Look Before You Leap: Confirming Edge Signs in Random Walk with Restart for Personalized Node Ranking in Signed Networks

Wonchang Lee*
Hanyang University
Seoul, Korea
wonchang24@hanyang.ac.kr

Yeon-Chang Lee*
Hanyang University
Seoul, Korea
lyc0324@hanyang.ac.kr

Dongwon Lee
The Pennsylvania State University
University Park, PA, USA
dongwon@psu.edu

Sang-Wook Kim†
Hanyang University
Seoul, Korea
wook@hanyang.ac.kr

ABSTRACT

In this paper, we address the *personalized node ranking* (PNR) problem for signed networks, which aims to rank nodes in an order most relevant to a given seed node in a signed network. The recently-proposed PNR methods introduce the concept of the *signed random surfer*, denoted as SRSurfer, that performs the score propagation between nodes using the balance theory. However, in real settings of signed networks, *edge relationships often do not strictly follow the rules of the balance theory*. Therefore, SRSurfer-based PNR methods frequently perform *incorrect score propagation* to nodes, thereby degrading the accuracy of PNR. To address this limitation, we propose a novel random-walk based PNR approach with *sign verification*, named as OBOE (lOok Before yOu lEap). Specifically, OBOE carefully *verifies* the score propagation of SRSurfer by using the topological features of nodes. Then, OBOE *corrects* all incorrect score propagation cases by exploiting the statistics of a given network. The experiments on 3 real-world signed networks show that OBOE consistently and significantly outperforms 5 competing methods with improvement up to 13%, 95%, and 249% in top-k PNR, bottom-k PNR, and troll identification tasks, respectively. All OBOE codes and datasets are available at: http://github.com/wonchang24/OBOE.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

personalized node ranking; signed networks; balance theory

ACM Reference Format:
Wonchang Lee, Yeon-Chang Lee, Dongwon Lee, and Sang-Wook Kim. 2021. Look Before You Leap: Confirming Edge Signs in Random Walk with Restart for Personalized Node Ranking in Signed Networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462923

*Two first authors have contributed equally to this work.
†Corresponding author.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462923

1 INTRODUCTION

Given a seed node in a network, the *personalized node ranking* (PNR) problem is to rank the remaining nodes in an order most relevant to the *seed node* by considering both the structure of the network and the connectivity with the seed node. Unlike the traditional node ranking problem [16, 23, 29], the PNR problem ranks nodes from the view point of a given seed node. In this sense, solutions [9, 15, 35, 40] to the PNR problem can be utilized in a variety of applications that need personalization such as friend recommendation and targeted marketing. Popular PNR methods include *random-walk with restart* (RWR) [34] and personalized SALSA (PSALSA) [1].

Recently, there has been a surge of interest on *signed networks* with both positive and negative edges between nodes [25, 38]. For example, sites such as Slashdot (a technology news site) or Epinions (a now-defunct consumer review site) allow users to decide whether users *trust* (*i.e.,* positive edges) or *distrust* (*i.e.,* negative edges) each other. In such a setting, the edge signs provide rich semantics between nodes [6, 10, 18, 20]. However, the aforementioned PNR methods were designed for (unsigned) networks with only positive edges between nodes. For this reason, many prior works have extended such PNR methods for signed networks–*e.g.,* FriendTNS [33], OPT+GAUC [31], SRWR [13, 14], and SSRW [27].

In particular, two recently-proposed and also best-performing PNR methods, *i.e.,* SRWR [13, 14] and SSRW [27], extend the existing RWR models by introducing the concept of the *signed random surfer*, denoted by SRSurfer in this paper. Basically, SRSurfer traverses the edges starting from a seed node n_x in a signed network while propagating the positive/negative scores of n_x to the visited nodes n_y using the *balance theory* [2], a well-known theory in psychology. This theory states that social relationships follow four rules: (R1) a friend of my friend is my friend; (R2) a friend of my enemy is my enemy; (R3) an enemy of my friend is my enemy; (R4) an enemy of my enemy is my friend.

Specifically, SRSurfer predicts an edge sign between two nodes n_x, n_y by analyzing the combination of edge signs along the random-walk path from n_x to n_y based on the balance theory. Then, if the predicted sign is positive (resp. negative), SRSurfer propagates the

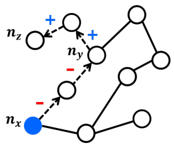

Figure 1: A signed network: dotted lines indicate the edges walked by SRSurfer.

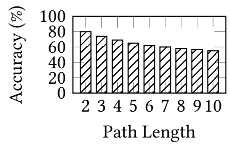

Figure 2: Prediction accuracy of SRSurfer per path length.

positive (resp. negative) score of n_x to n_y. As shown in Figure 1, when SRSurfer walks on two negative edges from n_x to n_y, it propagates a positive score of n_x to n_y based on R4 of the balance theory. Here, we note that this type of score propagation is performed under the assumption that the decades-old balance theory always holds in real settings. However, in real settings of signed networks, *edge relationships often do not strictly follow the rules of the balance theory* [7]; this observation will be elaborated in Section 3.

In this sense, the edge signs predicted by SRSurfer may be *inaccurate*. For instance, in the above example, n_x and n_y may not be a friend because the enemies of my enemy are not always my friends in the real world. Furthermore, as the length of the path between two nodes increases (*i.e.,* the path between n_x and n_z in Figure 1), the number of edge signs used for prediction also increases, rendering the accurate sign predictions more challenging. As a demonstration, Figure 2 shows that the prediction accuracy of SRSurfer using the Wikipedia dataset decreases rapidly as the path lengths increase. This result clearly indicates that SRSurfer fails to predict edge relationships accurately. In other words, SRSurfer-based PNR methods frequently perform *incorrect score propagation* to nodes, thereby degrading the accuracy of node rankings.

To address this limitation, in this paper, we aim to carefully *verify* the score propagation of SRSurfer and then *correct* incorrect score predictions. Toward this end, we propose a novel random-walk based PNR approach with *sign verification*, named as **OBOE** (lOok Before yOu lEap). Existing SRSurfer-based PNR methods perform the score propagation based on the balance theory as follows: if n_y and n_z are connected by a positive edge (resp. a negative edge), the methods propagate n_y's positive/negative scores to n_z's positive/negative (resp. negative/positive) scores, respectively. On the other hand, OBOE carefully validates whether such score propagation is *trustworthy or not* "before" propagating n_y's scores to n_z. To this end, OBOE first predicts a sign between the seed node n_x and n_z based on their 23 *topological features* such as degree distribution and triad types [24, 26].[1] Then, if the sign prediction turns out to be trustworthy, OBOE degenerates to a regular SRSurfer model and propagates scores based on the balance theory. Otherwise, OBOE uses a novel score propagation strategy that exploits the statistics related to the edge signs of all triangles in a given signed network. By iteratively performing the aforementioned process (*i.e.,* verification and score propagation), OBOE propagates the positive/negative scores of the seed node n_x to all other nodes. Finally, after convergence, OBOE ranks all nodes except for n_x based on their positive and negative scores.

In designing, implementing, and validating these ideas, in this paper, our contributions are summarized as follows:

- We demonstrate the limitation of SRSurfer-based PNR methods in propagating scores of nodes based on the balance theory.
- We propose a novel random-walk based PNR approach with a sign verification, named as OBOE.
  - We design a strategy to validate the score propagation based on the balance theory by exploiting topological features of nodes.
  - We design a strategy to leverage the statistics of a given network to replace the untrustworthy score propagation.
- We validate the effectiveness of OBOE via extensive experiments using three real-world datasets.
  - Specifically, OBOE dramatically improves the accuracy of top-k PNR/bottom-k PNR/troll identification tasks up to 13%/95%/249%, respectively, over the best performer among five competing methods.

The rest of this paper is organized as follows: Section 2 reviews existing PNR methods. Section 3 demonstrates the limitation of SRSurfer-based PNR methods and Section 4 presents our proposed approach in detail. Section 5 validates the effectiveness of the proposed approach through extensive experiments. Finally, Section 6 summarizes and concludes the paper.

2 RELATED WORK

In this section, we briefly review two families of existing PNR methods: one for unsigned networks and the other for signed networks.

PNR for Unsigned Networks. First, RWR [34], PSALSA [1], and MPR [8] analyze the structure of a given network by designing their own random-walk models. They traverse the edges starting from a seed node in the network while propagating the scores of the seed node to other nodes. Finally, they rank nodes, except for the seed node, based on their scores. Next, PALE [28] and RDL [36] represent the nodes in a given network as low-dimensional vectors by employing low-rank models. Then, they rank nodes, except for a seed node, based on a distance function that exploits the vector of the seed node and that of each remaining node. However, none of these methods took edge signs into consideration in their designs.

PNR for Signed Networks. To address this limitation, various PNR methods for signed networks have been proposed. In the early days of research, FriendTNS [33] and Zhu et al. [42] designed heuristic measures that calculate relevance scores between nodes in consideration of edge signs. Then, both methods rank the nodes based on the relevance scores between the seed node and remaining nodes. Next, Opt-GAUC [31] proposed a matrix factorization model that learns existent positive/negative edges between nodes and further learns non-existent edges as no-relation edges.

