-1}\mathbf{T}_{12}$ is the Schur complement of $\mathbf{T}_{11}$. Equations (9) and (10) are derived by applying block elimination described in Lemma 4 to the partitioned linear systems in Equations (7) and (8), respectively. Note that the sub-matrices $|\mathbf{H}|_{11}$ and $\mathbf{T}_{11}$ are invertible when $0 < c < 1$ and $0 < \gamma, \beta < 1$ since they are strictly diagonally dominant. If all matrices in Equations (9) and (10) are precomputed, then the SRWR score vectors $\mathbf{r}^+$ and $\mathbf{r}^-$ are efficiently and directly computed from the precomputed matrices.

Lemma 4 (Block Elimination (Boyd and Vandenberghe, 2004))**.** *Suppose a linear system* $\mathbf{Ax} = \mathbf{b}$ *is partitioned as follows:*

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

where $\mathbf{A}_{11}$ *and* $\mathbf{A}_{22}$ *are square matrices. If the sub-matrix* $\mathbf{A}_{11}$ *is invertible, then the solution* $\mathbf{x}$ *is represented as follows:*

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}^{-1}(\mathbf{b}_1 - \mathbf{A}_{12}\mathbf{x}_2) \\ \mathbf{S}^{-1}(\mathbf{b}_2 - \mathbf{A}_{21}(\mathbf{A}_{11}^{-1}(\mathbf{b}_1))) \end{bmatrix}$$

where $\mathbf{S} = \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$ *is the Schur complement of* $\mathbf{A}_{11}$*.* ■

Lemma 4 implies that a partitioned linear system is efficiently solved if it contains an easy-to-invert sub-matrix and the dimension of the Schur complement is small. Note that inverting $\mathbf{H}_{11}$ and $\mathbf{T}_{11}$ is trivial because they are block diagonal matrices as shown in Figure 5. Also, the dimension of $\mathbf{S}_{|\mathbf{H}|}$ and $\mathbf{S}_{\mathbf{T}}$ is n_2 where n_2 is the number of hubs and most real-world graphs have a small number of hubs compared to the number of nodes (see Table 2).

Preprocessing phase (Algorithm 3). Our preprocessing phase precomputes the matrices exploited for computing SRWR scores in the query phase. Our algorithm first reorders nodes of a given signed adjacency matrix $\mathbf{A}$ using the hub-and-spoke reordering method, and performs semi-normalization on $\mathbf{A}$ to obtain $\tilde{\mathbf{A}}_+$ and $\tilde{\mathbf{A}}_-$ using Algorithm 1 (lines 1∼2). Then our algorithm computes $|\mathbf{H}|$ and $\mathbf{T}$, and partitions the matrices (lines 3∼5). Our algorithm calculates the inverses of $|\mathbf{H}|_{11}$ and $\mathbf{T}_{11}$, and computes the Schur complements of $|\mathbf{H}|_{11}$ and $\mathbf{T}_{11}$ (lines 4∼7). When we compute $\mathbf{S}_{|\mathbf{H}|}^{-1}$ and $\mathbf{S}_{\mathbf{T}}^{-1}$, we invert the LU factors of $\mathbf{S}_{|\mathbf{H}|}$ and $\mathbf{S}_{\mathbf{T}}$ (lines 8 and 9) because this approach is faster and more memory efficient than directly inverting $\mathbf{S}_{|\mathbf{H}|}$ and $\mathbf{S}_{\mathbf{T}}$ as in (Shin et al., 2015).

Query phase (Algorithm 4). Our query phase computes SRWR score vectors $\mathbf{r}^+$ and $\mathbf{r}^-$ for a given seed node s using precomputed matrices from Algorithm 3. Our algorithm first creates a starting vector $\mathbf{q}$ whose entry at the index of the seed node s is 1 and otherwise 0, and partitions $\mathbf{q}$ into $\mathbf{q}_1$ and $\mathbf{q}_2$ (line 1). We then compute $\mathbf{p}_2$ and $\mathbf{p}_1$ based on Equation (9), and concatenate the vectors to obtain $\mathbf{p}$ (lines 2∼4). Our algorithm calculates $\mathbf{t} = \tilde{\mathbf{A}}_-^\top \mathbf{p}$ and partitions $\mathbf{t}$ into $\mathbf{t}_1$ and $\mathbf{t}_2$ (line 5). We compute $\mathbf{r}_2^-$ and $\mathbf{r}_1^-$ based on Equation (10), and concatenate the vectors to obtain $\mathbf{r}^-$ (lines 6∼8). Finally, we compute $\mathbf{r}^+ = \mathbf{p} - \mathbf{r}^-$ to obtain $\mathbf{r}^+$ (line 9).

Algorithm 3: Preprocessing phase of SRWR-Pre

Input: signed adjacency matrix: $\mathbf{A}$, restart probability: c, balance attenuation factors: β and γ
Output: preprocessed matrices from $|\mathbf{H}|$ and $\mathbf{T}$, negative semi-row normalized matrix $\tilde{\mathbf{A}}_-$

