IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 23, 2019, accepted August 12, 2019, date of publication August 26, 2019, date of current version December 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937508

Integrating Sign Prediction With Behavior Prediction for Signed Heterogeneous Information Networks

DONG LI, DERONG SHEN, YUE KOU, AND TIEZHENG NIE
School of Computer Science and Engineering, Northeastern University, Shenyang 110004, China

Corresponding author: Derong Shen (shenderong@cse.neu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1003404, in part by the National Natural Science Foundation of China under Grant 61672142 and Grant U1435216, and in part by the Program of China Scholarships Council under Grant 201806085016.

ABSTRACT People often use online social networks not only to express attitudes towards others, but also to make decisions, which forms signed heterogeneous information networks. Both sign prediction and behavior prediction can provide useful information for networks analysis, each of which has been a hot topic. However, existing methods for sign prediction mainly rely on the features from labeled links but ignore users' behavior and the features from unlabeled links, which often leads to dumb results. Similarly, inferring users' behavior without considering links' signs is dull as well. In order to solve this issue, in this paper, we present a novel model called SPBP to integrate Sign Prediction with Behavior Prediction in the context of signed heterogeneous information networks. It simultaneously captures users' social links (including both labeled links and unlabeled links) and users' behavior to improve the accuracy of prediction. First, due to the lack of labeled links in main stream social networks, we propose correlation estimation methods to estimate social correlation and behavioral correlation between users respectively. Then we encode structural balance-based features and status-based features according to social psychology theories. With the extracted features, we propose a sign prediction algorithm based on transfer learning to use knowledge extracted from related source networks to train the target network, which can effectively make up for the incompleteness of target samples. Finally, we propose a behavior prediction algorithm based on the predicted signs of links. Extensive experiments conducted on real-world signed heterogeneous information networks, Epinions, Slashdot and Wiki-RfA, demonstrate that SPBP can effectively solve both the sign prediction problem and the behavior problem.

INDEX TERMS Behavior prediction, sign prediction, signed heterogeneous information networks, social psychology, transfer learning.

I. INTRODUCTION

With the popularity of social networks, more and more social interaction exists in the networks. On one hand, people often use online social networks to express attitudes towards others, which forms signed social networks. In signed social networks, there are positive social links and negative social links among users. More specifically, users can use positive social links to express the relationships of friendship, support, or approval. They can also use negative social links to indicate disagreement or distrust of the opinions of others. For example, Epinions, Slashdot and Wiki-RfA are three well-known social networks, in which users can approve or denounce reviews and articles [1], [2], declare others to be either friends or foes [3], [4], and vote for or against others [5]. The recent availability of signed social networks has promoted the research on signed network analysis [6]. On the other hand, people can also use online social networks to make decisions. For example, users can use social networks to rate movies, download songs, purchase commodities, publish comments, etc. Users, social links between them and their behavioral decisions form signed heterogeneous information networks. Compared with homogeneous information networks or unsigned heterogeneous

The associate editor coordinating the review of this article and approving it for publication was Jun Li.

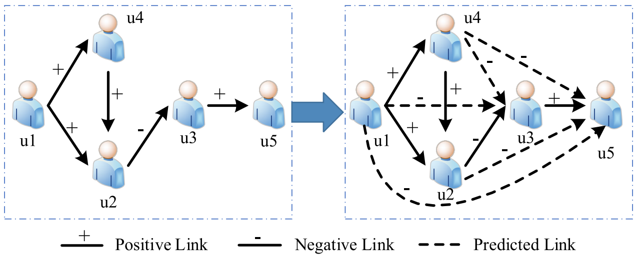

FIGURE 1. **An illustration of sign prediction in an ideal signed social network.**

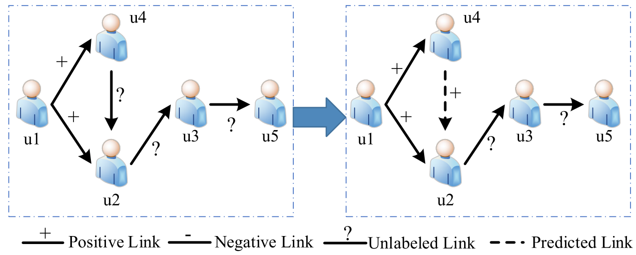

FIGURE 2. **An illustration of sign prediction in an incomplete signed social network.**

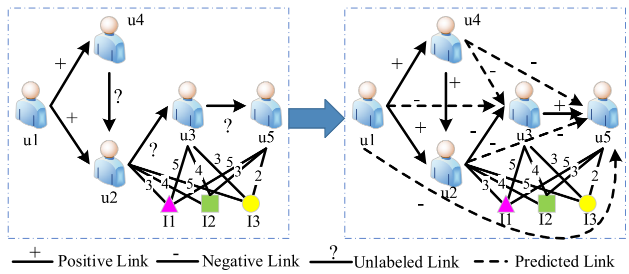

FIGURE 3. **An illustration of sign prediction by considering users' behavior.**

information networks, signed heterogeneous information networks can provide more abundant evidence for sign prediction and behavior prediction.

Both sign prediction and behavior prediction can provide useful information for networks analysis, each of which has been a hot topic. As a fundamental problem in signed social networks, sign prediction is to predict the signs of unlabeled social ties according to existing evidence. It is helpful to study the edges between entities as being signed either positively or negatively. The technique of sign prediction can be applied in the area of friendship recommendation, public opinion analysis, opinion polls, etc. In addition, behavior prediction is also a hot topic to predict the items which users are interested in. The technique of behavior prediction can be applied in the field of personalized advertising, personalized recommendation, profile construction, etc.

However, existing methods for sign prediction mainly rely on the features from labeled links but ignore users' behavior and the features from unlabeled links, which often leads to dumb results. Similarly, inferring users' behavior without considering links' signs is dull as well.

A. MOTIVATING SCENARIOS

Let us consider the following motivating scenarios.

Scenario 1: First we consider a scenario in an ideal signed social network. As shown in Figure 1, there are five users (u1∼u5) who are connected by a social network. The label of each social tie represents positive semantic (denoted as '+') or negative semantic (denoted as '-'). The ideal signed social network means that most edges in the network have been labeled. So we can acquire abundant evidence to predict new signed links (denoted as dashed). Here we can use social psychology theories, such as balance theory and status theory [7], to infer some new links. For example, we can infer a link labeled as "negative" from u1 to u3. It is consistent with the intuition that "the enemy of my friend is my enemy". Besides the tie from u1 to u3, lots of new links can be inferred as positive or negative. Therefore, by using the traditional techniques for sign prediction, we can acquire a better result for the ideal signed social networks.

Scenario 2: Now we consider the case of sign prediction in incomplete (i.e. non-ideal) signed social networks (shown in Figure 2). Here an incomplete network means that only a small fraction (e.g. lower than 10%) of the signed links is given. Most links are unlabeled in an incomplete network. In fact, there are usually less explicit signed links in mainstream online social networks. We focus on the question whether the traditional sign prediction approaches are still effective to infer signed ties in the absence of many labels, especially in the case that the vast majority of the labels are missed. However, most existing methods for sign prediction tend to rely on the features of labeled ties. Due to scarce labeled links in incomplete networks, there is inadequate evidence for them to make prediction. Similar to Scenario 1, we also use social psychology theories to predict links. As shown in Figure 2, only one link labeled as "positive" from u4 to u2 can be predicted.

Scenario 3: Now let us see what will happen to sign prediction if we consider users' behavior. Based on Figure 2, suppose the heterogeneous information network after adding users' behavior (items and ratings) is shown in Figure 3. It includes not only the labeled (or unlabeled) social ties between users, but also the ratings which users express on some items. For example, u2 expresses ratings (3, 4, 5) on items I1, I2 and I3 respectively. In [8], statistics have indicated that the majority (almost 90%) of positive links indeed show positive behavioral correlations. Also almost 40% of the negative links, the signs of relations are consistent with the signs of behavioral correlations. Therefore, we can use behavioral correlations to infer the signs of links between users. Due to negative behavioral correlation between u2 and u3, there is a higher probability that u2 distrusts u3 as well. On the contrary, due to positive behavioral correlation between u3 and u5, it is more likely to lead to a positive link from u3 to u5. Further, based on the social psychology theories, we can acquire the same result as Scenario 1.

Scenario 4: In this scenario, let us see what links' signs do to users' behavior prediction. Based on Figure 3, suppose we want to infer the behavior of u4. If we only consider the acquaintance relationship between users rather than links' signs (as is the case with most recommendation technologies in social networks), u2, u3 and u5 will be used to infer u4's behavior due to their adjacency to u4. But in fact, only u2 is the user whom u4 trusts and has the similar behavior with u4. While u3 and u5 will not promote the behavior prediction of u4, even though they will have a negative effect on it. Therefore, the behavior of one user is only similar with the users whom he trusts rather than whom he knows.