Recently, several studies such as ModifiedRWR [30], SRWR [13, 14], and SSRW [27] have extended the existing RWR models. First, ModifiedRWR [30] computes positive/negative RWR scores of nodes by performing RWR on the positive/negative subgraph, respectively, and then ranks nodes by subtracting negative RWR scores from positive ones. Next, SRWR [13, 14] and SSRW [27] compute positive/negative scores of nodes based on SRSurfer and then rank nodes in the same way as in ModifiedRWR. Specifically, SRWR introduced the balance attenuation factors β, γ into SRSurfer to consider the uncertainty for two rules (*i.e.,* R2 and R4) of the balance theory. However, the factors (1) apply to all (trustworthy and

[1]Note that here we consider n_z's relationship with the seed node n_x (not n_y), as OBOE is propagating the scores of n_y received from n_x to n_z.

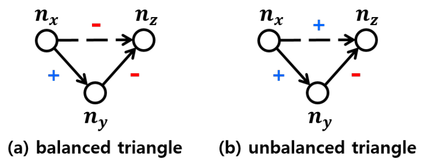

Figure 3: Examples of balanced/unbalanced triangles.

untrustworthy) score propagation for R2 and R4, and (2) do not cover the remaining two rules (*i.e.,* R1 and R3) of the balance theory. On the other hand, SSRW incorporated the social strengths into SRSurfer, which captures a node's different preferences towards different neighbors. Note that SSRW does not employ the balance attenuation factors. In summary, both SRSurfer-based PNR methods frequently perform incorrect score propagation to nodes, thereby adversely affecting the accuracy of PNR.

Lastly, we can perform PNR based on the network embedding methods for signed networks, denoted as signed NE [5, 21, 37, 39, 41]. Specifically, the signed NE methods represent the nodes of a given signed network by vectors in a low-dimensional embedding space so that the vectors preserve structural and semantic properties in the network. That is, they attempt to represent the nodes with positive edges to be close and those with the negative edges to be distant in the embedding space. Literature [5, 21, 37, 39, 41] has shown that the low-dimensional vectors can be used as effective features of nodes in solving various downstream tasks including node ranking and recommendation [3, 4, 19]. For instance, in terms of PNR, we rank the nodes using the vectors of the seed node and those of the remaining nodes.

3 MOTIVATION

In this section, we present SRSurfer [13, 14] in detail and demonstrate its limitation via experiments over real-world datasets.

SRSurfer predicts positive scores r_y^+ and negative scores r_y^- of all nodes n_y from the perspective of a seed node n_x. If $x = y$, the initial values of r_y^+ and r_y^- are set to 1 and 0, respectively. Otherwise, both are set to 0. Then, SRSurfer propagates r_x^+/r_x^- of n_x to all other nodes. Suppose that SRSurfer walks to a node n_z in the neighborhood of a node n_y with a probability of $(1 - c)$. In this case, SRSurfer propagates r_y^+/r_y^- of n_y to n_z based on the balance theory as follows: if n_y and n_z are connected by a positive edge (resp. a negative edge), it propagates r_y^+/r_y^- of n_y to r_z^+/r_z^- (resp. r_z^-/r_z^+) of n_z, respectively. In addition, SRSurfer restarts at n_x with a probability c for personalization w.r.t n_x. This probability c is empirically determined.

Given a seed node n_x, the aforementioned score propagation can be formulated as follows [13, 14]:

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

where $\mathbf{r}^+/\mathbf{r}^-$ represent positive/negative score vectors w.r.t. n_x whose y-th elements are r_y^+/r_y^- of n_y, respectively. $\tilde{\mathbf{A}}^+/\tilde{\mathbf{A}}^-$ indicate positive/negative semi-row normalized matrices that contain all positive/negative values in the adjacency matrix $\mathbf{A}$ of a given signed network, respectively. $\mathbf{q}$ is an unit vector whose x-th element (*i.e.,* seed node n_x) is 1 and all other elements are 0. Finally, c is a restart probability. Then, SRSurfer iteratively updates $\mathbf{r}^+$ and $\mathbf{r}^-$ via (Eq. 1) until $\mathbf{r}^+$ and $\mathbf{r}^-$ converge. Finally, SRSurfer computes the

Table 1: Ratios of balanced/unbalanced triangles per type

Triangles	Prior / Posterior	(+, +) +	(+, +) −	(+, −) +	(+, −) −	(−, +) +	(−, +) −	(−, −) +	(−, −) −
Dataset	Wikipedia	92%	8%	72%	28%	62%	38%	50%	50%
	Slahsdot	98%	2%	34%	66%	58%	42%	54%	46%
	Epinions	99%	1%	38%	62%	53%	47%	41%	59%
Balanced?		○	×	×	○	×	○	○	×

ranking score vector $\mathbf{r}^d$ of n_x as follows:

$$\mathbf{r}^d = \mathbf{r}^+ - \mathbf{r}^-, \tag{2}$$

where r_y^d indicates a final ranking score for n_y w.r.t n_x.

Note that the score propagation of SRSurfer totally depends on the predicted signs by the balance theory. As alluded in Section 1, however, edge relationships often do not strictly follow the rules of the balance theory. To show the evidence of this claim in real-world signed networks (*i.e.,* Wikipedia, Slashdot, and Epinions), we first sample all the triangles (n_x, n_y, n_z) where the edge directions between three nodes satisfy the transitivity. For instance, if n_x points to n_y, n_y points to n_z, and n_x points to n_z, the triangle (n_x, n_y, n_z) satisfies the transitivity. Then, we examine how much the triangles follow the rules of the balance theory. That is, given two signs (*i.e., prior signs*) in a triangle, we check whether the remaining sign (*i.e., posterior sign*) matches the rules of the balance theory (*e.g.,* a *balanced triangle* in Figure 3-(a)) or not (*e.g.,* a *unbalanced triangle* in Figure 3-(b)).

Table 1 shows the ratios of balanced and unbalanced triangles among all triangles in three signed networks, according to four types of prior signs – *i.e.,* (+, +), (+, −), (−, +), (−, −). We observe that (+, +) type follows the rules of the balance theory considerably, whereas other types often do *not* follow the rules strictly. For instance, for (+, −) type in Wikipedia or for (−, −) type in Epinions networks, 72% and 59% of triangles do not follow the balance theory (i.e., unbalanced triangles), respectively. This result clearly demonstrates that the balance theory does not always hold in real settings. Therefore, score propagation models that strictly follows the balance theory (*e.g.,* SRSurfer-based PNR methods) are likely to contain substantial amount of incorrect score propagation.

Next, further, we test *how much one can improve the sign prediction accuracy by not strictly following the balance theory.* To this end, we regard 80% of the edges in a signed network as a training set, and the remaining 20% as a test set. Then, we sample all *unbalanced* triangles (n_x, n_y, n_z) from the training set. Next, we perform sign prediction on the test set using SRSurfer. Here, we made two variants, denoted as Original SRSurfer and Modified SRSurfer, of predicting the signs of posterior edges (n_x, n_z) when SRSurfer walks on two prior edges (n_x, n_y), (n_y, n_z) of a *sampled unbalanced triangle* (n_x, n_y, n_z) in turn. Specifically, whenever two variants encounter two prior edges of the sampled unbalanced triangles during the random-walk process, the sign of each posterior edge is predicted differently as follows:[2]

- **Original SRSurfer** incorrectly predicts the sign of a posterior edge by *blindly* following the balance theory. For instance, in Figure 3-(b), it predicts the sign of (n_x, n_z) as negative. However,

[2] Such predictions are difficult to implement in terms of matrix-vector multiplication as shown in (Eq. 1). Thus, we performed both original/modified SRSurfer in a random-walk manner.

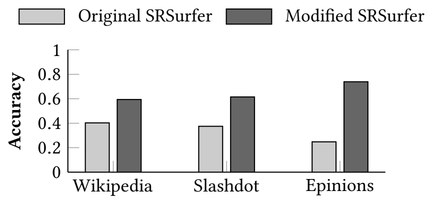

Figure 4: Accuracies of sign prediction by original and modified SRSurfer.

Table 2: Notations used in this paper

Notation	Description
$\mathcal{G}$	signed network
$\mathcal{N}$	set of nodes
$\mathcal{E}^+$	set of positive edges
$\mathcal{E}^-$	set of negative edges
n_x	seed node
c	restart probability
r_y^d	final ranking score of n_y w.r.t n_x
r_y^+/r_y^-	positive/negative scores of n_y w.r.t n_x
$sign(n_x, n_y)$	real edge sign between n_x and n_y
$sign_{feature}(n_x, n_y)$	predicted edge sign between n_x and n_y
$C(sign_{feature}(n_x, n_y))$	confidence score for the predicted sign

in the case of Wikipedia data, for instance, 72% of triangles does not follow this pattern, causing substantial errors in prediction.

- **Modified SRSurfer** correctly predicts the sign of a posterior edge by *not* following the balance theory. For instance, in Figure 3-(b), it predicts the sign of (n_x, n_z) as positive, which happens to be in agreement with 72% of triangles in Wikipedia, for instance.