reorder $\mathbf{A}$ using the hub-and-spoke reordering method (Kang and Faloutsos, 2011; Lim et al., 2014)
compute $\tilde{\mathbf{A}}_+$ and $\tilde{\mathbf{A}}_-$ from $\mathbf{A}$ using Algorithm 1
compute $|\mathbf{H}|$ and $\mathbf{T}$, i.e., $|\mathbf{H}| = \mathbf{I} - (1-c)|\tilde{\mathbf{A}}|^\top$ and $\mathbf{T} = \mathbf{I} - (1-c)(\gamma\tilde{\mathbf{A}}_+^\top - \beta\tilde{\mathbf{A}}_-^\top)$
partition $|\mathbf{H}|$ into $|\mathbf{H}|_{11}$, $|\mathbf{H}|_{12}$, $|\mathbf{H}|_{21}$, $|\mathbf{H}|_{22}$, and compute $|\mathbf{H}|_{11}^{-1}$
partition $\mathbf{T}$ into $\mathbf{T}_{11}$, $\mathbf{T}_{12}$, $\mathbf{T}_{21}$, $\mathbf{T}_{22}$, and compute $\mathbf{T}_{11}^{-1}$
compute the Schur complement of $|\mathbf{H}|_{11}$, i.e., $\mathbf{S}_{|\mathbf{H}|} = |\mathbf{H}|_{22} - |\mathbf{H}|_{21}|\mathbf{H}|_{11}^{-1}|\mathbf{H}|_{12}$
compute the Schur complement of $\mathbf{T}_{11}$, i.e., $\mathbf{S}_{\mathbf{T}} = \mathbf{T}_{22} - \mathbf{T}_{21}\mathbf{T}_{11}^{-1}\mathbf{T}_{12}$
compute the inverse of LU factors of $\mathbf{S}_{|\mathbf{H}|}$, i.e., $\mathbf{S}_{|\mathbf{H}|}^{-1} = \mathbf{U}_{|\mathbf{H}|}^{-1}\mathbf{L}_{|\mathbf{H}|}^{-1}$
compute the inverse of LU factors of $\mathbf{S}_{\mathbf{T}}$, i.e., $\mathbf{S}_{\mathbf{T}}^{-1} = \mathbf{U}_{\mathbf{T}}^{-1}\mathbf{L}_{\mathbf{T}}^{-1}$
return preprocessed matrices from $|\mathbf{H}|$: $\mathbf{L}_{|\mathbf{H}|}^{-1}$, $\mathbf{U}_{|\mathbf{H}|}^{-1}$, $|\mathbf{H}|_{11}^{-1}$, $|\mathbf{H}|_{12}$, and $|\mathbf{H}|_{21}$
preprocessed matrices from $\mathbf{T}$: $\mathbf{L}_{\mathbf{T}}^{-1}$, $\mathbf{U}_{\mathbf{T}}^{-1}$, $\mathbf{T}_{11}^{-1}$, $\mathbf{T}_{12}$, and $\mathbf{T}_{21}$
negative semi-row normalized matrix $\tilde{\mathbf{A}}_-$

Algorithm 4: Query phase of SRWR-Pre

Input: seed node: s, preprocessed matrices from Algorithm 3
Output: trust SRWR scores: $\mathbf{r}^+$, distrust SRWR scores: $\mathbf{r}^-$

create $\mathbf{q}$ whose s-th entry is 1 and the others are 0, and partition $\mathbf{q}$ into $\mathbf{q}_1$ and $\mathbf{q}_2$
compute $\mathbf{p}_2 = c(\mathbf{U}_{|\mathbf{H}|}^{-1}(\mathbf{L}_{|\mathbf{H}|}^{-1}(\mathbf{q}_2 - |\mathbf{H}|_{21}(|\mathbf{H}|_{11}^{-1}\mathbf{q}_1))))$
compute $\mathbf{p}_1 = |\mathbf{H}|_{11}^{-1}(c\mathbf{q}_1 - |\mathbf{H}|_{12}\mathbf{p}_2)$
create $\mathbf{p}$ by concatenating $\mathbf{p}_1$ and $\mathbf{p}_2$
compute $\mathbf{t} = \tilde{\mathbf{A}}_-^\top\mathbf{p}$, and partition it into $\mathbf{t}_1$ and $\mathbf{t}_2$
compute $\mathbf{r}_2^- = (1-c)(\mathbf{U}_{\mathbf{T}}^{-1}(\mathbf{L}_{\mathbf{T}}^{-1}(\mathbf{t}_2 - \mathbf{T}_{21}(\mathbf{T}_{11}^{-1}\mathbf{t}_1))))$
compute $\mathbf{r}_1^- = \mathbf{T}_{11}^{-1}((1-c)\mathbf{t}_1 - \mathbf{T}_{12}\mathbf{r}_2^-))$
create $\mathbf{r}^-$ by concatenating $\mathbf{r}_1^-$ and $\mathbf{r}_2^-$
compute $\mathbf{r}^+ = \mathbf{p} - \mathbf{r}^-$
return $\mathbf{r}^+$ and $\mathbf{r}^-$

Theoretical Analysis of Preprocessing Algorithm for Signed Random Walk with Restart

We analyze the theoretical complexities of our method SRWR-Pre.

Space Complexity Analysis of SRWR-Pre. We prove the space complexity for SRWR-Pre in Theorem 4.

Theorem 4 (Space Complexity of SRWR-Pre)**.** *The space complexity of* SRWR-Pre *is* $O((\sum_{i=1}^{b} n_{1i}^2) + \min(n_1 n_2, m) + m_- + n_2^2)$ *where* b *is the number of blocks,* n_{1i} *is the number of nodes in* i*-th block of* $|\mathbf{H}|_{11}$ *or* $\mathbf{T}_{11}$*,* n_2 *is the number of hubs, and* m_- *is the number of negative edges.*

Proof. Note that $|\mathbf{H}|$ and $\mathbf{T}$ have the same sparsity pattern; thus, the block structures of $|\mathbf{H}|_{11}$ and $\mathbf{T}_{11}$ are the same. Hence, the memory space required by $|\mathbf{H}|_{11}$ and $\mathbf{T}_{11}$ is $O(\sum_{i=1}^{b} n_{1i}^2)$ where n_{1i}^2 is the maximum number of non-zeros in i-th block. The matrices $|\mathbf{H}|_{12}$, $|\mathbf{H}|_{21}$, $\mathbf{T}_{12}$, and $\mathbf{T}_{21}$ require $O(\min(n_1 n_2, m))$ memory space. The space complexity for the inverse of LU factors $\mathbf{U}_{|\mathbf{H}|}^{-1}$, $\mathbf{L}_{|\mathbf{H}|}^{-1}$, $\mathbf{U}_{\mathbf{T}}^{-1}$ and $\mathbf{L}_{\mathbf{T}}^{-1}$ is $O(n_2^2)$. The memory space for $\tilde{\mathbf{A}}_-$ is $O(m_-)$ where m_- is the

number of negative edges. Therefore, the total space complexity of SRWR-Pre is $O((\sum_{i=1}^{b} n_{1i}^2) + \min(n_1 n_2, m) + m_- + n_2^2)$. □

Time Complexity Analysis of SRWR-Pre. We prove the time complexity of our preprocessing phase in Theorem 5 and the time complexity of our query phase in Theorem 6.