From the above scenarios, we can observe signed heterogeneous information networks in the real world have the following properties: Firstly, there are much more unlabeled links than labeled links in mainstream online social networks. If only the features from labeled ties are considered during prediction, the evidence for link prediction will be inadequate. Secondly, the training data available to us may be in short supply. Due to the shortage of the training data, the result of prediction generated by machine learning models (e.g. SVM or logistic regression) might be inaccurate and incomplete. Thirdly, in spite of the absence of many labels, users' behavior can provide more abundant evidence for sign prediction. At the same time, the predicted signed links are significantly more informative for the purpose of behavior prediction.

Typically, existing methods for sign prediction mainly rely on the features from labeled links but ignore the characteristics of unlabeled ties and users' behavior, which often leads to dumb results. Some methods try to take advantage of social psychology theories into the process of prediction, but they mainly rely on labeled ties to check whether balance theory or status theory is satisfied. That is to say, the accuracy of prediction is higher for the case of complete signed networks. However, the problem of missing labels is common in real networks. For such incomplete networks, if only relying on the features from labeled links, we are unable to acquire adequate evidence for prediction. On the other hand, inferring users' behavior without considering links' signs is dull as well. Although some recommendation technologies in social networks have been proposed, they only consider the acquaintance relationship between users rather than links' signs. While the sign is an important factor in calculating user similarity.

In our previous work [9], we have proposed a sign prediction model called UTLP for incomplete signed networks. However, UTLP is designed mainly for signed homogeneous information networks in which there is only one type of nodes (i.e. users themselves). It does not make effective use of users' behavioral information. This paper follows our previous work and tries to integrate sign prediction with behavior prediction in the context of signed heterogeneous information networks. We simultaneously captures users' social links (including both labeled links and unlabeled links) and users' behavior to improve the accuracy of prediction. In addition, in order to solve the problem of sample data shortage, we adopt the transfer learning framework to train the model and predict the hidden signs of the links.

B. CHALLENGES

However, both sign prediction and behavior prediction in the real signed heterogeneous information networks are highly challenging problems. The major challenges are as follows:

(1) As for sign prediction, how to introduce useful features in order to predict the signs of links, and how to select a suitable learning framework to train the model? Firstly, we should consider users' behavior as one type of feature. Most methods rely heavily on manually designed features and cannot work well in real-world scenario. So our first challenge lies in how to automatically learn features of users. Secondly, current social psychology theories-based features extraction methods only utilize the features from labeled ties. We should extend the methods via taking into account more features. Thirdly, existing sign prediction algorithms often use supervised methods which consider the link prediction problem as a classification problem. They use the existence of links as labels. There are so many classification models. We should select the most suitable one for incomplete signed networks.

(2) As for behavior prediction, how to effectively use links' signs to improve its performance? Current behavior prediction methods only use the acquaintance relationship between users rather than links' signs to infer users' behavior, which limits their power in the task of practical behavior prediction. We should apply more explicit social features of users to the process of behavior prediction and distinguish between positive links and negative links when calculating user similarity.

C. CONTRIBUTIONS AND ORGANIZATION

In this paper, we present a novel model called SPBP to integrate Sign Prediction with Behavior Prediction in the context of signed heterogeneous information networks. It simultaneously captures users' social links (including both labeled links and unlabeled links) and users' behavior to improve the accuracy of prediction. More specifically, we make the following contributions:

(1) We propose correlation estimation methods to estimate social correlation and behavioral correlation between users respectively. First, we adopt deep-learning based embedding technique to extract users' representations. Then we use these representations and users' behavior to estimate social correlation and behavioral correlation respectively. Also, by considering both labeled links and unlabeled links, we encode structural balance-based features and status-based features according to social psychology theories.

(2) With the extracted features, we propose a sign prediction algorithm based on transfer learning. We consider the sign prediction problem as a classification problem and employ the transfer learning framework to improve the performance of learning. We use knowledge extracted from

related source networks to train the target network, which can effectively make up for the incompleteness of target samples.

(3) We propose a behavior prediction algorithm based on the predicted signs of links. During the process of behavior prediction, we take into account the social features of users and distinguish between positive links and negative links which can provide more informative evidence for behavior prediction.

(4) We conduct extensive experiments on real-world signed heterogeneous information networks, Epinions, Slashdot and Wiki-RfA. Experiments demonstrate the effectiveness and the efficiency of our proposed methods compared with traditional methods.

The rest of this paper is organized as follows. Section II reviews the related work. Section III formulates the main problem and gives an overview of our model. Section IV proposes correlation estimation methods. Section V and Section VI propose our sign prediction algorithm and behavior prediction algorithm respectively. Section VII shows the experimental result and Section VIII concludes.

II. RELATED WORK

In this section, we briefly review existing research on sign prediction and behavior prediction respectively.

A. SIGN PREDICTION

Various approaches for sign prediction have been studied over the years, which mainly include nodes' features-based sign prediction, social psychology theories-based sign prediction and latent features-based sign prediction.

Nodes' features-based sign prediction methods consider nodes' features, including in-degree or out-degree, the number of common neighbors with another node, the influence of nodes and so on, as the evidence for sign prediction. Intuitively, the more similar two nodes are, the more likely there exists a positive link between them. Otherwise, the more likely there exists a negative link between them [10]. Some similarity measures such as CN [11], [12], RA [13] and AA [14] have been used for link prediction in signed networks. In addition, some work (e.g. [15]–[17]) uses deep learning frameworks to extract users' latent representations and to predict the sign of unobserved sentiment links. However, they tend to rely on training datasets with clear labels, while there are few social networks that have been explicitly labeled.

Social psychology theories-based sign prediction methods use structural balance theory [7], [18] or status theory [19] to infer the sign of each target link. Structural balance theory is based on the intuition that "the friend of my friend is my friend" and "the enemy of my enemy is my friend". For example, in [20] a supervised machine-learning algorithm is proposed based on structural balance theory. It tries to keep the balance of the triangles in signed networks. In other words, it assigns a sign to an edge by minimizing the number of unbalanced triangles involving the edge. In [21] sixteen triad relations among users are considered as features and some machine learning methods are adopted to learn and classify. Besides structural balance theory, status theory is another social psychology theory for sign prediction with the aim at determining an implicit ordering of the nodes. It is based on the principle that a positive link indicates that source node has a higher status than target node; while a negative link indicates that source node has a lower status than target node. The goal is to find an ordering of the nodes that maximizes the number of edges pointing from a node earlier in the ordering to one that is later in the ordering [19]. In addition, some work tries to combine structural balance theory and status theory to infer the signs. For example, in [9] by combining structural balance-based features and status-based features, a model is designed for sign prediction in incomplete signed networks.

Latent features-based sign prediction methods are based on the principle that the user adjacency matrix can be factorized into a user matrix and an item matrix by using the technique of matrix factorization (MF). The two matrixes consist of the user and item latent factor vectors respectively, which can be used to make a further prediction. For example, in [4] a classification method for link prediction is proposed from the viewpoint of spectrum analysis. A number of different matrixes (such as adjacency matrix and Laplacian matrix) are used as the objects for factorization. In [22] a low-rank matrix factorization approach with generalized loss functions is proposed as a practical method for sign inference. In [8], a latent factor model (called BRI) is proposed to infer social interactions and to infer users' behavior simultaneously. It relies heavily on behavioral correlation to infer the signs of the social ties.

B. BEHAVIOR PREDICTION

Behavior prediction aims at predicting the rating for user on a non-rated item or generally recommending some items for the given user based on the ratings that already exist. The related work includes CF (Collaborative filtering)-based behavior prediction, matrix factorization-based behavior prediction and social tie-based behavior prediction.

CF-based behavior prediction is a technique that automatically predicts the interest of an active user by collecting rating information from other similar users or items. The underlying assumption is that the active user will prefer those items which the similar users prefer. It mainly includes memory-based collaborative filtering [23]–[26] and model-based collaborative filtering [27]–[30]. But it is most effective only when users have expressed enough ratings to have common ratings with other users. When the user-item rating matrix is sparse, CF-based methods fail to find similar users.

Matrix factorization-based behavior prediction is a technique to learn latent features for users and items from the observed ratings. It can work well with sparsity of the user-item rating matrix. Several matrix factorization methods [31]–[33] have been proposed, which focus on fitting the user-item rating matrix using low-rank approximations and using it to make further predictions. In addition, some work also incorporates the content information into the matrix factorization process. For example, in [34] an approach combining the merits of traditional matrix factorization and probabilistic topic modeling is proposed. In [35] a Bayesian generative model is proposed, which considers both rating and content for recommendation. However, most of them are based on the assumption that users are independent and identically distributed, which ignore users' social activities.