Figure 4 shows the prediction accuracies of the original and modified SRSurfer models for *the edge signs on a test set*. We confirm that the modified SRSurfer significantly outperforms the original SRSurfer. More specifically, for Wikipedia/Slashdot/Epinions, the modified SRSurfer dramatically improves the accuracies of the original SRSurfer by 47.4%/64.0%/198.0%, respectively. The result indicates that the accuracy of sign predictions can be significantly improved by *correcting the incorrect sign predictions by the balance theory*. Therefore, we here conclude that (1) edge relationships often do not strictly follow the rules of the balance theory (as shown in Table 1) and (2) the balance theory results in incorrect sign prediction of SRSurfer (as shown in Figure 4).

4 THE PROPOSED APPROACH: OBOE

To address the limitation of SRSurfer-based PNR methods, we propose a new random-walk based PNR approach with a sign verification, named as OBOE. In Section 4.1, we first formulate the PNR problem for signed networks and present the overall procedure of our OBOE. In Section 4.2, we describe two key ideas, *i.e.,* sign verification and score propagation, of OBOE in detail. In Section 4.3, we formulate the iteration of OBOE as a matrix-vector multiplication form and show its iterative algorithm. Lastly, in Section 4.4, we discuss the convergence of OBOE.

4.1 Overall Procedure

The PNR problem for signed networks is formulated as follows: let $\mathcal{G} = (\mathcal{N}, \mathcal{E}^+, \mathcal{E}^-)$ be a given signed network, where $\mathcal{N} = \{n_1, n_2, \cdots, n_m\}$ represents a set of m nodes and $\mathcal{E}^+$ and $\mathcal{E}^-$ represent the sets of positive and negative edges, respectively. Note that $\mathcal{E}^+ \cap \mathcal{E}^- = \emptyset$, a node pair cannot have both positive and negative edges simultaneously. Given a seed node n_x, PNR methods for signed networks aim to output ranking of the nodes except for n_x by analyzing both the structure of the network and their connectivity with n_x. Table 2 summarizes a list of notations used in this paper.

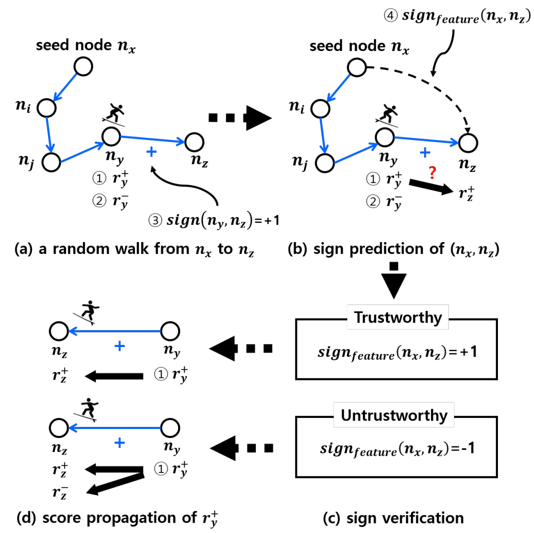

Figure 5: The procedure of score propagation of OBOE.

We present the overall procedure of OBOE. Given a seed node n_x, OBOE sets the positive/negative scores r_y^+/r_y^- of all nodes n_y from the perspective of n_x. r_y^+ and r_y^- represents the likelihood (measured by OBOE) to which the edge relationship between n_x and n_y has a positive sign (*i.e.,* friend) and a negative sign (*i.e.,* enemy), respectively. If $x = y$, the initial values of r_y^+ and r_y^- are set to 1 and 0, respectively. Otherwise, both are set to 0. Then, OBOE propagates r_x^+/r_x^- of n_x to all other nodes. Toward this end, the random surfers of OBOE start from n_x and then walk along the outgoing edges with a probability of $(1 - c)$, while going back to n_x with a probability of c.

Now, we describe the process of the score propagation of OBOE with Figure 5. As shown in Figure 5-(a), suppose that a random surfer is currently at a node n_y and n_y has already received the scores from the previously-visited node n_j by OBOE's score propagation strategy (to be explained later). When the surfer walks to a node n_z in the neighborhood of n_y with $(1 - c)$, OBOE propagates both r_y^+ (① in Figure 5-(a)) and r_y^- (② in Figure 5-(a)) of n_y to n_z. We here explain the propagations of r_y^+ since both propagation of r_y^+ and r_y^- are similar. Basically, if an edge sign, $sign(n_y, n_z)$ (③ in Figure 5-(a)), between n_y and n_z is positive (resp. negative), OBOE propagates r_y^+ to r_z^+ (resp. r_z^-) in the same way as SRSurfer. As demonstrated in Section 3, however, score propagation models based on SRSurfer are likely to contain a substantial amount of incorrect score propagation (Table 1).

To validate the score propagation from n_y to n_z based on the balance theory, OBOE predicts an edge sign, $sign_{feature}(n_x, n_z) \in \{+1, -1\}$(④ in Figure 5-(b)), between n_x and n_z with additional information. Toward this end, OBOE exploits 23 topological features of n_x and n_z. Then, OBOE examines whether both $sign_{feature}(n_x, n_z)$ and the sign of n_z's score, to which r_y^+ is propagated by SRSurfer, (*e.g.*, r_z^+ in the case of Figure 5) are consistent or not (Figure 5-(c)): if it is consistent, OBOE regards the score propagation as trustworthy and thus performs that propagation (the upper case in Figure 5-(d)). Otherwise, OBOE regards it as untrustworthy and uses a novel score propagation (the lower case in Figure 5-(d)). Specifically, OBOE propagates r_y^+ to both scores r_z^+ and r_z^-, *not a score determined by SRSurfer*. The intuition behind this idea is that the edge relationship between n_x and n_z are *uncertain*. Therefore, OBOE should set the ratios at which r_y^+ is propagated to r_z^+ and r_z^-. In this paper, we utilize the ratios of the prior and posterior signs of triangles in a given signed network, to be elaborated in Section 4.2. OBOE also performs the same process as above for the propagation of r_y^-.

So far, we have presented a situation in which OBOE propagates r_y^+/r_y^- of a node n_y to r_z^+/r_z^- of another node n_z. In OBOE at every iteration, each node propagates its own scores to its outgoing edges and receives the propagation from its incoming edges. OBOE iteratively performs this score propagation until the scores of all nodes converge. Finally, after convergence, OBOE ranks nodes except for n_x by subtracting their negative scores from their positive ones.

4.2 Sign Verification and Score Propagation

We describe the key ideas of OBOE in detail. Given a seed node n_x, a currently-visiting node n_y, and a next-visiting node n_z, OBOE performs the following two steps: (STEP 1) sign verification and (STEP 2) score propagation. OBOE first *verifies* the score propagation from n_y to n_z by directly exploiting the *relationship between* n_x *and* n_z and then *classifies* it into *trustworthy* or *untrustworthy* one. Next, OBOE performs the trustworthy one as it is, while it *corrects* the untrustworthy one by performing a *safe* score propagation *exploiting the statistics of a given network.*

STEP 1: Sign Verification. In this step, OBOE predicts a $sign_{feature}$ (n_x, n_z) between n_x and n_z based on their topological features and validates whether the score propagation from n_y to n_z is trustworthy or not by using $sign_{feature}(n_x, n_z)$.

For sign prediction, we employ a well-known feature-based method, FExtra [24, 26]. For all node pairs (n_x, n_y) in a network, we first construct a vector $\mathbf{f}_{x,y} = \{f_1, \cdots, f_{23}\}$ consisting of values for 23 topological features for each of the pairs. Specifically, we first exploit 7 features related to nodes' degrees: (1) the number of outgoing positive edges of n_x, (2) the number of outgoing negative edges of n_x, (3) the number of incoming positive edges of n_y, (4) the number of incoming negative edges of n_y, (5) the total number of common neighbors of n_x and n_y, (6) the total out-degree of n_x, and (7) the total in-degree of n_y. In addition, we exploit features related to 16 distinct types of triads containing n_x, n_y, and their common neighbors n_z in the network: the edge between n_x and n_z can be in either direction and of either sign, and the edge between n_y and n_z can also be in either direction and of either sign; this leads to $2 \cdot 2 \cdot 2 \cdot 2 = 16$ possibilities. Thus, we set the number of triads of each type to the value of the corresponding feature.

Next, we train a logistic regression classifier model based on the $\mathbf{f}_{x,y}$ for node pairs (n_x, n_y) with real edge signs in the network. Using the learned classifier, we finally predict the $sign_{feature}(n_x, n_z)$ for all node pairs (n_x, n_z) without real edge signs in the network. During the prediction process, FExtra also computes the *confidence* score $C(sign_{feature}(n_x, n_z)) \in [0, 1]$ for the predicted sign

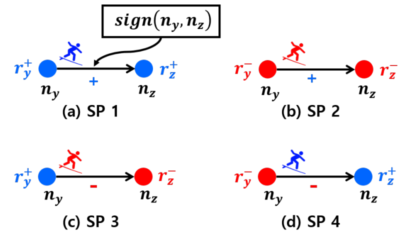

Figure 6: Score propagation based on the balance theory.