Theorem 5 (Time Complexity of Algorithm 3)**.** *The preprocessing phase in Algorithm 3 takes* $O(T(m+n\log n)+\sum_{i=1}^{b} n_{1i}^3+(n_2\sum_{i=1}^{b} n_{1i}^2)+n_2^3+\min(n_2^2 n_1, n_2 m))$ *where* T *is the number of iterations required by the hub-and-spoke reordering.*

Proof. It takes $O(T(m+n\log n)+\sum_{i=1}^{b} n_{1i}^3+(n_2\sum_{i=1}^{b} n_{1i}^2)+n_2^3+\min(n_2^2 n_1, n_2 m))$ time to obtain the preprocessed matrices from $|\mathbf{H}|$. The time complexity of preprocessing $\mathbf{T}$ is equal to that of preprocessing $|\mathbf{H}|$ since $\mathbf{T}$ has the same number of nodes (n), spokes (n_1) and hubs (n_2) with $|\mathbf{H}|$. □

Theorem 6 (Time Complexity of Algorithm 4)**.** *The query phase in Algorithm 4 takes* $O((\sum_{i=1}^{b} n_{1i}^2) + n_2^2 + \min(n_1 n_2, m) + m_- + n)$ *where* m_- *is the number of negative edges.*

Proof. It takes $O((\sum_{i=1}^{b} n_{1i}^2) + n_2^2 + \min(n_1 n_2, m))$ time to obtain $\mathbf{p}_2$ and $\mathbf{r}_2^-$ (lines 2 and 6), and takes $O((\sum_{i=1}^{b} n_{1i}^2) + \min(n_1 n_2, m))$ to compute $\mathbf{p}_1$ and $\mathbf{r}_1^-$ (lines 3 and 7). Computing $\mathbf{t} = \tilde{\mathbf{A}}_-^\top \mathbf{p}$ requires $O(m_-)$ time where m_- is the number of negative edges since $\tilde{\mathbf{A}}_-^\top \mathbf{p}$ is a sparse matrix vector multiplication (line 5). It takes $O(n)$ time to concatenate the partitioned vectors (lines 4 and 8) and compute $\mathbf{r}^+$ (line 9). Hence, the total time complexity of the query phase is $O((\sum_{i=1}^{b} n_{1i}^2) + n_2^2 + \min(n_1 n_2, m) + m_- + n)$. □

Theorems 4, 5, and 6 indicate that SRWR-Pre performs efficiently in terms of time and space if the given graph is split into many small components (large $n_1 = \sum_{i=1}^{b} n_{1i}$) by removing a small number of hubs (small n_2), which is true in real-world signed networks as shown in Figure 5 and Table 2.

Experiments

We evaluate the effectiveness of SRWR compared to existing ranking methods. Since there is no ground-truth of personalized rankings for each node in real-world graphs, we exploit an indirect way by examining the performance of applications such as a sign prediction task and a troll identification task. We also investigate the performance of our proposed approaches in terms of time and space. Based on these settings, we aim to answer the following questions from the experiments:

- **Q1.** How effective is our proposed SRWR model for the sign prediction task? (Section 4.2)
- **Q2.** How helpful is SRWR for identifying trolls who are abnormal users compared to other ranking models? (Section 4.3)
- **Q3.** How effective are the balance attenuation factors of SRWR for the sign prediction and the troll identification tasks? (Section 4.4)
- **Q4.** How fast and memory efficient is our preprocessing method SRWR-Pre for computing SRWR scores compared to other baselines? (Section 4.5)

Table 2. Dataset statistics. n is the number of nodes and m is the total number of edges. m_+ is the number of positive edges, m_- is the number of negative edges, and n_2 is the number of hubs.

Dataset	n	m	m_+	m_-	n_2
Wikipedia[1]	7,118	103,617	81,285	22,332	1,800
Slashdot[2]	79,120	515,561	392,316	123,245	10,160
Epinions[3]	131,828	841,372	717,667	123,705	10,164

[1] http://snap.stanford.edu/data/wiki-Vote.html
[2] http://dai-labor.de/IRML/datasets
[3] http://www.trustlet.org/wiki/Extended_Epinions_dataset

Experimental Settings

Machines. The experiments on the effectiveness of SRWR in Sections 4.2, 4.3, and 4.4 are conducted on a PC with Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz and 8GB memory. The experiments on the performance of SRWR-Pre in Section 4.5 are performed on a workstation with a single Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz and 256GB memory.

Datasets. The signed networks used in our experiments are summarized in Table 2. We use all datasets in the sign prediction task and the experiments for evaluating the performance of the proposed methods (Sections 4.2 and 4.5). We use the Slashdot dataset in the troll identification task (Section 4.3) since there is a troll list only in the dataset.

Methods. To answer **Q1-3**, we compare our proposed model with Random Walk with Restart (RWR) (Haveliwala, 2002), Modified Random Walk with Restart (M-RWR) (Shahriari and Jalili, 2014), Modified Personalized SALSA (M-PSALSA) (Ng, Zheng and Jordan, 2001), Personalized Signed spectral Rank (PSR) (Kunegis et al., 2009), Personalized Negative Rank (PNR) (Kunegis et al., 2009), and Troll-Trust Model (TR-TR) (Wu, Aggarwal and Sun, 2016). Note that RWR is computed on the absolute adjacency matrices of signed networks. Since most existing methods compute global trust and distrust rankings in the context of PageRank, we set a starting vector of each method as in line 1 of Algorithm 2 to obtain personalized rankings. For **Q4**, we compare our preprocessing method SRWR-Pre to other baseline methods Inversion and LU mentioned in Section 3.5.1 including our iterative method SRWR-Iter.

Parameters. We set the following parameters which give the best performance in each experiment:

- Sign prediction task (Section 4.2): We set c to 0.15 for all random walk based method including our model. In our model, we set $\beta = 0.5, \gamma = 0.9$ in the Epinions dataset, $\beta = 0.6, \gamma = 0.9$ in the Slashdot dataset, and $\beta = 0.1, \gamma = 0.6$ in the Wikipedia dataset. We set $\beta = 0.5, \lambda_1 = 1.0$ in TR-TR.
- Troll identification task (Section 4.3): We set c to 0.15. In our model, we set $\beta = 0.1, \gamma = 1.0$. We set $\beta = 0.5, \lambda_1 = 1.0$ in TR-TR.
- Effect on balance attenuation factor (Section 4.4): We set various balance factors of SRWR for each dataset as mentioned in the sign prediction task setting. The standard SRWR is compared with a version of SRWR (which we call H-SRWR) without balance attenuation factors, which corresponds to setting $\beta = 1$ and $\gamma = 1$ in SRWR.
- Performance of our proposed methods (Section 4.5): We set $\beta = 0.5$, $\gamma = 0.5$, $c = 0.05$ for all tested methods. In SRWR-Pre, we set the hub selection ratio