In order to take advantage of the social activities between users effectively, some social tie-based behavior prediction methods are proposed [36]–[40]. For example, in [36], a trust-aware collaborative filtering method for recommender systems is proposed. The reputation of users is computed by propagating trust. In [37], a trust-based recommender system for the semantic web is proposed which uses the web of trust to generate the recommendations. In [38], a factor analysis approach based on probabilistic matrix factorization is proposed to solve the data sparsity and poor prediction accuracy problems by employing both users' social network information and rating records. In [39], a context-dependent matrix factorization model is proposed, which considers users' social ties as a kind of context. In [40] a model-based approach for recommendation in social networks is proposed, which employs matrix factorization techniques and uses the mechanism of trust propagation.

C. DIFFERENCES FROM EXISTING WORK

The differences between our work and existing work are as follows: Firstly, existing methods for sign prediction mainly rely on the features from social links but ignore users' behavior. Although some methods consider the interplay between users' behavior and social interaction, it focuses heavily on behavioral correlation rather than social correlation. In our model, we consider users' behavior as one type of feature and try to integrate sign prediction with behavior prediction in the context of signed heterogeneous information networks. Also we adopt deep-learning based embedding technique to extract users' representations and estimate both social correlation and behavioral correlation. Secondly, most existing work focuses on extracting features from labeled links only. However, there are usually less explicit signed links in mainstream online social networks, especially in real world sparse social networks. Different from the existing work, we utilize the features from both labeled links and unlabeled links to extract social psychology theories-based features, which can supply abundant evidence for prediction. Finally, although existing methods of social tie-based behavior prediction consider the social activities between users, they only use the acquaintance relationship between users rather than links' signs to infer users' behavior. We not only take into account the social features of users, but also distinguish between positive links and negative links which can provide more informative evidence for behavior prediction.

As is mentioned above, this paper follows our previous study in [9]. We make the following extensions in this paper. First, our previous work focuses on solving the problem of sign prediction in signed homogeneous information networks. In this paper, we are dealing with more complex networks, i.e. signed heterogeneous information networks. The properties of signed heterogeneous information networks can provide us with more features for prediction. Besides the features defined in [9], in this paper we extract users' deep representations to quantify both social correlation and behavioral correlation. Second, instead of solving the problem of sign prediction alone as in [9], we try to combine sign prediction with behavior prediction in this paper. The aim is to improve their accuracy simultaneously based on their mutual promotion. Correspondingly we propose a sign prediction algorithm and a behavior prediction algorithm. Third, compared with [9], we extend the experiment by evaluating effectiveness and performance of sign prediction, impact of parameters, and performance of behavior prediction respectively.

III. MODEL OVERVIEW

In this section, we first give some definitions and formulate the main problem. Then we give an overview of our model.

A. PROBLEM STATEMENT

First, we introduce a few preliminaries before going into details. Let $U = \{u_1, u_2, \ldots, u_m\}$ and $I = \{I_1, I_2, \ldots, I_n\}$ be the set of m users and the set of n items respectively in a signed heterogeneous information network. Now we define the signed heterogeneous information network.

Definition 1 (Signed Heterogeneous Information Network): A signed heterogeneous information network is a graph $G = (V, E)$, where V is the vertex set (i.e. $U \cup I$), $E \subseteq V \times V$ denotes the set of edges. There are two kinds of nodes in G, i. e. users and items. Also there are two kinds of edges in G: One is between a pair of users with the social tie labeled as +1, -1 or missing. The other is between a user and an item representing a decision mapping, e.g. a user makes a decision regarding an item with a rating.

For better illustration, we split the above signed heterogeneous information network into two single-type networks: signed network and behavioral network.

Definition 2 (Signed Network): A signed network is a graph $G^S = (V^S, E^S, A^S)$, where V^S is the vertex set (i.e. the user set U), $E^S \in R^{m\times m}$ is the social adjacency matrix, and $A^S \in R^{m\times m}$ is the signed adjacency matrix associated with G^S. The entries of E^S and A^S are as follows:

$$E^S_{ij} = \begin{cases} 1, & \text{if the edge } (i,j) \text{ exists in } G^S \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

$$A^S_{ij} = \begin{cases} 1, & \text{if the edge } (i,j) \text{ is labeled as positive} \\ -1, & \text{if the edge } (i,j) \text{ is labeled as negative} \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

Here E^S and A^S play different roles. E^S is used to denote the acquaintance relationship between users. If two users know each other, the corresponding value in E^S is 1, whether the link is labeled or not. Otherwise, its value is 0. As long as there's an edge between i and j, whatever its sign is or whether

TABLE 1. Notations.

Notation	Meaning
U, I	User set and item set
u_i or i	A user in U
G^S=(V^S, E^S, A^S)	Signed network
G^B=(V^B, E^B)	Behavioral network
f^S	Sign prediction function
f^B	Behavior prediction function
$\widehat{x_i}$	Representation of u_i
C_{ij}^S	Social correlation between u_i and u_j
C_{ij}^B	Behavioral correlation between u_i and u_j
sign(i, j)	Link's sign between u_i and u_j
r_{ij}	The final rating made by u_i regarding I_j

it is labeled, then $E_{ij}^S = 1$. In A^S, the edges can be classified into three types: positive, negative and others. It is used to denote the signed user-user relationships where $A_{ij}^S = 1$ if the sign of edge (i, j) is positive, $A_{ij}^S = -1$ if the sign of edge (i, j) is negative, and $A_{ij}^S = 0$ otherwise. Here $A_{ij}^S = 0$ means: there is not an edge between i and j, or we do not currently know its sign. As for an incomplete signed network, there are very few (lower than 10%) entries in A^S with values of 1 or -1.

Definition 3 (Behavioral Network): A behavioral network is a graph $G^B = (V^B, E^B)$, where V^B is the vertex set (i.e. the union of the user set U and the item set I), $E^B \in R^{m\times n}$ is the rating matrix and each entry in it represents a rating made by a user regarding an item.

Now let us define our two tasks: sign prediction and behavior prediction.

Definition 4 (Sign Prediction): Given a signed network G^S and a behavior network G^B, a signed social tie is a mapping f^S (shown in (3)) which characterizes the overall impression (positive or negative) of user i to user j. The aim of sign prediction is to develop a predictor f^S to label the edges of the graph G^S with signed labels $+1$ or -1.

$$f^S : \{E_{ij}^S \rightarrow \{+1, -1\}, \tag{3}$$

Definition 5 (Behavior Prediction): Given a signed network G^S and a behavior network G^B, the aim of behavior prediction is to develop a mapping f^B (shown in (4)), which predicts the ratings made by users regarding items, which are missing in the rating matrix.

$$f^B : U \times I \rightarrow r, \tag{4}$$

Frequently used notations in this paper are summarized in Table 1.

B. OVERVIEW OF OUR MODEL

As for the properties of signed heterogeneous information networks, we present a novel model called SPBP to integrate Sign Prediction with Behavior Prediction in the context of signed heterogeneous information networks. The basic idea of SPBP model is shown as Figure 4. It includes two tasks: sign prediction and behavior prediction.

1) SIGN PREDICTION

As for sign prediction, the input is training data set, a signed network and a behavior network. The output is parameters estimated and the sign prediction result. Formally, given a signed network G^S, a test edge set $E_{test}^S \subseteq E^S$ and a behavior network G^B, we want to predict the signs of E_{test}^S, using only the edges in $E^S - E_{test}^S$. It includes two major phases (training phase and predicting phase).

a: TRAINING PHASE

The training data consists of pairs $(E^S - E_{test}^S, +1/\text{-}1)$ where $E^S - E_{test}^S$ ranges over edges whose signs are given to us. Its aim is to estimate the parameters based on the training data. Firstly, some features are extracted from the training data. The features are mainly divided into two classes. The first class is based on social correlation and behavioral correlation between users (see Section IV). On one hand, we adopt deep-learning based embedding technique to extract users' representations and further to estimate social correlation between users. On the other hand, we use the rating matrix in the behavior network to estimate behavioral correlation between users. The second class is based on social psychology theories, which includes the features of the nodes (e.g. their degrees or the number of common neighbors), balance-based features and status-based features (see Section V.A). Both the labeled ties and the unlabeled ties are considered to extract the features. Secondly, we use a classification model to optimize the predictive function f^S that maps these features into a label (see Section V.B). Here we borrow the idea of transfer learning to train our model. Finally, the parameters are learned, which can be used in the predictive function f^S during the phase of prediction.

b: PREDICTING PHASE

We take the signed network G^S as the input. During the predicting phase, the aim is to label the edges in E_{test}^S as $+1$ or -1. Similar to the training phase, first of all, features are extracted from G^S. Then, according to the parameters and the predictive function f^S learned during the training phase, each edge in E_{test}^S is labeled as $+1$ or -1.

2) BEHAVIOR PREDICTION

As for behavior prediction, the input is a behavior network and a signed network. The output is the behavior prediction result. It includes two parts: matrix factorization and decision making (see Section VI).