$sign_{feature}$ (n_x, n_z). For instance, if $sign_{feature}(n_x, n_z) = +1$ and $C(sign_{feature}(n_x, n_z)) = 0.95$, it indicates FExtra predicted the sign between n_x and n_z as positive and is quite confident for that prediction. Note that, before OBOE performs the random-walk process, it predicts $sign_{feature}(n_x, n_z)$ for all node pairs (n_x, n_z) without real edge signs in advance as a preprocessing task.

Now, OBOE verifies the score propagation from n_y to n_z using the $sign_{feature}(n_x, n_z)$ and the $C(sign_{feature}(n_x, n_z))$. Note that we exploit the relationship between the seed node n_x (not n_y) and n_z, as OBOE is propagating the scores of n_y *received from* n_x to n_z. Here, there are four possible score propagations (SP) from n_y to n_z as follows (Figure 6):

- If $sign(n_y, n_z) = +1$, r_y^+ is propagated to r_z^+ (**SP 1** in Figure 6-(a)) and r_y^- is propagated to r_z^- (**SP 2** in Figure 6-(b)).
- If $sign(n_y, n_z) = -1$, r_y^+ is propagated to r_z^- (**SP 3** in Figure 6-(c)) and r_y^- is propagated to r_z^+ (**SP 4** in Figure 6-(d)).

To validate the above score propagations, OBOE checks the following two conditions: (1) does the $sign_{feature}(n_x, n_z)$ and the sign of n_z's score, to which n_y's score is propagated, consistent? and (2) does the $C(sign_{feature}(n_x, n_z))$ exceed a predefined threshold? For (1), OBOE considers the score propagation in which the two signs are consistent as trustworthy. For instance, if $sign_{feature}(n_x, n_z) = +1$, OBOE considers the SP 1/4 (resp. the SP 2/3) where n_y's scores are propagated to r_z^+ (resp. r_z^-) as *trustworthy* (resp. *untrustworthy*), respectively.[3] For (2), OBOE finally uses only the score propagation considered trustworthy by the confident $sign_{feature}(n_x, n_z)$. In the above example, if $C(sign_{feature}$ $(n_x, n_z) = +1)$ is lower than the predefined threshold, OBOE does not regard the SP 1/4 as trustworthy. In this paper, we introduce a threshold β_+ for $C(sign_{feature}$ $(n_x, n_z) = +1)$ and another threshold β_- for $C(sign_{feature}$ (n_x, n_z) $= -1)$ *separately*. We will analyze the sensitivity of OBOE to β_+ and β_- in Section 5.2.

Here, there remains a question on whether the $sign_{feature}(n_x, n_z)$ is more reliable than the sign predicted by SRSurfer. To address this concern, we confirmed that FExtra consistently and significantly improves the accuracy of sign prediction based on SRSurfer in our preliminary experiments. Thanks to (STEP 1) of OBOE, we can detect untrustworthy score propagation. In Section 5.2, we confirm that such untrustworthy score propagation occurs considerably but is detected via our OBOE.

STEP 2: Score Propagation. In this step, OBOE performs different score propagation from n_y to n_z, according to trustworthy or untrustworthy one. OBOE first normalizes the r_y^+/r_y^- of n_y to the number of n_y's outgoing edges $|O_y|$ as follows:

$$r_y^+ = \frac{r_y^+}{|O_y|}, \qquad r_y^- = \frac{r_y^-}{|O_y|}. \tag{3}$$

[3]We can handle the case of $(sign_{feature}(n_x, n_z) = -1)$ as well in the same way.

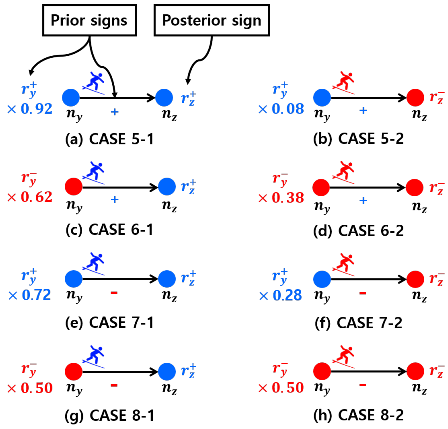

Figure 7: Score propagation strategy of OBOE (Wikipedia).

Again, note that r_y^+ and r_y^- represents the degree to which the edge relationship between n_x and n_y has a positive sign and a negative sign, respectively. Then, OBOE propagates n_y's r_y^+/r_y^- to n_z by considering the following eight cases: (1) **CASE 1/5** regard **SP 1** as trustworthy/untrustworthy, respectively; (2) **CASE 2/6** regard **SP 2** as trustworthy/untrustworthy, respectively; (3) **CASE 3/7** regard **SP 3** as trustworthy/untrustworthy, respectively; (4) **CASE 4/8** regard **SP 4** as trustworthy/untrustworthy, respectively.

For CASE 1/2/3/4 (*i.e.,* trustworthy score propagation), OBOE performs each score propagation as it is. On the other hand, for CASE 5/6/7/8 (*i.e.,* untrustworthy score propagation), we assume that the edge relationship between n_x and n_z are uncertain. Therefore, OBOE uses a different score propagation strategy, instead of each (untrustworthy) score propagation. For CASE 5/7 (*i.e.,* for propagating r_y^+ to r_z), OBOE propagates r_y^+ to both r_z^+ and r_z^- while, for CASE 6/8 (*i.e.,* for propagating r_y^- to r_z), OBOE propagates r_y^- to both r_z^+ and r_z^-.

For determining ratios of the propagation, as a heuristic policy, OBOE utilizes the ratios (*i.e.,* Table 1 in Section 3) of the prior and posterior signs of the (real) triangles in a given signed network. Specifically, OBOE regards (1) the sign of n_y's score (*i.e.,* positive and negative one for CASE 5/7 and CASE 6/8, respectively) and (2) $sign(n_y, n_z)$ (*i.e.,* positive and negative one for CASE 5/6 and CASE 7/8, respectively) as two prior signs. Also, OBOE regards the signs of n_z's score received from n_y as the posterior signs. Given two prior signs in an input network, OBOE then checks the ratios at which the posterior signs are positive and negative. Finally, OBOE propagates n_y's scores to both r_z^+/r_z^- in accordance with the observed ratio of positive/negative signs, respectively.

Figure 7 depicts the cases in which n_y's scores are propagated to n_z in the Wikipedia dataset. Each case is divided into two sub-cases according to the posterior signs. Suppose that both prior signs are given as positive as shown in Figures 7-(a) and 7-(b). In this case, from Table 1, OBOE checks that the ratios of posterior signs to positive (*i.e.,* +) and negative (*i.e.,* −) are 92% and 8% in a Wikipedia dataset, respectively. Then, OBOE propagates 92%/8% of r_y^+ to r_z^+/r_z^-, respectively. Similar to the sign predictions in (STEP 1), in general, OBOE *pre-computes* the statistics for a given signed network as part of the pre-processing task.

Thanks to (STEP 2) of OBOE, we can now correct the incorrect score propagation by referring to the statistics of a given network.

Algorithm 1 Iterative process of OBOE

Input: a seed node n_x, semi-row normalized adjacency matrices $\tilde{\mathbf{A}}^+$ and $\tilde{\mathbf{A}}^-$, restart probability c, and error tolerance ϵ
Output: a ranking vector $\mathbf{r}^d$

1: set the sub-matrices $\tilde{\mathbf{A}}_t^{++}$, $\tilde{\mathbf{A}}_t^{-+}$, $\tilde{\mathbf{A}}_t^{+-}$, $\tilde{\mathbf{A}}_t^{--}$, $\tilde{\mathbf{A}}_u^{+++}$, $\tilde{\mathbf{A}}_u^{++-}$, $\tilde{\mathbf{A}}_u^{-++}$, $\tilde{\mathbf{A}}_u^{-+-}$, $\tilde{\mathbf{A}}_u^{+-+}$, $\tilde{\mathbf{A}}_u^{+--}$, $\tilde{\mathbf{A}}_u^{--+}$, and $\tilde{\mathbf{A}}_u^{---}$
2: set $\mathbf{q}$ from n_x
3: set $\mathbf{r}^+ = \mathbf{q}$, $\mathbf{r}^- = 0$, and $\mathbf{h}' = [\mathbf{r}^+; \mathbf{r}^-]^\top$
4: **repeat**
5: compute $\mathbf{r}^+$ and $\mathbf{r}^-$ using (Eq. 4)
6: concatenate $\mathbf{r}^+$ and $\mathbf{r}^-$ into $\mathbf{h} = [\mathbf{r}^+; \mathbf{r}^-]^\top$
7: compute δ between $\mathbf{h}$ and $\mathbf{h}'$
8: update $\mathbf{h}' \leftarrow \mathbf{h}$
9: **until** $\delta < \epsilon$
10: compute $\mathbf{r}^d = \mathbf{r}^+ - \mathbf{r}^-$
11: **return** $\mathbf{r}^d$

By employing this heuristic policy, we can consider the structural property per given signed network. In Section 5.2, we show that our score propagation strategy with this policy to replace the untrustworthy score propagation is more effective than other strategies (*e.g.,* uniform propagation).