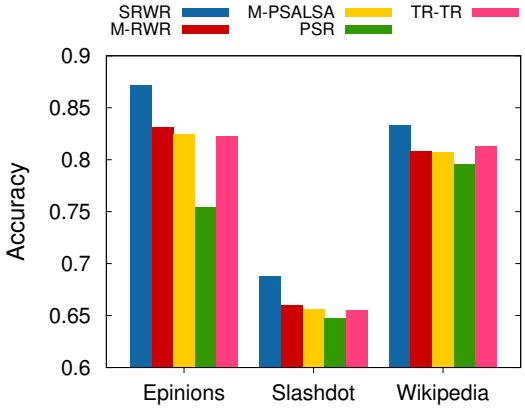
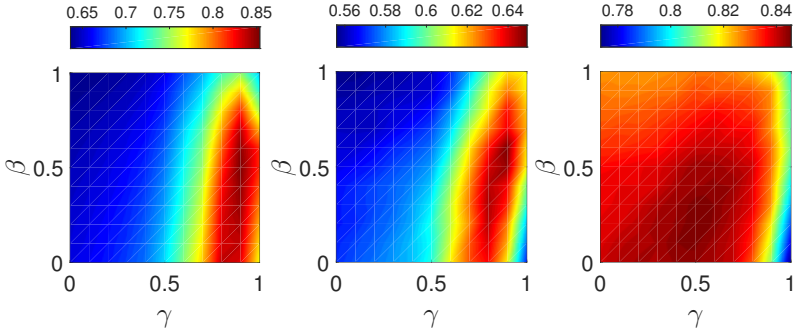


Fig. 6. The accuracy of SRWR in the sign prediction task on different signed networks. As shown in the results, our proposed Signed Random Walk with Restart provides the best performance in terms of accuracy.



(a) Epinions (b) Slashdot (c) Wikipedia

Fig. 7. Accuracy maps according to β and γ where color indicates a degree of accuracy. The Epinions and the Slashdot datasets present similar tendencies while the Wikipedia dataset shows a different result from those of the two datasets.

$t = 0.001$ for the hub-and-spoke reordering method to make the number of hubs n_2 small enough as in (Shin et al., 2015).

Sign Prediction Task

We evaluate ranking models on the sign prediction task defined as follows: given a signed network containing missed signs of edges connected from a node, predict those signs. We randomly select $5,000$ seed nodes for the experiment and choose 20% edges of positive and negative links of each node as a test set. Then, we remove each selected edge $(s \rightarrow t)$, and predict the edge's sign based on personalized ranking scores w.r.t. node s. We compute $\mathbf{r}^d = \mathbf{r}^+ - \mathbf{r}^-$ whose values range from -1 to 1. If $\mathbf{r}^d_t$ is greater than 0, then we predict the sign of the edge $(s \rightarrow t)$ as positive. Otherwise, it is considered as negative. We measure the prediction accuracy by comparing $\mathbf{r}^d_t$ with the true sign of the edge.

Results. We compare the performance of SRWR, M-RWR, M-PSALSA, TR-TR, and PSR on the sign prediction task. As shown in Figure 6, SRWR is the most accurate in predicting signs for all the datasets. This implies that SRWR provides more effective personalized rankings than other methods.

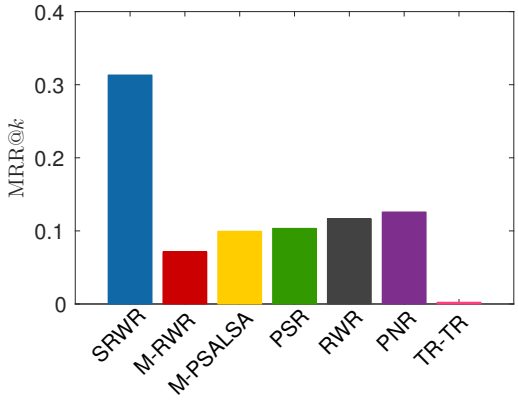


Fig. 8. MRR@k of SRWR ($k = 2,000$). The measure indicates how trolls are ranked high in a personalized distrust ranking. The MRR value of SRWR is the highest among other competitors.

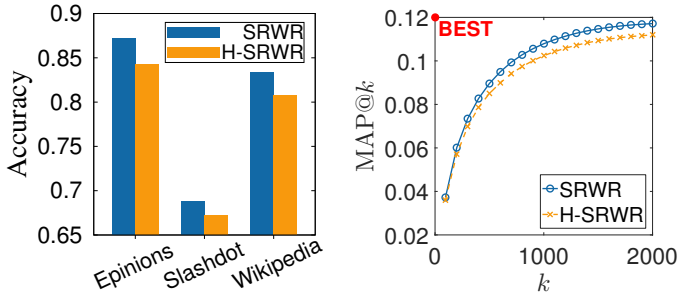
Balance attenuation factors. We adjust the balance attenuation factors of SRWR, and evaluate the sign prediction task to examine how well balance theory explains signed networks. In this experiment, we use the top-100 highest degree nodes as a test set for each network. The Epinions and the Slashdot datasets show similar results where larger values of β and γ achieve high accuracy as shown in Figures 7(a) and 7(b). Unlike these two datasets, the accuracy is high when β is small in the Wikipedia network as shown in Figure 7(c). This implies that "an enemy of my enemy is my friend" would not be correct in the network, which means balance theory does not apply well to the Wikipedia dataset. The reason is that the Wikipedia network represents votes between users to elect administrators; thus, the dataset is different from the Epinions and the Slashdot networks which are general social networks. Another observation is that the ideal balance theory does not apply to real-world signed networks because the accuracy is not the best over all datasets when $\beta = 1$ and $\gamma = 1$ (i.e., the ideal balance theory).