Firstly, the rating matrix is factorized into latent user and item feature matrices, with column vectors representing user-specific and item-specific latent feature vectors respectively. Here, based on users' representations extracted via G^S embedding, the information of users' social ties is introduced into the process of matrix factorization. By fusing users' social ties with the user-item rating matrix, the result of behavior prediction will be more accurate. Secondly, according to user-specific latent feature vectors, item-specific latent

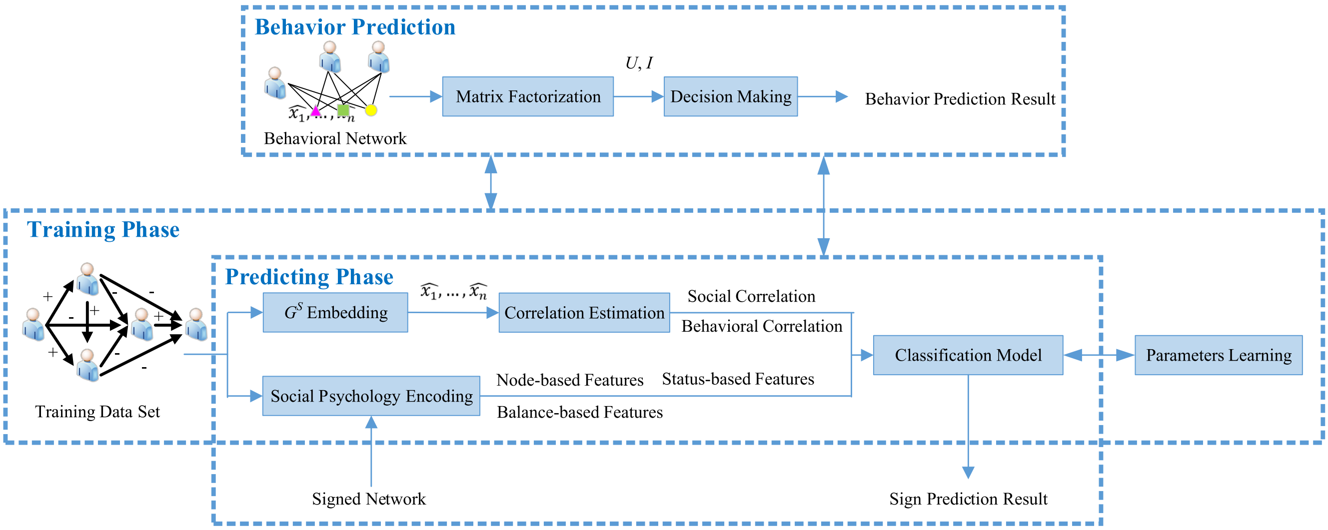

FIGURE 4. Overview of SPBP model.

feature vectors and the predicted links' signs between users, the ratings (i.e. decisions) made by users regarding items are predicted. Also the new predicted behaviors will be applied to the process of sign prediction again.

The advantages of our model are mainly reflected in two aspects. On one hand, it makes full use of the interplay between users' behavior and social interaction to make the prediction result more accurate. In our model, we consider users' behavior as one type of feature (i.e. behavioral correlation), and with the new prediction of behaviors, the feature will be updated as well. Also we fuse users' social ties with the user-item rating matrix and use the predicted links' signs between users to predict the behavior of users. On the other hand, during the process of sign prediction, we utilize the features of both labeled ties and unlabeled ties of the networks to extract social psychology theories-based features, which can supply more abundant evidence for prediction.

IV. CORRELATION ESTIMATION

In this section, we adopt deep-learning based embedding technique to extract users' representations and estimate both social correlation and behavioral correlation.

A. SOCIAL CORRELATION ESTIMATION

Given a signed network G^S, for each user $i \in V^S$, we define its adjacency vector $x_i = (E^S_{i1}, \ldots, E^S_{im})$. Here each element in x_i means the acquaintance relationship between i and each user in G^S. If we directly use x_i to represent a user, it will be impractical due to the long size of the vector and the sparsity of the network. In this paper, we adopt deep-learning based embedding technique to extract users' representations. More specifically, we use autoencoder model proposed in [41] to learn the representations of users, which is a kind of neural network designed for dimensionality reduction. It contains multiple nonlinear layers for mapping the input data to representation space and reconstructing original input from representation respectively.

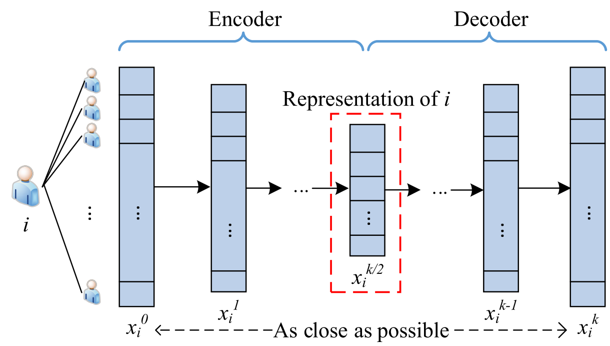

FIGURE 5. The basic idea of autoencoder-based user representation.

The basic idea of autoencoder-based user representation is illustrated in Figure 5. Suppose we want to acquire the representation of user i. There are k layers in the autoencoder and we denote the vector generated at the layer z as x_i^z. The input is $x_i^0 = x_i$ and the output is $x_i^k = x_i'$. The autoencoder is an unsupervised model which is composed of two parts, i.e. encoder and decoder. Firstly, it starts out by compressing the data into a lower-dimensional representation, i.e. the process of encoder. We use the encoder to generate a more meaningful representation of the inputted user, i.e. $x_i^{k/2}$. Secondly, the autoencoder converts that lower-dimensional representation back to a reconstruction of the original input, i.e. the process of decoder. We use the decoder to convert $x_i^{k/2}$ back to x_i (or as close to x_i as possible).

The goal of the autoencoder is to minimize the reconstruction loss between input and output representations. Similar to [17], we define the loss function as (5). Here $\odot$ means the Hadamard product, and $L_i = (l_{i1}, \ldots, l_{im})$ is the reconstruction weight vector (defined as (6)). We use the loss function to impose more penalty to the reconstruction error of the non-zero elements than that of zero elements in each adjacency vector x_i. That is because a non-zero element carries more explicit adjacency information than a zero element. Finally, for each user x_i, we use $\hat{x}_i = x_i^{k/2}$ to represent him.

$$L = \sum_{i \in V^S} ||(x_i - x_i') \odot L_i||_2^2, \tag{5}$$

$$l_{ij} = \begin{cases} \beta > 1, & \text{if } E_{ij}^S = 1 \\ 1, & \text{otherwise}, \end{cases} \tag{6}$$

Based on users' representations, we can estimate social correlation between them. In this paper, we use Pearson correlation score between the representations of two users (i and j) to measure their social correlation (shown in (7)). If they are positively correlated in their representations, the more likely they are to be linked positively. Otherwise, the more likely there is a negative link between them.

$$C_{ij}^S = \frac{cov(\hat{x}_i, \hat{x}_j)}{\sigma_{\hat{x}_i}\sigma_{\hat{x}_j}}, \tag{7}$$

B. BEHAVIOR CORRELATION ESTIMATION

Besides social correlation between users, we also consider behavioral correlation between them as one type of feature.

Firstly we need to represent each user's behavior as a vector. For user i, we use $y_i = (E_{i1}^B, \ldots, E_{im}^B)$ to denote his behavior vector, where E_{ij}^B means the rating made by user i regarding the item j. For example, suppose there are five items in G^B. User i and user j express ratings (3, 4, 5, null, 1) and (4, null, 5, null, 2) on these items respectively. Then we use y_i = (3, 4, 5, null, 1) and y_j = (4, null, 5, null, 2) to denote their behavior vectors respectively.

Secondly, based on the behavior vectors, we can estimate behavioral correlation between users. Similar to social correlation, we also use Pearson correlation score between the behavior vectors of two users (i and j) to measure their behavioral correlation (shown in (8)). When calculating behavioral correlation, we only consider the non-null elements in behavior vectors. As for the above example, both user i and user j rate items I_1, I_3 and I_5, so their behavior vectors only containing these items are (3, 5, 1) and (4, 5, 2) respectively, which will be compared in (8). Similarly, as for two users, if they are positively correlated in their behavior vectors, the more likely they are to be linked positively. Otherwise, the more likely there is a negative link between them.

$$C_{ij}^B = \frac{cov(\hat{y}_i, \hat{y}_j)}{\sigma_{\hat{y}_i}\sigma_{\hat{y}_j}}, \tag{8}$$

Therefore, we consider both social correlation and behavioral correlation by combining them as one of the features for sign prediction (shown in (9)). Here $\in [0, 1]$ is the weight with the meaning how the overall correlation depends on social correlation.

$$sign(i, j) = sign(\lambda C_{ij}^S + (1 - \lambda) C_{ij}^B), \tag{9}$$

V. SIGN PREDICTION

In this section, we propose a sign prediction algorithm. First we encode node-based features, structural balance-based features and status-based features according to social psychology theories by considering both labeled links and unlabeled links. Then our sign prediction algorithm based on transfer learning is proposed.