4.3 Formulation for OBOE

We formulate the iteration of OBOE *as a matrix-vector multiplication form*. Toward this end, we extend (Eq. 1) of SRSurfer as follows:

$$\begin{aligned}
\mathbf{r}^+ &= (1-c)[\{(\tilde{\mathbf{A}}_t^{++})^\top \mathbf{r}^+ + (\tilde{\mathbf{A}}_t^{--})^\top \mathbf{r}^-\} + \\
&\quad \{(\tilde{\mathbf{A}}_u^{+++})^\top \mathbf{r}^+ + (\tilde{\mathbf{A}}_u^{+-+})^\top \mathbf{r}^+ + (\tilde{\mathbf{A}}_u^{--+})^\top \mathbf{r}^- + (\tilde{\mathbf{A}}_u^{-++})^\top \mathbf{r}^-\}] + c\mathbf{q}, \\
\mathbf{r}^- &= (1-c)[\{(\tilde{\mathbf{A}}_t^{+-})^\top \mathbf{r}^+ + (\tilde{\mathbf{A}}_t^{-+})^\top \mathbf{r}^-\} + \\
&\quad \{(\tilde{\mathbf{A}}_u^{+--})^\top \mathbf{r}^+ + (\tilde{\mathbf{A}}_u^{++-})^\top \mathbf{r}^+ + (\tilde{\mathbf{A}}_u^{-+-})^\top \mathbf{r}^- + (\tilde{\mathbf{A}}_u^{---})^\top \mathbf{r}^-\}],
\end{aligned} \tag{4}$$

where $\mathbf{r}^+/\mathbf{r}^-$ represent positive/negative score vectors w.r.t. n_x. $\mathbf{q}$ is an unit vector whose x-th element (*i.e.,* seed node n_x) is 1 and all other elements are 0. Also, $\tilde{\mathbf{A}}_t^*$ and $\tilde{\mathbf{A}}_u^*$ indicate adjacency matrices for trustworthy and untrustworthy score propagations, respectively. Note that the matrices of $\tilde{\mathbf{A}}_t^*$ (*e.g.,* $\tilde{\mathbf{A}}_t^{+-}$) have two signs, while the matrices of $\tilde{\mathbf{A}}_u^*$ (*e.g.,* $\tilde{\mathbf{A}}_u^{+-+}$) have three signs. Specifically, in each of $\tilde{\mathbf{A}}_t^*$, both signs represent two prior signs. For instance, $\tilde{\mathbf{A}}_t^{+-}$ represents a situation in which r_y^+ (*i.e.,* the first prior sign) of each node n_y is propagated to each node n_z whose $sign(n_y, n_z) = -1$ (*i.e.,* the second prior sign) among the neighborhood of n_y (*i.e.,* CASE 3). Next, in each of $\tilde{\mathbf{A}}_u^*$, the first two signs are the same as those of $\tilde{\mathbf{A}}_t^*$ and the last sign is a posterior sign. For instance, $\tilde{\mathbf{A}}_u^{+-+}$ represents a situation where part of r_y^+ (*i.e.,* the first prior sign) of each node n_y is propagated to r_z^+ (*i.e.,* the posterior sign) of each node n_z whose $sign(n_y, n_z) = -1$ (*i.e.,* the second prior sign) among the neighborhood of n_y (*i.e.,* CASE 7-1). To build 12 adjacency matrices in (Eq. 4), we use the positive and negative semi-row normalized matrices $\tilde{\mathbf{A}}^+/\tilde{\mathbf{A}}^-$. Specifically, we first construct 6 (sub) adjacency matrices based on $\tilde{\mathbf{A}}^+$: $\tilde{\mathbf{A}}_t^{++}$ (CASE 1), $\tilde{\mathbf{A}}_t^{-+}$ (CASE 2), $\tilde{\mathbf{A}}_u^{+++}$ (CASE 5-1), $\tilde{\mathbf{A}}_u^{++-}$ (CASE 5-2), $\tilde{\mathbf{A}}_u^{-++}$ (CASE 6-1), and $\tilde{\mathbf{A}}_u^{-+-}$ (CASE 6-2). In addition, we construct 6 remaining (sub) adjacency matrices based on $\tilde{\mathbf{A}}^-$: $\tilde{\mathbf{A}}_t^{+-}$ (CASE 3), $\tilde{\mathbf{A}}_t^{--}$ (CASE 4), $\tilde{\mathbf{A}}_u^{+-+}$ (CASE 7-1), $\tilde{\mathbf{A}}_u^{+--}$ (CASE 7-2), $\tilde{\mathbf{A}}_u^{--+}$ (CASE 8-1), and $\tilde{\mathbf{A}}_u^{---}$ (CASE 8-2).

Finally, Algorithm 1 sketches the iterative process of OBOE. Given a seed node n_x, OBOE constructs 12 (sub) adjacency matrices using $\tilde{\mathbf{A}}^+$ and $\tilde{\mathbf{A}}^-$ (line 1). Then, OBOE sets $\mathbf{q}$ to x-th unit vector (line 2). OBOE initializes $\mathbf{r}^{+}$ and $\mathbf{r}^{-}$ and then sets $\mathbf{h}'$ by concatenating them (line 3). Next, OBOE updates $\mathbf{r}^{+}$ and $\mathbf{r}^{-}$ via (Eq. 4) and concatenates them into $\mathbf{h}$ (lines 5-6). Then, OBOE computes the error δ between $\mathbf{h}$ at the current iteration and $\mathbf{h}'$ at the previous iteration (line 7). OBOE updates $\mathbf{h}$ into $\mathbf{h}'$ for the next iteration (line 8). It performs the iterations until the error δ becomes less than the threshold ϵ (line 9). Finally, OBOE builds a vector $\mathbf{r}^{d}$ and uses it as the final ranking vector w.r.t n_x (line 10).

4.4 Convergence Analysis

We show that the iteration in Algorithm 1 converges to the solution of a linear system as follows:

THEOREM 1. *Suppose that* $\mathbf{h} = [\mathbf{r}^{+}; \mathbf{r}^{-}]^{\top}$ *and* $\mathbf{q}_x = [\mathbf{q}; \mathbf{0}]^{\top}$. *Then, the iteration for* $\mathbf{h}$ *in Algorithm 1 converges to the following solution:*

$$\mathbf{h} = c(\mathbf{I} - (1-c)\tilde{\mathbf{B}}^{\top})^{-1}\mathbf{q}_x,$$

where $\tilde{\mathbf{B}}^{\top} = \begin{bmatrix} \tilde{\mathbf{A}}_t^{++\top} + \tilde{\mathbf{A}}_u^{+++\top} + \tilde{\mathbf{A}}_u^{+-+\top} & \tilde{\mathbf{A}}_t^{--\top} + \tilde{\mathbf{A}}_u^{--+\top} + \tilde{\mathbf{A}}_u^{-++\top} \\ \tilde{\mathbf{A}}_t^{+-\top} + \tilde{\mathbf{A}}_u^{+--\top} + \tilde{\mathbf{A}}_u^{++-\top} & \tilde{\mathbf{A}}_t^{-+\top} + \tilde{\mathbf{A}}_u^{-+-\top} + \tilde{\mathbf{A}}_u^{---\top} \end{bmatrix}$.

PROOF. (Eq. 4) is represented as follows:

$$\begin{bmatrix} \mathbf{r}^{+} \\ \mathbf{r}^{-} \end{bmatrix} = (1-c) \begin{bmatrix} \tilde{\mathbf{A}}_t^{++\top} + \tilde{\mathbf{A}}_u^{+++\top} + \tilde{\mathbf{A}}_u^{+-+\top} & \tilde{\mathbf{A}}_t^{--\top} + \tilde{\mathbf{A}}_u^{--+\top} + \tilde{\mathbf{A}}_u^{-++\top} \\ \tilde{\mathbf{A}}_t^{+-\top} + \tilde{\mathbf{A}}_u^{+--\top} + \tilde{\mathbf{A}}_u^{++-\top} & \tilde{\mathbf{A}}_t^{-+\top} + \tilde{\mathbf{A}}_u^{-+-\top} + \tilde{\mathbf{A}}_u^{---\top} \end{bmatrix} \begin{bmatrix} \mathbf{r}^{+} \\ \mathbf{r}^{-} \end{bmatrix}$$
$$+ c \begin{bmatrix} \mathbf{q} \\ \mathbf{0} \end{bmatrix} \Leftrightarrow \mathbf{h} = (1-c)\tilde{\mathbf{B}}^{\top}\mathbf{h} + c\mathbf{q}_x.$$

Therefore, the iteration in Algorithm 1 is written as follows:

$$\begin{aligned} \mathbf{h}^{(k)} &= (1-c)\tilde{\mathbf{B}}^{\top}\mathbf{h}^{(k-1)} + c\mathbf{q}_x \\ &= ((1-c)\tilde{\mathbf{B}}^{\top})^2\mathbf{h}^{(k-2)} + c((1-c)\tilde{\mathbf{B}}^{\top} + \mathbf{I})\mathbf{q}_x = \cdots \\ &= ((1-c)\tilde{\mathbf{B}}^{\top})^k\mathbf{h}^{(0)} + c\sum_{d=0}^{k-1}((1-c)\tilde{\mathbf{B}}^{\top})^d)\mathbf{q}_x. \end{aligned}$$