Troll Identification Task

We evaluate ranking models on the task of identifying trolls who behave abnormally or cause normal users to be irritated. The task is defined as follows: given a signed network and a user, identify trolls using a personalized ranking w.r.t. the user. As in the previous work (Kunegis et al., 2009), we also use the enemies of a user called *No-More-Trolls* in the Slashdot dataset as trolls. The user is an administrative account created for the purpose of collecting a troll list. There are 96 trolls in the list. Since we focus on personalized ranking in signed networks, we consider the following case: trolls are likely to be enemies of each normal user. Hence, trolls would be ranked high in a personalized distrust ranking ($\mathbf{r}^-$) w.r.t. the user. The task consists of identifying trolls based on personalized rankings for each normal user in the Slashdot network after excluding edges adjacent to *No-More-Trolls*. For each user with the minimum out-degree 1, we search trolls within the top-k distrust ranking, and measure Mean Average Precision (MAP@k), Normalized Discount Cumulated Gain (NDCG@k), Precision@k, Recall@k, and Mean Reciprocal Rank (MRR@k). Since there are no user-graded scores for the troll list, we set those scores to 1 for NDCG.

Table 3. Troll prediction results of SRWR and other models w.r.t. a normal user *"yagu"*. For each model, we show top-20 trusted and distrusted nodes. Red-colored users are trolls, a blue-colored user is a query user, and the black-colored are normal users. Note that SRWR shows the best result: in SRWR, the query user is ranked 1st in the trust ranking, and many trolls are ranked high in the distrust ranking. M-RWR and M-PSALSA provide inferior results since they rank the query user high in the distrust ranking, although the query user is the most trusted user for this task. PSR and TR-TR are not satisfactory either: they do not contain many trolls in their top distrusted users.

	SRWR (proposed)		**M-RWR**		**M-PSALSA**		**PSR**		**TR-TR**	
Rank	**Trust Ranking**	**Distrust Ranking**	**Trust Ranking**	**Distrust Ranking**	**Trust Ranking**	**Distrust Ranking**	**Trust Ranking**	**Distrust Ranking**	**Trust Ranking**	**Distrust Ranking**
1	**yagu**	**Klerck**	**yagu**	**yagu**	CleverN	**Klerck**	**yagu**	SmurfBu	**yagu**	(TK14)D
2	Bruce+P	GISGEOL	Bruce+P	TheJesu	CmdrTac	**yagu**	Uruk	Dr.Seus	KshGodd	Odder
3	CmdrTac	**Adolf+H**	CmdrTac	Fnkmast	Bruce+P	**CmderTa**	Photon+	Doctor_	TheIndi	relrelr
4	stukton	Nimrang	CleverN	Profess	John+Ca	spinloc	clump	artoo	daoine	willeyh
5	Uruk	Kafka_C	Uruk	rqqrtnb	**JonKatz**	JonKatz	TTMuskr	Juggle	SarahAn	06metzp
6	Photon+	Thinkit	Photon+	dubba-d	**yagu**	twitter	stukton	FreakyG	Jamie+Z	3p1ph4n
7	CleverN	SteakNS	stukton	drhairs	kfg	tomstde	RxScram	RunFatB	leoPetr	517714
8	RxScram	**CmderTa**	clump	howcoom	NewYork	StarMan	charlie	jmpoast	slothdo	A8bbNjw
9	clump	**%24%24%**	TTMuskr	khuber	AKAImBa	Doc+Rub	ssbg	El_Muer	hixie	ahpaway
10	TTMuskr	dada21	RxScram	Skapare	Alan+Co	NineNin	Idarubi	Ghost+H	phkamp	aidfarh
11	aphor	JonKatz	John+Ca	AsnFkr	FortKno	dada21	spotted	The+Hob	shadows	aldheor
12	TripMas	Henry+V	aphor	Laura_0	Doc+Rub	**stratja**	Golias	bananac	turg	alexibu
13	chrisd	aussers	chrisd	Nossie	TripMas	michael	Slider4	Sodade	aWalrus	Alex+Re
14	kfg	twitter	TripMas	derago	dada21	Ralph+J	Twid	peattle	dead_pe	AlfaGii
15	Hemos	fimbulv	kfg	Mitreya	miguel	Overly+	Toast	GISGEOL	gstein	AlphaJo
16	CowboyN	Quantum	Hemos	msfodde	Tackhea	Profani	Unknown	Nimrang	http	AmandaH
17	davesch	tomstde	davesch	bobthep	As+Seen	HanzoSa	nanojat	**Adolf+H**	Jamesda	AndreiK
18	John+Ca	VAXGeek	CowboyN	cramus	Rei	**%24%24%**	lucasth	Kafka_C	kris	antek9
19	dada21	**I+Am+Th**	NewYork	howcome	**Quantum**	Quantum	pitboss	SteakNS	Mournin	arbitra
20	Doc+Rub	Miguel+	dada21	deanata	grub	**SexyKel**	harlows	Thinkit	OctaneZ	Armadni



(a) Accuracy of the sign prediction task

(b) MAP@k of the troll identification task

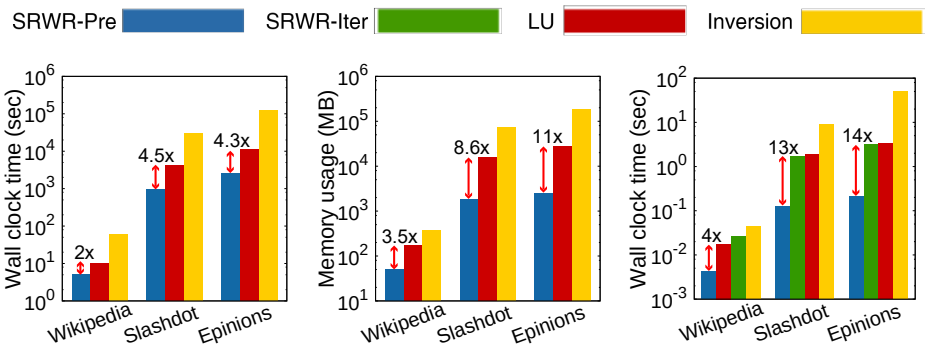
Fig. 9. Effect of the balance attenuation factors of SRWR. The performance of SRWR is better than that of H-SRWR (SRWR without using balance attenuation factors) in terms of the sign prediction and the troll identification tasks.