A. ENCODING SOCIAL PSYCHOLOGY

Besides the correlation estimated in Section IV, our SPBP model also encodes the features of both labeled ties and unlabeled ties based on social psychology theories. Here three types of features are captured, including node-based features, structural balance-based features and status-based features, which will be as the input of the classification model.

1) NODE-BASED FEATURES

Nodes' features are used to describe the aggregate local relations of the nodes to the rest of the world. Here we consider not only labeled ties-based nodes' features, but also unlabeled ties-based nodes' features.

Suppose we want to predict the sign of the edge from i to j. Similar to [9], we consider the number of incoming positive edges to j (denoted as $d_{in}^+(j)$), the number of incoming negative edges to j (denoted as $d_{in}^-(j)$), the number of outgoing positive edges from i (denoted as $d_{out}^+(i)$), and the number of outgoing negative edges from i (denoted as $d_{out}^-(i)$). Here we just consider the labeled links to extract the above features. If the network is incomplete, we will acquire few labeled ties-based nodes' features.

Therefore, besides labeled ties-based nodes' features, we also consider the following unlabeled ties-based nodes' features: the number of incoming unlabeled edges to j (denoted as $d_{in}^?(j)$), the number of outgoing unlabeled edges from i (denoted as $d_{out}^?(i)$), and the total number of common neighbors of i and j (denoted as $cn(i, j)$). When extracting the above features, we do not consider the signs of edges.

2) STRUCTURAL BALANCE-BASED FEATURES

The principles of the structural balance theory are based on the following psychology: "my friend's friend is my friend", "my friend's enemy is my enemy", "my enemy's friend is my enemy", and "my enemy's enemy is my friend". Based on the discussion in [6], the directions of links are ignored in the study of balance theory because balance theory is designed for undirected networks.

The structural balance theory is often studied in connected signed triads. It has been proven that a balanced structure has odd number of occurrences for positive edges. Suppose we want to predict the sign of the edge between i and j. As shown in Figure 6, there are four kinds of triads involving the edge (the directions of links are ignored). The structural balance theory is based on the principle that it is more likely to assign a sign to the target link which can keep the triad

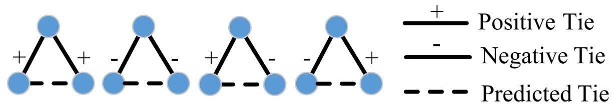

FIGURE 6. Triad features based on structural balance theory.

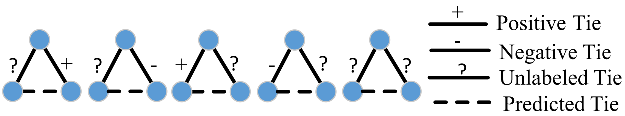

FIGURE 7. Triad features extending structural balance theory.

relations balanced. So the signs can be assigned by maximizing the balance of all such triads.

In order to measure the balance of triads involving the edge (i, j), we define the features B^+ and B^- (shown in (10) and (11)), which are the total number of balanced triangles in G^S in the case of assigning '+' and '-' to the edge (i, j) respectively. Some triads favor a negative sign and some favor a positive sign. In such situations, we choose the sign resulting in more balanced triads. The sign of (i, j) can be inferred according to B^+ and B^- (shown in (12)).

$$B^+(i,j) = \sum_k I[(i,j,k)\ is\ balanced\ |sign(i,j) = '+'], \tag{10}$$

$$B^-(i,j) = \sum_k [(i,j,k)\ is\ balanced\ |sign(i,j) = '+'], \tag{11}$$

$$sign(i,j) = sign\left(B^+(i,j) - B^-(i,j)\right), \tag{12}$$

In order to simplify (12), we use an alternate method (shown in (13)) by computing $(A^S_{ij})^2$, where A^S is the signed adjacency matrix.

$$\begin{aligned} sign(i,j) &= sign\left(\sum\nolimits_k sign(i,k) \times sign(k,j)\right) \\ &= sign((A^S_{ij})^2), \end{aligned} \tag{13}$$

The above structural balance features are extracted based on the labeled links. In addition to these features, we also take into account the unlabeled ties in G^S to extract more structural balance features. So, besides the triads described in Figure 6, there are another five types of triads involving (i, j) (shown in Figure 7). They include the situations which triads are made up of unlabeled edges. As for the triads in Figure 6, we have extracted features from them according to (10)-(13). As for the triads in Figure 7, we use a 5-dimensional vector in which each element represents the number of triads with the corresponding type.

3) STATUS-BASED FEATURES

The directions of links are ignored in the study of structural balance theory. While, social status theory specifically deals with directed signed networks which can better make up for structural balance theory. A positive edge from i to j means that i has lower status compared to j. On the contrary, if user i expresses a negative link to user j, then j will assert that i has higher status. For the link from i to j, the sign of it will be $+1$ if the status of j is higher than i. Otherwise, the sign of it will be -1.

In order to represent status-based features, we consider all of triads involving the edge (i, j) and use $S(i, j)$ to denote the status feature of (i, j) (shown in (14)).

$$S(i,j) = \sum\nolimits_k sign(i,k) + sign(k,j), \tag{14}$$

The above status-based feature is extracted based on the labeled links. We also extend it by considering the unlabeled ties in G^S. For each triad in Figure 7, four new triads are generated. That is because, the edge between i and k can be in either direction, and the edge between k and j can also be in either direction. Thus, there are $2\times2\times5 = 20$ distinct types of triads involving (i, j). We use a 20-dimensional vector to specify the number of triads of each type that (i, j) is involved in.

B. SIGN PREDICTION ALGORITHM BASED ON TRANSFER LEARNING

Usually, the sign prediction problem is considered as a classification problem. However, in mainstream signed networks, only a very small amount of sign information is available, which is not adequate to train a good classifier. In this section, with the extracted features, we propose a sign prediction algorithm based on transfer learning. As described in Section III, the process of sign prediction includes the training phase and the predicting phase. Correspondingly, the algorithm includes the two phases too.

During the training phase, we adopt transfer learning framework to train the classifier, in which knowledge is extracted from related source networks to train the target network. The aim is to effectively make up for the incompleteness of target samples. More specifically, we borrow the idea of TrAdaboost [42], a transfer learning model, to train our classifier. Here we use S and T to denote a set of training data from source and target respectively. Both of them include pairs of (p_i, q_i) $(i = 1 \sim n)$, where p_i is a feature vector of an edge i and $q_i \in \{1, -1\}$ is the sign of the edge. Given S and T, the output of sign prediction algorithm is a set of classifiers $\{C_1, \ldots, C_K\}$ and coefficients $\{\beta^1_T, \ldots, \beta^K_T\}$. Here each classifier contains a group of weights learned for extracted features. K is the number of iterations. We use β^z_T $(z = 1 \sim K)$ to denote the coefficient (i.e. weight) assigned to the z-th iteration. The pseudocode is shown in Algorithm 1.

Step 1: Initialization. The set of candidate classifiers C, coefficients β_T and β_S, the weight set WS^1 for source sample data and the weight set WT^1 for target sample data are initialized (Line 1-6).

Step 2: Iterative optimization. A set of classifiers and coefficients are learned by iterative optimization. More specifically, for the j-th iteration $(j = 1 \sim K)$,

Firstly, WS^j and WT^j are normalized (Line 8).

Secondly, a classifier (i.e. the j-th classifier C_j in C) is trained (Line 9). We will acquire C_j which is used to label each edge in the training set according to its feature vector.

Algorithm 1 Sign Prediction Algorithm Based on Transfer Learning

Input: Source data $S = \{p_i, q_i\}_{i=1}^{S_N}$, target data $T = \{p_i, q_i\}_{i=1}^{T_N}$, the maximum number of iterations K
Output: Set of classifiers $\{C_1, \ldots, C_K\}$ and coefficients $\{\beta_T^1, \ldots, \beta_T^K\}$
Process:
//Empty the set of candidate classifiers C and coefficients β_T
1. $C \leftarrow \emptyset$; $\beta_T \leftarrow \emptyset$;
//Initialize the weight set WS^1 and WT^1 for source data and target data respectively
2. For $i \leftarrow 1$ to S_N do
3. $WS_i^1 \leftarrow 1/S_N$;
4. For $i \leftarrow 1$ to T_N do
5. $WT_i^1 \leftarrow 1/T_N$;
6. $\beta_S \leftarrow 1/(1+(2 \ln n/K)^{1/2})$;
// Iterative training
7. For $j \leftarrow 1$ to K do
8. Normalize WS^j and WT^j;
9. $C_j \leftarrow$ LearnWeights(S, T, WS^j, WT^j, SVM);
10. $\varepsilon_j \leftarrow$ ComputeErrorRate(S, WS^j, C_j);
11. $\beta_T^j \leftarrow \varepsilon_j$ / (1-ε_j);
12. For $z \leftarrow 1$ to S_N do
13. $WS_z^{j+1} \leftarrow$ UpdateSourceWeight(WS_z^j, β_S, C_j, S_z);
14. For $z \leftarrow 1$ to T_N do
15. $WT_z^{j+1} \leftarrow$ UpdateTargetWeight(WT_z^j, β_T^j, C_j, T_z);
16. $C \leftarrow C \cup \{C_j\}$;
17. $\beta_T \leftarrow \beta_T \cup \{\beta_T^j\}$;
Return C and β_T.