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 [32]. Therefore, $\lim_{k\to\infty}((1-c)\tilde{\mathbf{B}}^{\top})^k\mathbf{h}^{(0)} = 0$ [32] and $\lim_{k\to\infty}\mathbf{h}^{(k)}$ converges as:

$$\lim_{k\to\infty}\mathbf{h}^{(k)} = \mathbf{c}\sum_{d=0}^{\infty}((1-c)\tilde{\mathbf{B}}^{\top})^d\mathbf{q}_x = c(\mathbf{I} - (1-c)\tilde{\mathbf{B}}^{\top})^{-1}\mathbf{q}_x,$$

where $\sum_{d=0}^{\infty}((1-c)\tilde{\mathbf{B}}^{\top})^d)$ is a geometric series of the matrix $(1-c)\tilde{\mathbf{B}}^{\top}$, and it 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 1 [32]. Note that the inverse matrix is a non-negative matrix whose entries are positive or zero because the matrix is the sum of non-negative matrices (*i.e.,* $\sum_{d=0}^{\infty}((1-c)\tilde{\mathbf{B}}^{\top})^d)$. Therefore, each entry of $\mathbf{h}$ is non-negative (*i.e.,* $\mathbf{h}_y \geq 0$ for any node n_y). That is, the iterative algorithm of OBOE converges to a unique solution $\mathbf{h} = c(\mathbf{I} - (1-c)\tilde{\mathbf{B}}^{\top})^{-1}\mathbf{q}_x$. □

5 EVALUATION

In this section, we validate the effectiveness of our approach via extensive experiments. We designed our experiments, aiming at answering the following key evaluation questions (EQs):

- **EQ1**: Does our score propagation strategy help PNR?
- **EQ2**: Does OBOE provide a more accurate top-k PNR than competing methods?
- **EQ3**: Does OBOE provide a more accurate bottom-k PNR than competing methods?

Table 3: Dataset statistics

Datasets	Nodes	Edges	Positive Edges	Negative Edges	Density
Wikipedia	7,118	107,080	78.4%	21.6%	0.200%
Slashdot	82,140	549,202	77.4%	22.6%	0.008%
Epinions	131,828	841,372	85.3%	14.7%	0.005%

- **EQ4**: Does OBOE identify trolls more accurately than competing methods?
- **EQ5**: How sensitive is the accuracy of OBOE per different values of parameters?

5.1 Experimental Settings

Datasets. Following [13, 14], we used three real-world signed network datasets: Wikipedia, Slashdot, and Epinions. The datasets are all publicly available.[4] Table 3 shows the detailed statistics of the three datasets.

- **Wikipedia** is a voting network for electing managers in Wikipedia. This network contains users' supporting vote (*i.e.,* positive) and opposing vote (*i.e.,* negative) edges.
- **Slashdot** is a friendship network among users of a technology news site. This network contains friend (*i.e.,* positive) and enemy (*i.e.,* negative) edges between users.
- **Epinions** is a trust network among users of a product review site. This network contains trust (*i.e.,* positive) and distrust (*i.e.,* negative) edges between users.

Competing Methods. We compare OBOE with 5 competing methods. First, we employ three PNR methods for signed networks: two baselines, *i.e.,* ModifiedRWR (M-RWR, in short) [30], FriendTNS (F-TNS, in short) [33], and a recently-proposed and best performing PNR method, *i.e.,* SRWR [13, 14].[5] Second, we employ the recently-proposed two signed NE methods: BESIDE [5] and SLF [39]. For evaluation, we used the source codes provided by the authors [5, 13, 14, 30, 33, 39]. For parameter tuning, we found the best settings in competing methods and OBOE via grid search. Specifically, for parameters of competing methods, we used the best settings found via extensive grid search in the ranges suggested in their respective papers.[6] The best settings found in OBOE are as follows: $c = 0.5$ (Wikipedia, Slashdot, Epinions); $\beta_{+} = 0.9$ (Wikipedia, Slashdot, Epinions); $\beta_{-} = 0.9$ (Wikipedia and Slashdot) / 0.7 (Epinions).

Evaluation Tasks. Following [13, 14, 27], we evaluate the effectiveness of OBOE and competing methods via three types of tasks: (1) top-k PNR, (2) bottom k-PNR, and (3) troll identification. For top-k and bottom-k PNR tasks, following [13, 14], we randomly sample 5,000 seed nodes in a given network. Then, we consider 20% of positive (resp. negative) edges of the seed nodes as test edges and also use them as the ground truth of the top-k (resp. bottom-k) PNR task. We consider all edges except for the test edges in the network as training edges, and perform each method based on the training edges and then output the top-k and bottom-k PNRs per seed node. Finally, we evaluate how much each node's top-k and bottom-k PNRs obtained by each method contain the ground truth

[4] http://snap.stanford.edu/data
[5] As to SSRW, as its source code is not available and it is difficult to implement due to many missing details in [27], we could not evaluate the accuracy of SSRW.
[6] Refer to https://sites.google.com/view/oboe-sigir21/implementation for more details.

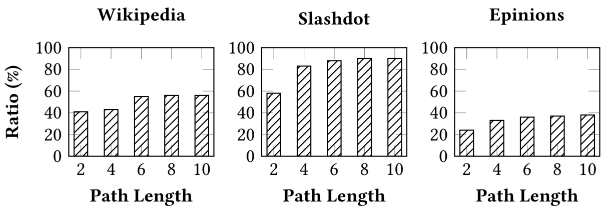

Figure 8: Ratios of untrustworthy score propagation.

of each task. Toward this end, we use the following two metric popularly employed in other PNR research [13, 14, 27]: F1 score and *normalized discounted cumulative gain* (NDCG) [11, 12]. For troll identification task, following [13, 14], we use 96 enemies of a user, called *No-More-Trolls*, in Slashdot as *trolls*. Specifically, the user is an administrative account created for the purpose of collecting a troll list [17]. Note that such a list exists only in the Slashdot. We evaluate how much the bottom-k PNRs obtained by each method contain the trolls by using F1 score and NDCG.

5.2 Results

Due to space limitation, we omit some experimental results in this paper. The details for all experiments are available at http://sites.google.com/view/oboe-sigir21.

EQ1: Effectiveness of Our Score Propagation Strategy. To verify the effectiveness of our key ideas (*i.e.,* sign verification and score propagation), we conduct experiments to answer the following two sub-questions:

- **EQ1-1 (Sign Verification)**: How much of the balance theory based score propagation does OBOE detect as the untrustworthy score propagation?
- **EQ1-2 (Score Propagation)**: Is it effective to exploit the statistics of a given network for correcting score propagation?

For EQ1-1, we examine the ratio of untrustworthy score propagation detected during the random-walk process of OBOE in each dataset. Figure 8 shows the results. The x-axis represents path lengths and the y-axis represents the ratio of untrustworthy score propagation per path length. Overall, we see that the ratio increases as the path length increases. We note, as shown in Figure 2, the prediction accuracy of SRSurfer decreases as the path length increases. Thus, we can say that OBOE detects untrustworthy score propagations more in the situation where the prediction accuracy of SRSurfer is decreasing. Moreover, we see that, on Wikipedia and Slashdot, our sign verification strategy considers a lot of score propagations (based on the balance theory) to be untrustworthy while, on Epinions, it considers relatively less score propagations to be untrustworthy. This is likely because of the fact that on Epinions, 90% of *all triangles* are of the "balance" (+,+) type.

Next, for EQ1-2, we validate the new score propagation of OBOE. Note that when OBOE encounters the untrustworthy score propagation, it propagates to both scores of two signs by leveraging the ratios of the prior and posterior signs of the triangles in a given signed network ((STEP 2) in Section 4.2). To verify this strategy, we first made two variants of OBOE, denoted as OBOE(Balance) and OBOE(FExtra), which propagate only to a score of a single sign predicted by the balance theory and FExtra, respectively. Here, OBOE(Balance) coincides with SRWR that does not use the balance attenuation factors. Also, we made two variants of OBOE, denoted as OBOE(Uniform) and OBOE(Reverse), which propagate to both scores at different ratios from our strategy. Specifically, OBOE(Uniform) uniformly propagates to scores of both signs as the ratios of 50:50, while OBOE(Reverse) propagates to those by using the *inverse* of ratios used in our strategy. For comparison, we denote a variant using our strategy as OBOE(Ours).

Table 4: Accuracies of the variants of OBOE according to different score propagation strategies (Slashdot)

Tasks	Top-k PNR		Bottom-k PNR	
Metrics	F1@10	NDCG@10	F1@10	NDCG@10
OBOE(Balance)	0.027	0.061	0.012	0.025
OBOE(FExtra)	0.012	0.018	0.016	0.033
OBOE(Uniform)	0.028	0.052	0.023	0.039
OBOE(Reverse)	0.024	0.039	0.011	0.021
OBOE(Ours)	**0.032**	**0.071**	**0.024**	**0.044**

Table 4 shows the accuracies of the variants of OBOE for top-k and bottom-k PNR tasks on Slashdot. First, among four variants except for OBOE(Ours), we found that OBOE(Uniform) showed the best accuracy except for NDCG@10 in the top-k task. The results represent that, for untrustworthy score propagation, propagating to both scores helps to improve the accuracy of PNR tasks than propagating to a single score. However, we observe that OBOE(Ours) consistently outperforms all other variants in all tasks. Specifically, OBOE(Ours) improves F1@10 of OBOE(Balance), OBOE(FExtra), OBOE(Uniform), and OBOE(Reverse) up to 15.7%/93.8%, 165.4%/39.7%, 14.7%/3.2%, and 33.4%/113.4% for top-k/bottom-k PNR tasks, respectively. This indicates that, when propagating to both scores, it is most effective to consider the inherent property of a given network.