Results. SRWR significantly outperforms other ranking models for the troll identification task as shown in Figures 1 and 8. More trolls are identified by our proposed model within the top-k ranking according to MAP@k shown in Figure 1(a). Figures 1(c) and 1(d) indicate that SRWR achieves higher Precision@k and Recall@k for identifying trolls than other methods. SRWR provides 4× better performance than PNR, the second best one, in terms of Precision@k. Moreover, the ranking of a top ranked troll from our proposed model is higher than that of other ranking models because MRR@k of our model is the highest among other competitors as shown in Figure 8. Many trolls tend to be ranked high in our distrust ranking because SRWR achieves better MAP@k and NDCG@k than other ranking models as presented in Figures 1(a) and 1(b). We list the top-20 personalized rankings for a user called *"yagu"* in Table 3. According to the result, more trolls are ranked high in the distrust ranking from SRWR, indicating that our model is more sensitive in identifying trolls than other models. Also, the query user is ranked high in the distrust rankings from M-RWR and M-PSALSA while the user is ranked low in the distrust ranking from our model. The query user should trust himself; thus, the user should be ranked high in a trust ranking, whereas the user must be positioned low in a distrust ranking. This implies our model is more desirable than other models for personalized rankings in signed networks.

Effectiveness of Balance Attenuation Factors

We examine the effects of the balance attenuation factors of SRWR on the performance of the sign prediction and the troll identification tasks. In this experiment, we use H-SRWR ($\beta = 1$ and $\gamma = 1$) and SRWR with various balance factors for each dataset as mentioned in Section 4.1. H-SRWR indicates that we compute SRWR scores using Equation (2) which does not adopt balance attenuation factors. We measure the accuracy of the sign prediction, and MAP@k of the troll identification as described in Sections 4.2 and 4.3, respectively.

The accuracy of SRWR is better than that of H-SRWR over all datasets for the sign prediction as presented in Figure 9(a). Also, SRWR achieves better MAP@k than H-SRWR on the troll identification as shown in Figure 9(b). These results imply that adjusting the balance attenuation factors is helpful for the performance of the applications.



(a) Preprocessing time (b) Memory space for preprocessed data (c) Query time

Fig. 10. Performance of SRWR-Pre: (a) and (b) show the comparison of the preprocessing time and the memory space for preprocessed data among preprocessing methods; (c) compares the query time among all tested methods. SRWR-Pre presents the best performance compared to other preprocessing methods in terms of preprocessing time and memory efficiency. SRWR-Pre also computes SRWR scores more quickly than SRWR-Iter and the baseline preprocessing methods.

Performance of SRWR-Pre

We investigate the performance of our preprocessing method SRWR-Pre in terms of preprocessing time, memory space for precomputed data, and query time. We compare SRWR-Pre to other baseline preprocessing methods Inversion and LU as well as our iterative method SRWR-Iter. Preprocessing and query time are measured in wall-clock time, and we measure the average query time for 1,000 random seed nodes. We also measure how much memory space each preprocessing method needs for the precomputed matrices to compare memory efficiency. We omit bars for SRWR-Iter in Figures 10(a) and 10(b) because SRWR-Iter does not involve a heavy preprocessing phase (i.e., the time cost for the normalization phase of SRWR-Iter in Algorithm 1 is trivial, and the memory usage of SRWR-Iter is equal to that of the input graph).

Figures 10(a) and 10(b) show that SRWR-Pre provides better performance than LU and Inversion in terms of preprocessing time and memory space for preprocessed data. SRWR-Pre is up to 4.5× faster than the second best preprocessing method LU in terms of preprocessing time. Also, SRWR-Pre requires up to 11× less memory space than LU. Especially, our method SRWR-Pre uses 2.6GB memory for the precomputed data in the Epinions dataset while LU and Inversion require 28GB and 180GB memory, respectively. These results imply that SRWR-Pre is fast and memory-efficient compared to other preprocessing methods. SRWR-Pre also shows the fastest query speed among other competitors including our iterative method SRWR-Iter as presented in Figure 10(c). SRWR-Pre is up to 14× faster than SRWR-Iter, and up to 15× faster than the second best preprocessing method LU in the Epinions dataset. Note that SRWR-Pre computes SRWR scores for a given seed node in less than 0.3 second over all signed networks. Inversion is the slowest among the tested methods over all datasets. The main reason is that Inversion produces a very large number of non-zeros in precomputed matrices (e.g., Inversion produces about 11 billion

Table 4. Total number of non-zeros (nnz_t) in precomputed matrices for each preprocessing method. Our method SRWR-Pre generates less non-zeros in precomputed matrices than other preprocessing methods.

Dataset	A: nnz_t in SRWR-Pre	B: nnz_t in LU	C: nnz_t in Inversion	Ratio B/A	Ratio C/A
Wikipedia	3,207,758	11,257,644	23,928,232	3.51	7.46
Slashdot	119,580,272	1,032,276,955	4,817,461,830	8.63	40.29
Epinions	165,006,379	1,825,755,902	11,755,245,476	11.06	71.24

non-zeros in the Epinions dataset as presented in Table 4). These results indicate that SRWR-Pre is appropriate to serve given queries in real-time on the datasets with low memory usage compared to other methods.

Discussion. In this work, we propose two methods for SRWR: SRWR-Iter and SRWR-Pre which are iterative and preprocessing methods computing SRWR scores, respectively. SRWR-Iter does not require heavy precomputed data to compute SRWR scores. However, SRWR-Iter shows slow query speed as presented in Figure 10(c) because SRWR-Iter should perform matrix vector multiplications many times for a given seed node. On the other hand, SRWR-Pre is faster up to 14× than SRWR-Iter in term of query speed since SRWR-Pre directly computes SRWR scores from precomputed data. However, in SRWR-Pre, the values of the parameters c, β, and γ of SRWR are fixed through the preprocessing phase (Algorithm 3); thus, SRWR-Pre cannot change the parameters in the query phase (Algorithm 4). To obtain SRWR scores with the different values of the parameters, we need to perform the preprocessing phase with the parameters again. On the contrary, SRWR-Iter easily handles the change of the parameters in the query phase (Algorithm 2) without additional operations such as preprocessing. One appropriate usage for our methods is that a user uses SRWR-Iter to find proper parameters for a specific application; and then, the user exploits SRWR-Pre with the discovered parameters to accelerate the query speed in the application.

Related Works

In this section, we review related works, which are categorized into three parts: 1) ranking in unsigned networks, 2) fast ranking methods in unsigned networks, and 3) ranking in signed networks.