Because SVM has been proven as a highly noise-tolerant classifier, we choose it as our basic classifier. By training, the parameters of the classifier C_j (the weights for features) can be learned by maximizing the margin, i.e. by finding the optimal hyperplane with maximal margin between positive signed instances and negatively signed instances. Based on the method proposed in [43], we adopt positive slack variables ς_i $(i = 1 \sim n)$. Then the objective function subject to the constraints can be defined as (15), where W is the weight vector for the extracted features. Also during parameter learning, we assign different weights, i.e. WS^j and WT^j, to the source data S and the target data T respectively.

$$\begin{aligned} &\min_{W,b,\varsigma_i} \varsigma\varsigma \frac{||W||_2^2}{2} + \sum\nolimits_i \varsigma_i \\ &s.t.\ q_i (W \cdot p_i + b) \geq 1 - \varsigma_i, \\ &\varsigma_i \geq 0, \quad i = 1 \sim n \end{aligned} \tag{15}$$

Thirdly, the coefficient β_T^j, the source weight set WS^j and the target weight set WT^j are updated respectively for the next iteration. More specifically, by comparing the result of classifier $C_j(p_z)$ with the real label q_z $(z = 1 \sim n)$, the error rate ε_j *of* C_j can be computed (shown in (16), Line 10).

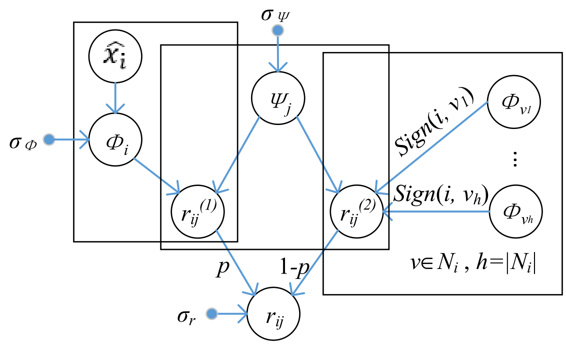

FIGURE 8. The basic idea of our behavior prediction algorithm.

Then the coefficient β_T^j can be computed (Line 11). And the values of WS^{j+1} and WT^{j+1} are recomputed and updated for the next iteration (shown in (17) and (18), Line 12-15). Here, the instances in S and T play different roles during training. For those instances in S that are more similar to the target instances in T, we should attach more importance to them. That is, their weights will be higher.

$$\varepsilon_j = \frac{\sum_{z=1}^{S_N} WS_z^j \times |C_j(p_z) - q_z|}{2 \times \sum_{z=1}^{S_N} WS_z^j}, \tag{16}$$

$$WS_z^{j+1} = WS_z^j \times \beta_S^{|C_j(p_z) - q_z|/2}, \tag{17}$$

$$WT_z^{j+1} = WT_z^j \times (\beta_T^j)^{-|C_j(p_z) - q_z|/2}, \tag{18}$$

Finally, add the current classifier C_j and coefficient β_T^j to the result (Line 16-17).

Based on the coefficients β_T^j and the classifiers C_j $(j = 1 \sim K)$ learned during the training phase, we can predict the sign of each edge z with its feature vector p_z (shown in (19)).

$$sign(z|p_z) = sign(\sum\nolimits_{j=1}^{K} \log(1/\beta_T^j) \times C_j(p_z)), \tag{19}$$

VI. BEHAVIOR PREDICTION

The basic idea of our behavior prediction algorithm is shown in Figure 8. It is a linear combination of matrix factorization approach and sign based approach. Correspondingly, it includes two parts: matrix factorization and decision making. On one hand, the rating matrix is factorized into the user feature matrix Φ and the item feature matrix Ψ. We fuse users' representations, i.e. $\hat{x}_i$, to the process of matrix factorization. Different from traditional matrix factorization method, we consider users' representations as the social contents of users, which can improve the accuracy of matrix factorization. However, as described in Section IV.A, users' representations are generated based on the adjacency vector s. That is, only the adjacency between users rather than the signs of links between them is considered. Therefore, on the other hand, we not only take into account the neighbors of each user i (denoted as N_i), but also distinguish between positive links and negative links, which can provide more informative evidence for behavior prediction. The final rating r_{ij} is the linear combination of the result of matrix factorization (i.e. $r_{ij}^{(1)}$) and the result of sign-based prediction (i.e. $r_{ij}^{(2)}$). The former represents user i's own interest in item j. The latter indicates the interest of user i's neighbors (both trusted and distrusted) in item j. We use the parameter p to control their effects on the final rating (i.e. $r_{ij} = p * r_{ij}^{(1)}$+(1-$p$) $* r_{ij}^{(2)}$).

The pseudocode of our behavior prediction algorithm is shown in Algorithm 2. Given a behavioral network $G^B = (U \cup I, E^B)$, a signed network $G^S = (V^S, E^S, A^S)$, the user representation set $\hat{x}$ and the parameter p, the goal is to predict the behavior of each user.

Algorithm 2 Behavior Prediction Algorithm

Input: A behavioral network $G^B = (U \cup I, E^B)$, a signed network $G^S = (V^S, E^S, A^S)$, the user representation set $\hat{x}$, the parameter p
Output: The result set B
Process:
1. $B \leftarrow \emptyset$;
2. $\Phi, \Psi \leftarrow$ MatrixFactorization($G^B, \hat{x}$);
3. For each node i in U do
4. For each node j in I do
5. $r_{ij}^{(1)} \leftarrow \Phi_i^T \Psi_j$;
6. $r_{ij}^{(2)} \leftarrow 0$;
7. $N_i \leftarrow$ FindNeighbors(i, G^S);
8. For each node v in N_i do
9. $r_{ij}^{(2)} \leftarrow r_{ij}^{(2)} + Sign(i, v) * \Phi_v^T \Psi_j$;
10. $r_{ij} \leftarrow p * r_{ij}^{(1)} +$ (1-p) $* r_{ij}^{(2)}$;
11. $B \leftarrow B \cup (i, j, Sigmoid(r_{ij}))$;
Return B.

Step 1: Initialize the result set B (Line 1).

Step 2: Factorize the rating matrix E^B (Line 2). We introduce the representation $\hat{x}_i$ of each user to the process of matrix factorization. The loss function is defined as (20), where τ_{ij} is the indicator indicating whether user i has a historical rating for item j (If so, its value is 1. Otherwise, its value is 0.). τ_u and τ_v are the regularized parameters, which can control the effect of the social information on the rating. By minimizing the value of the loss function, the user feature matrix Φ and the item feature matrix Ψ can be calculated.

$$\mathcal{L}_{MF} = \min \sum_i \sum_j \tau_{ij}(r_{ij} - \Phi_i^T \Psi_j)^2 \Big/ 2 + \tau_u \left(\Phi_i - \hat{x}_i\right)^T (\Phi_i - \hat{x}_i) \Big/ 2 + \tau_v \Psi_j^T \Psi_j \Big/ 2, \quad (20)$$

Step 3: For each user i and each item j, compute the probability that i is interested in j (Line 3-11). We consider the effect of the feature vectors of direct neighbors of i (denoted as Φ_v). Here we use the sign of the link between them (i.e. $Sign\ (i, v)$) to measure the trust degree from i to v. If the sign is positive, it will show that i trusts v and the rating of user v on item j will improve the value of r_{ij}. Otherwise, v will be the one whom i distrusts and the rating of user v on item j will have a negative effect. By linear combining, the predicted rating of user i on item j is as (21). Finally, we use the Sigmoid function [44] acting on r_{ij} to calculate the probability value of i's behavioral decision on j.

$$r_{ij} = p\Phi_i^T \Psi_j + (1-p) \sum_{v \in N_i} Sign(i, v) \times (\Phi_v^T \Psi_j), \quad (21)$$

TABLE 2. Datasets.