EQ2 and EQ3: Accuracy Comparisons in Top-k/Bottom-k PNR Tasks. We conducted comparative experiments to show greater accuracy of OBOE than those of the competing methods in top-k and bottom-k tasks. Table 5 illustrates the results. The values in bold face and underlined indicate the best and 2nd best accuracies in each row, respectively.

We summarize the results shown in Table 5 as follows. First, surprisingly, the accuracies of signed NE methods are always quite low in both tasks. We know that signed NE methods have mainly been validated with the task that predicts the signs of the given positive and negative edges [5, 21, 39]. In this work, we found that the nodes' vectors obtained by them are effective in the task but not in measuring the degree of positivity or negativity for the edges in terms of personalization [22]. Second, we confirm that SRWR has better accuracy than the other PNR methods in most cases, which coincides with the results in [13, 14].

However, OBOE significantly outperforms all competing methods, except for the bottom-k task in NDCG on Epinions. More specifically, OBOE improves the F1@10 and NDCG@10 of SRWR up to 13.4% and 13.0%/95.9% and 57.8% for top-k/bottom-k PNR tasks, respectively. The results demonstrate that our sign verification and score propagation strategies are effective in improving the accuracy of PNR tasks. Moreover, we highlight that OBOE shows dramatic improvements in the bottom-k task. To more understand such results, we once again refer to the ratios of balanced and unbalanced triangles per type of prior signs, demonstrated in Section 3. Specifically, a type (*i.e.,* (+, +)) involving only a positive sign follows the rules of the balance theory considerably, whereas other types (*i.e.,* (+, −), (−, +), (−, −)) involving a negative sign often do not

Table 5: Accuracy of 5 competing methods and OBOE in top-k and bottom-k tasks

(a) Top-k Task

Metrics	Signed NE methods		PNR methods			OBOE	Gain
	BESIDE	SLF	M-RWR	F-TNS	SRWR		
(1) Wikipedia							
F1@10	0.008	0.014	0.043	0.011	0.044	**0.050**	**13.4%**
F1@20	0.008	0.015	0.044	0.012	0.043	**0.047**	**7.5%**
NDCG@10	0.010	0.017	0.054	0.016	0.055	**0.062**	**12.7%**
NDCG@20	0.012	0.022	0.068	0.021	0.067	**0.074**	**8.8%**
(2) Slashdot							
F1@10	0.009	0.019	0.029	0.007	0.028	**0.032**	**12.8%**
F1@20	0.008	0.013	0.022	0.005	0.022	**0.025**	**10.7%**
NDCG@10	0.018	0.034	0.063	0.019	0.062	**0.071**	**13.0%**
NDCG@20	0.022	0.038	0.069	0.020	0.068	**0.079**	**13.1%**
(3) Epinions							
F1@10	0.002	0.007	0.063	0.014	0.074	**0.075**	**0.8%**
F1@20	0.002	0.006	0.050	0.010	0.057	**0.060**	**3.4%**
NDCG@10	0.002	0.007	0.121	0.034	0.135	**0.138**	**2.1%**
NDCG@20	0.002	0.008	0.130	0.038	0.144	**0.149**	**3.4%**

(b) Bottom-k Task

Metrics	Signed NE methods		PNR methods			OBOE	Gain
	BESIDE	SLF	M-RWR	F-TNS	SRWR		
(1) Wikipedia							
F1@10	0.005	0.006	0.009	0.007	0.014	**0.020**	**38.6%**
F1@20	0.005	0.005	0.009	0.006	0.015	**0.019**	**30.4%**
NDCG@10	0.012	0.012	0.014	0.011	0.027	**0.034**	**27.3%**
NDCG@20	0.014	0.014	0.018	0.014	0.036	**0.045**	**23.9%**
(2) Slashdot							
F1@10	0.004	0.001	0.008	0.001	0.012	**0.024**	**95.9%**
F1@20	0.003	0.002	0.006	0.001	0.010	**0.017**	**63.7%**
NDCG@10	0.005	0.003	0.014	0.003	0.026	**0.044**	**68.7%**
NDCG@20	0.007	0.004	0.016	0.004	0.031	**0.049**	**57.8%**
(3) Epinions							
F1@10	0.003	0.001	0.005	0.002	0.026	**0.028**	**10.5%**
F1@20	0.005	0.001	0.005	0.002	0.022	**0.024**	**5.9%**
NDCG@10	0.004	0.001	0.009	0.003	**0.049**	0.047	-4.5%
NDCG@20	0.008	0.001	0.010	0.004	**0.054**	0.051	-5.9%

Table 6: Accuracy of 5 competing methods and OBOE in troll identification task (Slashdot)

Metrics	Signed NE methods		PNR methods			OBOE	Gain (%)
	BESIDE	SLF	M-RWR	F-TNS	SRWR		
F1@100	0.0009	0.0014	0.0003	0.0172	0.0481	**0.1632**	**239.0%**
F1@200	0.0011	0.0031	0.0004	0.0191	0.0553	**0.1729**	**212.5%**
NDCG@100	0.0009	0.0011	0.0003	0.0197	0.0520	**0.1814**	**249.2%**
NDCG@200	0.0013	0.0033	0.0005	0.0275	0.0757	**0.2471**	**226.4%**

follow the rules strictly. In this sense, SRSurfer-based PNR methods frequently perform inaccurate predictions for negative relationships, while OBOE effectively corrects such predictions, thereby improving the accuracy of bottom-k PNR considerably. Lastly, note the OBOE is relatively less effective on Epinions than other datasets. As mentioned in EQ1, on Epinions, 90% of all triangles are of the "balanced" $(+,+)$ type, thus leaving little room for OBOE to exploit new ideas (so only small gain (%) compared to the results of SRWR).

EQ4. Accuracy Comparison in Troll Identification Task. We verify that OBOE outperforms the competing methods in a troll identification task. Note that the bottom-k task aims to evaluate how much accurately each method finds the real negative edges of each seed node (*i.e.,* personalized ranking), while the troll identification task aims to evaluate how much accurately each method finds the trolls regardless of a specific seed node (*i.e.,* global ranking). Table 6 shows the results. We observe that OBOE consistently and remarkably outperforms all competing methods. More specifically, OBOE dramatically improves the F1@100 and NDCG @100 of SRWR, the best performer among the competing methods, up to 239.0% and 249.2%, respectively. The results indicate that correcting incorrect score propagation contributes to accurately measuring not only PNR of each node but also global ranking as well.

EQ5. Parameter Analysis for OBOE. We carefully analyze the sensitivity of accuracy according to different values for parameters, β_+ (*i.e.,* $C(sign_{feature} = +1)$) and β_- (*i.e.,* $C(sign_{\ feature} = -1)$). Figure 9 shows the accuracy changes of OBOE for Slashdot with varying β_+ and β_- in the bottom-k PNR task. The x-axis and y-axis represent the values of β_- and those of β_+, respectively, and the z-axis represents the accuarcies in terms of NDCG@10. In general, as β_+ and β_- increase, the accuracies of OBOE increase as well.

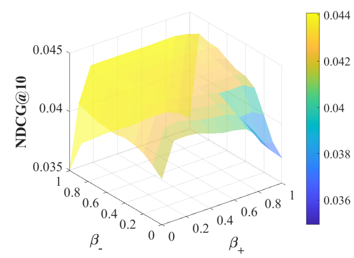

Figure 9: Accuracy changes according to β_+ and β_-.

Also, the result shows that the accuracy is the highest when β_+ and β_- are 0.9 and 0.9, respectively. This indicates that we should carefully use the FExtra-based predictions for the sign verification task, only when the predictions by FExtra are sufficiently confident.

6 CONCLUSIONS

In this paper, we investigated the limitation of SRSurfer-based PNR methods based on the balance theory: (1) edge relationships often do not strictly follow the rules of the balance theory and (2) the balance theory results in incorrect sign predictions of SRSurfer. To address this limitation, we proposed a novel random-walk based PNR approach, named as OBOE. OBOE is composed of (1) a *sign verification* based on nodes' topological features and a *score propagation* based on the statistics of a given signed network. In addition, we formulated the iteration of OBOE as a *matrix-vector multiplication form*. Furthermore, we analyzed that the iterative algorithm of OBOE *converges* to a unique solution. Through comprehensive experiments using three real-world datasets, we demonstrated that (1) our sign verification and score propagation strategies are *effective* and (2) OBOE consistently and significantly outperforms all competing methods in three types of tasks, *i.e.,* top-k PNR, bottom-k PNR, and troll identification, with three real-world datasets.