Ranking in unsigned networks. There are various global ranking measures based on link structure and random walk, e.g., PageRank (PR) (Page et al., 1999), HITS (Kleinberg, 1999), and SALSA (Lempel and Moran, 2001). Furthermore, personalized ranking methods are explored in terms of relevance such as Personalized PageRank (PPR) (Haveliwala, 2002), Personalized SALSA (PSALSA) (Bahmani, Chowdhury and Goel, 2010). Among these measures, RWR has received much interests and has been applied to many applications (Kang, Tong and Sun, 2012; Backstrom and Leskovec, 2011; Gleich and Seshadhri, 2012). Note that these methods are not applicable to signed graphs because they assume only positive edges; on the contrary, our model works on signed networks as well as on unsigned networks.

Fast ranking methods in unsigned networks. Tong et al. (Tong, Faloutsos and Pan, 2008) proposed an approximate method which exploits a low-rank approximation based on matrix decomposition in the preprocessing phase, and computes an RWR query from the decomposed matrices in the query phase. Fu-

jiwara et al. utilized LU factorization (Fujiwara et al., 2012) with degree ordering to speed up the RWR computation. Shin et al. (Shin et al., 2015) proposed a block elimination approach based on node reordering to accelerate RWR computation speed. Although those approaches significantly increase the performance of ranking in terms of running time, they only focus on ranking in unsigned networks. However, our preprocessing method SRWR-Pre quickly computes personalized rankings in signed networks.

Ranking in signed networks. Wu et al. (Wu et al., 2016) proposed Troll-Trust model (TR-TR) which is a variant of PageRank. In the algorithm, the trustworthiness of individual data is modeled as a probability that represents the underlying ranking values. Kunegis et al. (Kunegis et al., 2009) presented Signed spectral Ranking (SR) by extending PageRank to the case of negative edge. Shahriari et al. (Shahriari and Jalili, 2014) suggested Modified PageRank (MPR), which computes PageRank in a positive subgraph and a negative subgraph separately, and subtracts negative PageRank scores from positive ones. These models cannot explain complex relationships between negative and positive edges; in contrast, our model has the ability to account for the relationships.

In our previous work (Jung, Jin, Sael and Kang, 2016), we designed a random surfer model for ranking nodes in signed networks, and developed an iterative method for computing trust and distrust scores. However, the iterative method is not appropriate for real-time applications since the method is not fast in large signed networks as shown in Figure 10(c). Contrary to the previous method, our proposed method SRWR-Pre is the fastest for computing SRWR scores among other baselines as presented in Figure 10(c).

Conclusion

We propose Signed Random Walk with Restart, a novel model which provides personalized trust or distrust rankings in signed networks. In our model, we introduce a signed random surfer so that she considers negative edges by changing her sign for surfing on signed networks. Consequently, our model provides personalized trust or distrust rankings reflecting signed edges. Our model is a generalized version of Random Walk with Restart working on both signed and unsigned networks. We also devise SRWR-Iter and SRWR-Pre, iterative and preprocessing methods to compute SRWR scores, respectively. We experimentally show that SRWR achieves the best accuracy (up to 87%) for sign prediction, and predicts trolls 4× more accurately than other ranking models. SRWR-Pre preprocesses a signed network up to 4.5× faster, and requires 11× less memory space than other preprocessing methods; SRWR-Pre computes SRWR scores up to 14× faster than other methods.

References

Backstrom, L. and Leskovec, J. (2011), Supervised random walks: predicting and recommending links in social networks, *in* 'Proceedings of the fourth ACM international conference on Web search and data mining', ACM, pp. 635–644.

Bahmani, B., Chowdhury, A. and Goel, A. (2010), 'Fast incremental and personalized pagerank', *Proceedings of the VLDB Endowment* **4**(3), 173–184.

Boyd, S. and Vandenberghe, L. (2004), *Convex optimization*, Cambridge university press.

Cartwright, D. and Harary, F. (1956), 'Structural balance: a generalization of heider's theory.', *Psychological review* **63**(5), 277.

Duff, I. S., Grimes, R. G. and Lewis, J. G. (1989), 'Sparse matrix test problems', *ACM Transactions on Mathematical Software (TOMS)* **15**(1), 1–14.

Fujiwara, Y., Nakatsuji, M., Onizuka, M. and Kitsuregawa, M. (2012), 'Fast and exact top-k search for random walk with restart', *Proceedings of the VLDB Endowment* **5**(5), 442–453.

Gleich, D. F. and Seshadhri, C. (2012), Vertex neighborhoods, low conductance cuts, and good seeds for local community methods, *in* 'Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 597–605.

Haveliwala, T. H. (2002), Topic-sensitive pagerank, *in* 'Proceedings of the 11th international conference on World Wide Web', ACM, pp. 517–526.

Jung, J., Jin, W., Sael, L. and Kang, U. (2016), Personalized ranking in signed networks using signed random walk with restart, *in* 'IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain', pp. 973–978.
URL: *http://dx.doi.org/10.1109/ICDM.2016.0122*

Kang, U. and Faloutsos, C. (2011), Beyond 'caveman communities': Hubs and spokes for graph compression and mining, *in* 'ICDM'.

Kang, U., Tong, H. and Sun, J. (2012), Fast random walk graph kernel, *in* 'Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012.', pp. 828–838.

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

Kunegis, J., Lommatzsch, A. and Bauckhage, C. (2009), The slashdot zoo: mining a social network with negative edges, *in* 'Proceedings of the 18th international conference on World wide web', ACM, pp. 741–750.

Langville, A. N., Meyer, C. D. and Fernández, P. (2008), 'Googles pagerank and beyond: the science of search engine rankings', *The Mathematical Intelligencer* **30**(1), 68–69.

Lempel, R. and Moran, S. (2001), 'Salsa: the stochastic approach for link-structure analysis', *ACM Transactions on Information Systems (TOIS)* **19**(2), 131–160.

Leskovec, J., Huttenlocher, D. and Kleinberg, J. (2010), Predicting positive and negative links in online social networks, *in* 'Proceedings of the 19th international conference on World wide web', ACM, pp. 641–650.

Lim, Y., Kang, U. and Faloutsos, C. (2014), 'Slashburn: Graph compression and mining beyond caveman communities', *IEEE Trans. Knowl. Data Eng.* **26**(12), 3077–3089.
URL: *http://doi.ieeecomputersociety.org/10.1109/TKDE.2014.2320716*

Ng, A. Y., Zheng, A. X. and Jordan, M. I. (2001), Stable algorithms for link analysis, *in* 'Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 258–266.