	Epinions	Slashdot	Wiki-RfA
Nodes	75951	15999	6998
Edges	690179	371122	121151
Average degrees	64.462	46.394	34.623

VII. EXPERIMENTS

In this section, we conduct experimental studies to evaluate the effectiveness of our proposed model.

A. DATASET

We implement the experiments on a PC with Intel Core i7-2600 @ 3.40GHZ and 8GB main memory. We evaluate the effectiveness of our model on three data sets: Epinions, Slashdot and Wiki-RfA (Table 2).

Epinions: It is the only publicly available dataset (from www.epinion.com) we are aware of that contains both signed social relationship and user behavioral data. Epinions is a well-known knowledge sharing site and review site. Users can assign products or reviews integer ratings from 1 to 5. The site maintains a signed social network that allows each user to indicate explicitly other users that he is connected to as either positive (trust) or negative (distrust).

Slashdot: Slashdot is a social news website. It features news stories on science, technology, and politics that are submitted and evaluated by site users and editors. In 2002, Slashdot introduced ''Slashdot Zoo'', which allows users to add another user to a list of friends or enemies based on his or her comments. The users in this network form a signed network.

Wiki-RfA: Wiki-RfA [16] (Wikipedia Requests for Adminship) is a signed network, corresponding to votes cast by Wikipedia uses in election for promoting individuals to the role of administrator. A signed link indicates a positive or negative vote by one user on the promotion of another.

Because Epinions includes both social information and behavioral information, it can be used for both sign prediction and behavior prediction. For the latter two datasets, due to the lack of the ratings assigned by users, we are not able to use them to evaluate behavior prediction. Instead, we will use them for sign prediction only.

For most signed networks, there are more positive links than negative signs, which might bias the results. So, for each negative edge, we randomly select a positive edge to be retained in the dataset which can ensure the same ratio as negative edges. In order to avoid over-fitting, we adopt 10-fold cross validation. Ten disjoint test folds each consisting of 10% of edges are created. For each test fold, the remaining 90% of edges serve as the training set.

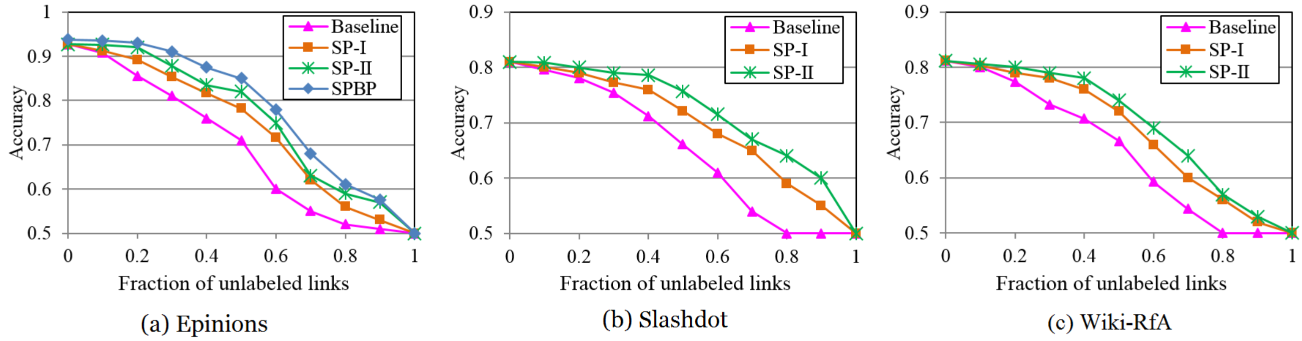

FIGURE 9. **Effectiveness evaluation of different sign prediction models.**

B. EXPERIMENTAL RESULTS OF SIGN PREDICTION

In sign prediction setting, our task is to predict the sign of an unlabeled link between two given nodes. As the existing links in the original network are known and can serve as the ground truth, we randomly hide some of links in the signed network (let them become unlabeled links) and select a balanced test set out of them, while use the remaining network to train our model. We use accuracy as the evaluation metrics in sign prediction task.

1) EFFECTIVENESS EVALUATION OF SIGN PREDICTION MODELS

We varied the percentage of hidden links, that is, the percentage of unlabeled links for sign prediction (set as 10%∼90%). We evaluate the quality of the result produced by the following different models.

(1) Baseline: Baseline only considers labeled links in the signed network to predict unlabeled links.

(2) SP-I: Based on social psychology theories, the features of both labeled links and unlabeled links in the signed network are encoded.

(3) SP-II: Based on SP-I, social correlation is also considered as one feature for sign prediction. Here we design a 4-layer autoencoder to acquire users' representations, which can avoid heavier computational overhead.

(4) SPBP: Based on SP-I, both social correlation and behavioral correlation are taken into account. That is, it simultaneously captures users' social links and users' behavior to predict the signs of links. Note that since Slashdot and Wiki-RfA lack the rating data, behavioral correlation cannot be estimated. So there are not experimental results of SPBP on Slashdot and Wiki-RfA.

Results: The accuracy of different models are illustrated in Figure 9. Due to the lack of user behavior information in Slashdot and Wiki-RfA, we only evaluate the first three models on these two datasets. The variation trend of experimental result on three datasets is similar. A random assignment process will result in an accuracy of 50% on average. As shown in Figure 9, all the models can outperform the random assignment process. Baseline just uses the features of labeled links. With the increase of the percentage of unlabeled links, the evidence becomes less and less, resulting in lower accuracy. SP-I improves upon Baseline by taking into account the features of unlabeled links. But it only considers the features based on social psychology theories and ignores the correlation between users. SP-II improves upon SP-I by taking into account social correlation between users. However, it still ignores behavioral correlation between users. SPBP further improves upon SP-II by considering both social correlation and behavioral correlation. It makes full use of users' representations and behavior to estimate social correlation and behavioral correlation respectively, leading to the best performance.

2) PERFORMANCE EVALUATION OF SIGN PREDICTION ALGORITHMS

We evaluate the performance of sign prediction algorithms with different fractions of target samples. Since only the dataset Epinions in the three datasets contains users' behavior information, we only consider the non-user behavior features in Epinions for prediction when transferring from Epinions to the other or transferring from the other to Epinions. The following three sign prediction algorithms are compared.

(1) SRC: Only source samples are used as training data.

(2) TRT: Only target samples are used as training data.

(3) TL: Both source samples and target samples are used as training data. The idea of transfer learning model, TrAdaboost, is borrowed to train our model.

Results: The accuracy of the above algorithms are illustrated in Figure 10. With the decrease of target samples, the accuracy of most algorithms (except for SRC which only considers source samples) is reducing. This shows the lack of training data in the target network will lead to inaccuracy of the result of prediction. In spite of this, our transfer learning-based sign prediction algorithm (TL) can still maintain higher accuracy. This is because our transfer learning-based sign prediction algorithm can use knowledge extracted from related source networks to train the target network. Source samples can make up for the incompleteness of target samples, resulting in higher accuracy. We also find that when the distribution difference between the source network and the

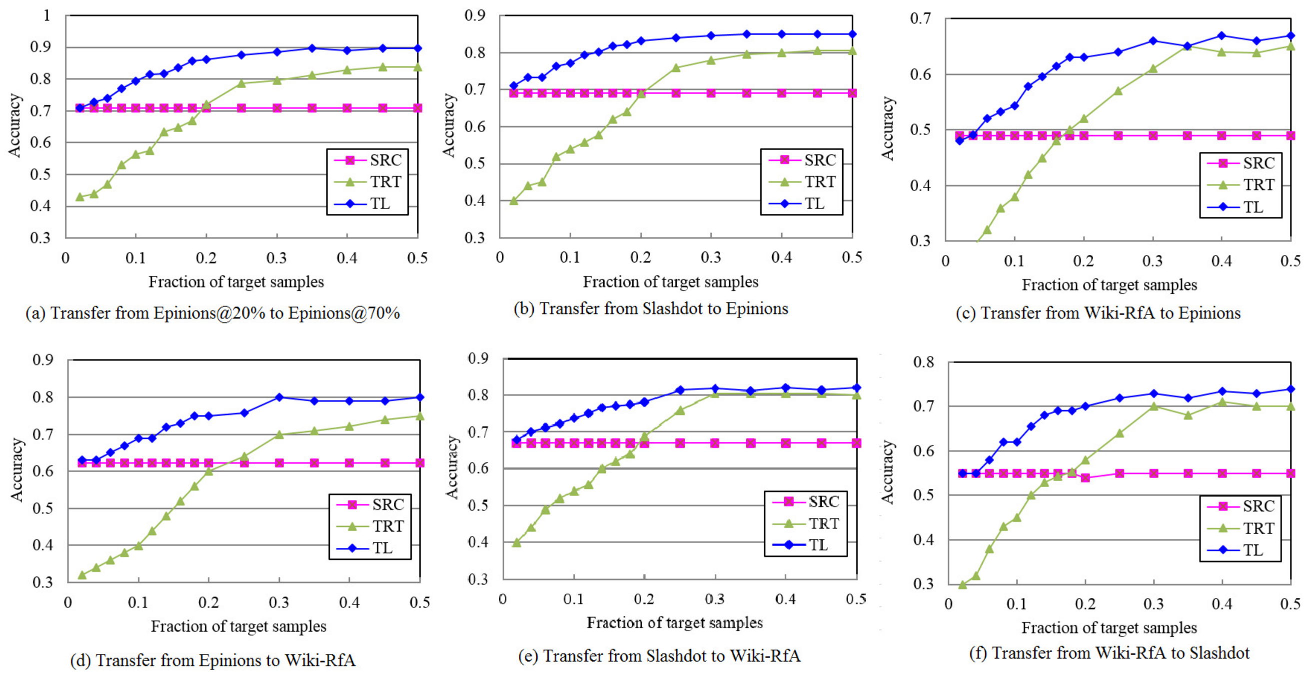

FIGURE 10. **Accuracy comparison of different sign prediction algorithms with different fractions of target samples.**