ACKNOWLEDGMENT

The work of Sang-Wook Kim was supported by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT1901-03. The work of Dongwon Lee was in part supported by NSF awards #1742702, #1820609, #1909702, #1915801, and #1934782.

REFERENCES

[1] Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. 2010. Fast Incremental and Personalized PageRank. *Proceedings VLDB Endow.* 4, 3 (2010), 173–184.

[2] Dorwin Cartwright and Frank Harary. 1956. Structural balance: a generalization of Heider's theory. *Psychological review* 63, 5 (1956), 277.

[3] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. 2018. CFGAN: A Generic Collaborative Filtering Framework based on Generative Adversarial Networks. In *Proceedings of the ACM International Conference on Information and Knowledge Management (ACM CIKM).* 137–146.

[4] Dong-Kyu Chae, Jihoo Kim, Duen Horng Chau, and Sang-Wook Kim. 2020. AR-CF: Augmenting Virtual Users and Items in Collaborative Filtering for Addressing Cold-Start Problems. In *Proceedings of the ACM International Conference on Research & Development in Information Retrieval (ACM SIGIR).* 1251–1260.

[5] Yiqi Chen, Tieyun Qian, Huan Liu, and Ke Sun. 2018. "Bridge": Enhanced Signed Directed Network Embedding. In *Proceedings of the ACM International Conference on Information and Knowledge Management (ACM CIKM).* 773–782.

[6] Kyung-Jae Cho, Yeon-Chang Lee, Kyungsik Han, Jaeho Choi, and Sang-Wook Kim. 2019. No, That's Not My Feedback: TV Show Recommendation Using Watchable Interval. In *Proceedings of the IEEE International Conference on Data Engineering (IEEE ICDE).* 316–327.

[7] Ernesto Estrada. 2019. Rethinking structural balance in signed social networks. *Discret. Appl. Math.* 268 (2019), 70–90.

[8] Taher H. Haveliwala. 2002. Topic-sensitive PageRank. In *Proceedings of The Web Conference (WWW)*, David Lassner, David De Roure, and Arun Iyengar (Eds.). 517–526.

[9] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial Personalized Ranking for Recommendation. In *Proceedings of the ACM International Conference on Research & Development in Information Retrieval (ACM SIGIR).* 355–364.

[10] Won-Seok Hwang, Juan Parc, Sang-Wook Kim, Jongwuk Lee, and Dongwon Lee. 2016. "Told you i didn't like it": Exploiting uninteresting items for effective collaborative filtering. In *Proceedings of the IEEE International Conference on Data Engineering (IEEE ICDE).* 349–360.

[11] Kalervo Järvelin and Jaana Kekäläinen. 2000. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the ACM International Conference on Research & Development in Information Retrieval (ACM SIGIR).* 41–48.

[12] Kalervo Järvelin and Jaana Kekäläinen. 2017. IR evaluation methods for retrieving highly relevant documents. *SIGIR Forum* 51, 2 (2017), 243–250.

[13] Jinhong Jung, Woojeong Jin, and U Kang. 2020. Random walk-based ranking in signed social networks: model and algorithms. *Knowl. Inf. Syst.* 62, 2 (2020), 571–610.

[14] Jinhong Jung, Woojeong Jin, Lee Sael, and U. Kang. 2016. Personalized Ranking in Signed Networks Using Signed Random Walk with Restart. In *Proceedings of the IEEE International Conference on Data Mining (IEEE ICDM).* 973–978.

[15] Jung Hyun Kim, Mao-Lin Li, K. Selçuk Candan, and Maria Luisa Sapino. 2017. Personalized PageRank in Uncertain Graphs with Mutually Exclusive Edges. In *Proceedings of the ACM International Conference on Research & Development in Information Retrieval (ACM SIGIR).* 525–534.

[16] Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *J. ACM* 46, 5 (1999), 604–632.

[17] Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. 2009. The slashdot zoo: mining a social network with negative edges. In *Proceedings of The Web Conference (WWW).* 741–750.

[18] Jérôme Kunegis, Julia Preusse, and Felix Schwagereit. 2013. What is the added value of negative links in online social networks?. In *Proceedings of The Web Conference (WWW).* 727–736.

[19] Jongwuk Lee, Dongwon Lee, Yeon-Chang Lee, Won-Seok Hwang, and Sang-Wook Kim. 2016. Improving the accuracy of top-N recommendation using a preference model. *Inf. Sci.* 348 (2016), 290–304.

[20] Yeon-Chang Lee, Sang-Wook Kim, and Dongwon Lee. 2018. gOCCF: Graph-Theoretic One-Class Collaborative Filtering Based on Uninteresting Items. In *Proceedings of the the AAAI Conference on Artificial Intelligence (AAAI).* 3448–3456.

[21] Yeon-Chang Lee, Nayoun Seo, Kyungsik Han, and Sang-Wook Kim. 2020. ASiNE: Adversarial Signed Network Embedding. In *Proceedings of the ACM International Conference on Research & Development in Information Retrieval (ACM SIGIR).* 609–618.

[22] Yeon-Chang Lee, Nayoun Seo, and Sang-Wook Kim. 2020. Are Negative Links Really Beneficial to Network Embedding?: In-Depth Analysis and Interesting Results. In *Proceedings of the ACM International Conference on Information and Knowledge Management (ACM CIKM).* 2113–2116.

[23] Ronny Lempel and Shlomo Moran. 2001. SALSA: the stochastic approach for link-structure analysis. *ACM Trans. Inf. Syst.* 19, 2 (2001), 131–160.

[24] Jure Leskovec, Daniel P. Huttenlocher, and Jon M. Kleinberg. 2010. Predicting positive and negative links in online social networks. In *Proceedings of The Web Conference (WWW).* 641–650.

[25] Jure Leskovec, Daniel P. Huttenlocher, and Jon M. Kleinberg. 2010. Signed networks in social media. In *Proceedings of the ACM International Conference on Human Factors in Computing Systems (ACM CHI).* 1361–1370.

[26] Xiaoming Li, Hui Fang, and Jie Zhang. 2017. Rethinking the Link Prediction Problem in Signed Social Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).* 4955–4956.

[27] Xiaoming Li, Hui Fang, and Jie Zhang. 2019. Supervised User Ranking in Signed Social Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).* 184–191.

[28] Tong Man, Huawei Shen, Shenghua Liu, Xiaolong Jin, and Xueqi Cheng. 2016. Predict Anchor Links across Social Networks via an Embedding Approach. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).* 1823–1829.

[29] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[30] Moshen Shahriari and Mahdi Jalili. 2014. Ranking Nodes in Signed Social Networks. *Soc. Netw. Anal. Min.* 4, 1 (2014), 172.

[31] Dongjin Song and David A. Meyer. 2015. Recommending Positive Links in Signed Social Networks by Optimizing a Generalized AUC. In *Proceedings of the the AAAI Conference on Artificial Intelligence (AAAI).* 290–296.

[32] Gilbert Strang. 2006. Linear Algebra and its Applications. *Thomson Brooks/Cole, Pacific Groove* (2006).

[33] Panagiotis Symeonidis and Eleftherios Tiakas. 2014. Transitive node similarity: predicting and recommending links in signed social networks. *World Wide Web* 17, 4 (2014), 743–776.

[34] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast Random Walk with Restart and Its Applications. In *Proceedings of the IEEE International Conference on Data Mining (IEEE ICDM).* 613–622.

[35] Hongning Wang, Xiaodong He, Ming-Wei Chang, Yang Song, Ryen W. White, and Wei Chu. 2013. Personalized ranking model adaptation for web search. In *Proceedings of the ACM International Conference on Research & Development in Information Retrieval (ACM SIGIR).* 323–332.

[36] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. 2017. Relational Deep Learning: A Deep Latent Variable Model for Link Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).* 2688–2694.

[37] Suhang Wang, Jiliang Tang, Charu C. Aggarwal, Yi Chang, and Huan Liu. 2017. Signed Network Embedding in Social Media. In *Proceedings of the SIAM International Conference on Data Mining (SDM).* 327–335.

[38] Zhaoming Wu, Charu C. Aggarwal, and Jimeng Sun. 2016. The Troll-Trust Model for Ranking in Signed Networks. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM).* 447–456.

[39] Pinghua Xu, Wenbin Hu, Jia Wu, and Bo Du. 2019. Link Prediction with Signed Latent Factors in Signed Social Networks. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (ACM KDD).* 1046–1054.

[40] Yaowei Yan, Yuchen Bian, Dongsheng Luo, Dongwon Lee, and Xiang Zhang. 2019. Constrained Local Graph Clustering by Colored Random Walk. In *Proceedings of The Web Conference (WWW).* 2137–2146.

[41] Shuhan Yuan, Xintao Wu, and Yang Xiang. 2017. SNE: Signed Network Embedding. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, Vol. 10235. 183–195.

[42] Tianchen Zhu, Zhaohui Peng, Xinghua Wang, and Xiaoguang Hong. 2017. Measuring the Similarity of Nodes in Signed Social Networks with Positive and Negative Links. In *Proceedings of the APWeb-WAIM Joint Conference on Web and Big Data (APWeb-WAIM)*, Vol. 10366. 399–407.