Page, L., Brin, S., Motwani, R. and Winograd, T. (1999), 'The pagerank citation ranking: bringing order to the web.'.

Shahriari, M. and Jalili, M. (2014), 'Ranking nodes in signed social networks', *Social Network Analysis and Mining* **4**(1), 1–12.

Shin, K., Jung, J., Lee, S. and Kang, U. (2015), Bear: Block elimination approach for random walk with restart on large graphs, *in* 'Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data', ACM, pp. 1571–1585.

Strang, G. (2006), *Linear Algebra and Its Applications*, Thomson, Brooks/Cole.
URL: *https://books.google.ie/books?id=q9CaAAAACAAJ*

Szell, M., Lambiotte, R. and Thurner, S. (2010), 'Multirelational organization of large-scale social networks in an online world', *Proceedings of the National Academy of Sciences* **107**(31), 13636–13641.

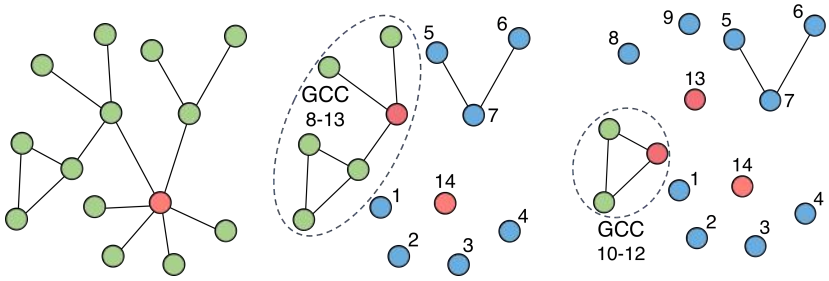
Tong, H., Faloutsos, C., Gallagher, B. and Eliassi-Rad, T. (2007), Fast best-effort pattern matching in large attributed graphs, *in* 'Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 737–746.

Tong, H., Faloutsos, C. and Pan, J.-Y. (2008), 'Random walk with restart: fast solutions and applications', *Knowledge and Information Systems* **14**(3), 327–346.

Van Loan, C. F. (1996), 'Matrix computations (johns hopkins studies in mathematical sciences)'.

Wu, Z., Aggarwal, C. C. and Sun, J. (2016), The troll-trust model for ranking in signed networks, *in* 'Proceedings of the Ninth ACM International Conference on Web Search and Data Mining', ACM, pp. 447–456.

Yang, B., Cheung, W. K. and Liu, J. (2007), 'Community mining from signed social networks', *Knowledge and Data Engineering, IEEE Transactions on* **19**(10), 1333–1348.



(a) Step 1 (b) Step 2 (c) Step 3

Fig. 11. Node reordering based on hub-and-spoke method when $\lceil tn \rceil = 1$ where $\lceil tn \rceil$ indicates the number of selected hubs at each step, and t is the hub selection ratio ($0 < t < 1$). Red nodes are hubs; blue nodes are spokes that belong to the disconnected components; green colored are nodes that belong to the giant connected component. At Step 1 in (a), the method disconnects a hub node, and assigns node ids as shown in (b). The hub node gets the highest id (14), the spoke nodes get the lowest ids ($1 \sim 7$), and the GCC gets the middle ids ($8 \sim 13$). The next iteration starts on the GCC in (b), and the node ids are assigned as in (c)

Appendix

Details of the hub-and-spoke reordering method

SlashBurn (Kang and Faloutsos, 2011; Lim et al., 2014) is a node reordering algorithm which concentrates non-zero entries of the adjacency matrix of a given graph based on the hub-and-spoke structure. Let n be the number of nodes in a graph, and t be the hub selection ratio whose range is between 0 and 1 where $\lceil tn \rceil$ indicates the number of nodes selected by SlashBurn as hubs. For each iteration, SlashBurn disconnects $\lceil tn \rceil$ high degree nodes, called *hub nodes*, from the graph; then the graph is split into the giant connected component (GCC) and the disconnected components. The nodes in the disconnected components are called *spokes*. Then, SlashBurn reorders nodes such that the hub nodes get the highest ids, the spokes get the lowest ids, and the nodes in the GCC get the ids in the middle. SlashBurn repeats this procedure on the GCC recursively until the size of GCC becomes smaller than $\lceil tn \rceil$. After SlashBurn is done, the reordered adjacency matrix contains a large and sparse block diagonal matrix in the upper left area, as shown in Figure 5. Figure 11 depicts the procedure of SlashBurn when $\lceil tn \rceil = 1$.

Sum of positive and negative SRWR scores

Lemma 5. *Consider the recursive equation* $\mathbf{p} = (1-c)|\tilde{\mathbf{A}}|^\top \mathbf{p} + c\mathbf{q}$ *where* $\mathbf{p} = \mathbf{r}^+ + \mathbf{r}^-$ *and* $|\tilde{\mathbf{A}}|^\top$ *is a column stochastic matrix. Then* $\mathbf{1}^\top \mathbf{p} = \sum_i \mathbf{p}_i = 1$.

Proof. By multiplying both sides of the recursive equation by $\mathbf{1}^\top$, the equation is represented as follows:

$$\mathbf{p} = (1-c)|\tilde{\mathbf{A}}|^\top \mathbf{p} + c\mathbf{q} \Leftrightarrow \mathbf{1}^\top \mathbf{p} = (1-c)\mathbf{1}^\top |\tilde{\mathbf{A}}|^\top \mathbf{p} + c\mathbf{1}^\top \mathbf{q}$$

Note that $\mathbf{1}^\top |\tilde{\mathbf{A}}|^\top = (|\tilde{\mathbf{A}}|\mathbf{1})^\top$, and $|\tilde{\mathbf{A}}|$ is a row stochastic matrix; thus, $(|\tilde{\mathbf{A}}|\mathbf{1})^\top = \mathbf{1}^\top$. Hence, the above equation is represented as follows:

$$\mathbf{1}^\top \mathbf{p} = (1-c)\mathbf{1}^\top |\tilde{\mathbf{A}}|^\top \mathbf{p} + c\mathbf{1}^\top \mathbf{q} \Leftrightarrow \mathbf{1}^\top \mathbf{p} = (1-c)\mathbf{1}^\top \mathbf{p} + c \Leftrightarrow \mathbf{1}^\top \mathbf{p} = 1 \qquad \square$$