target network is greater, the result of using transfer learning is worse. For example, the differences between Slashdot and Wiki-RfA are greater than the differences between Epinions and Epinions itself. In Figure 10(a)-(c), Epinions is used as the target network to evaluate the accuracy of sign prediction. In Figure 10 (a), we use Epinions containing 20% unlabeled links (denoted as Epinions@20%) and containing 70% unlabeled links (denoted as Epinions@70%) as the source network and the target network respectively. That is, Epinions is not only the target network, but also the source network, with an average accuracy rate of 82.91%. In Figure 10(b) and (c), Slashdot and Wiki-RfA are used as the source networks respectively. Their average accuracy is 80.37% and 59.78% respectively. In Figure (d), Epinions is used as the source network, and Wiki-RfA as the target network, with an accuracy rate of 72.74%. In Figure (e), Slashdot is used as the source network, and Wiki-RfA as the target network, with an accuracy rate of 76.78%. In Figure 10(f), Wiki-RfA is used as the source network, and Slashdot as the target network, with an accuracy rate of 66.94%. This shows that the effect is better when the source network is less different from the target network.

C. EXPERIMENTAL RESULTS OF BEHAVIOR PREDICTION

In addition to sign prediction, we also conduct experiments on behavior prediction. For each user we aim to recommend a set of items whose ratings are not explicitly given by him. We evaluate the performance of behavior prediction in terms of mean absolute error (MAE) and root mean square error (RMSE) on the dataset Epinions.

1) IMPACT OF PARAMETERS ON THE RESULTS

In our behavior prediction algorithm, we use the regularized parameters, τ_u and τ_v, to control the effect of the social information on the rating. Here we evaluate the impact of them on the results. In order to reduce the computational complexity, we set $\tau_u = \tau_v$. And we set the dimensionality of the factorized feature vectors to 5.

Results: Figure 11 shows the impacts of τ_u (or τ_v) on MAE and RMSE respectively. We observe that the value of τ_u (or τ_v) impacts the prediction results significantly. As τ_u (or τ_v) increases, the value of MAE (or RMSE) decreases at first, but when τ_u (or τ_v) surpasses a certain threshold, the value of MAE (or RMSE) increases with further increase of τ_u (or τ_v). This phenomenon coincides with the intuition that purely using the user-item rating matrix or purely using the user social information cannot generate better performance than fusing these two resources together. From Figure 11, our behavior prediction method achieves the best performance when τ_u (or τ_v) $= 0.15$.

2) PERFORMANCE EVALUATION OF BEHAVIOR PREDICTION ALGORITHMS

We evaluate the performance of behavior prediction algorithms with 5-dimensional ($d = 5$) and 10-dimensional ($d = 10$) feature vectors respectively. The following four behavior prediction algorithms are compared.

(1) CF: CF is based on the idea of collaborative filtering.

(2) MF: MF is the basic matrix factorization algorithm, which does not take the social information between users into account.

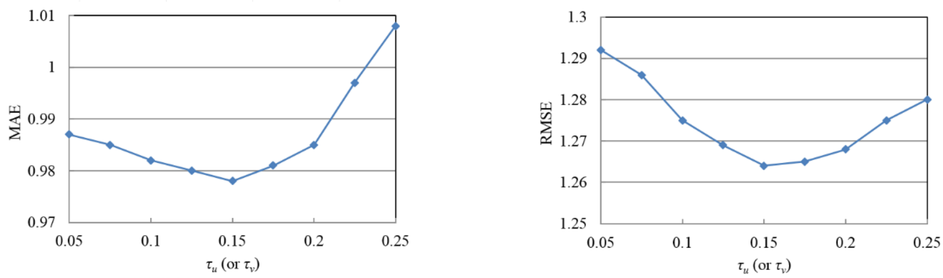

FIGURE 11. The impacts of τ_u (or τ_v) on MAE and RMSE.

TABLE 3. Performance of different behavior prediction algorithms.

Algorithm	MAE		RMSE	
	d=5	d=10	d=5	d=10
CF	1.15	1.15	1.4	1.4
MF	1.05	0.89	1.29	1.12
AdjMF	0.98	0.86	1.27	1.11
SignMF	0.97	0.84	1.22	1.06

(3) AdjMF: Based on MF, AdjMF also takes into account users' representations as the social contents of users. These users' representations are generated based on the adjacency vector s. So the adjacency between users is considered during the process of matrix factorization.

(4) SignMF: Based on AdjMF, SignMF also considers the signs of links between users. That is, it not only takes into account the neighbors of each user, but also distinguishes between positive links and negative links.

Results: The performance of different behavior prediction algorithms on Epinions are illustrated in Table 3. Note that since the algorithm CF has no latent features, there is no dimensionality associated with it and hence the result for different values of dimensionality are the same. Due to the sparsity of the user-item rating matrix, CF based behavior prediction algorithm fails to find similar users, resulting in higher MAE (or RMSE). The algorithm MF improves upon CF by learning latent features for users and items from the observed ratings, especially when the rating matrix is sparse. But it ignores the social links between users. AdjMF improves upon MF by taking into account the users' representations and the adjacency between users. However, it still ignores the signs of links between users. SignMF further improves upon AdjMF by distinguishing whether the links are positive or negative. The signs of links can provide more informative evidence for behavior prediction.

VIII. CONCLUSION

In this paper, we present a novel model called SPBP to integrate sign prediction with behavior prediction in the context of signed heterogeneous information networks. Both users' social links and users' behavior are captured to improve the accuracy of prediction. First, we propose correlation estimation methods to estimate social correlation and behavioral correlation between users respectively. Then we encode structural balance-based features and status-based features according to social psychology theories by considering both labeled links and unlabeled links. With the extracted features, we propose a sign prediction algorithm based on transfer learning. We use knowledge extracted from related source networks to train the target network, which can effectively make up for the incompleteness of target samples. Finally, we propose a behavior prediction algorithm by taking into account the social links between users and their signs.

Our future work will mainly focus on the alternate iteration between sign prediction and behavior prediction. In addition, we will conduct further research into the incremental maintenance of the predicted results.

REFERENCES

[1] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *Proc. WWW*, New York, NY, USA, May 2004, pp. 403–412.

[2] P. Massa and P. Avesani, "Controversial users demand local trust metrics: An experimental study on epinions. com community," in *Proc. AAAI*, Pittsburgh, PA, USA, Jul. 2005, pp. 121–126.

[3] M. J. Brzozowski, T. Hogg, and G. Szabo, "Friends and foes: Ideological social networking," in *Proc. SIGCHI*, Florence, Italy, Apr. 2008, pp. 817–820.

[4] J. Kunegis, A. Lommatzsch, and C. Bauckhage, "The Slashdot zoo: Mining a social network with negative edges," in *Proc. WWW*, Madrid, Spain, Apr. 2009, pp. 741–750

[5] M. Burke and R. Kraut, "Mopping up: Modeling Wikipedia promotion decisions," in *Proc. CSCW*, San Diego, CA, USA, Nov. 2008, pp. 27–36.

[6] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in Online social networks," in *Proc. WWW*, Raleigh, CA, USA, Apr. 2010, pp. 641–650

[7] D. Cartwright and F. Harary, "Structural balance: A generalization of Heider's theory," *Psychol. Rev.*, vol. 63, no. 5, pp. 277–293, 1956.

[8] S. H. Yang, A. J. Smola, and B. Long, "Friend or frenemy?: Predicting signed ties in social networks," in *Proc. SIGIR*, Portland, OR, USA, Aug. 2012, pp. 555–564.

[9] D. Li, D. Shen, Y. Kou, Y. Shao, T. Nie, and R. Mao, "Exploiting unlabeled ties for link prediction in incomplete signed networks," in *Proc. IRC*, Naples, Italy, Feb. 2019, pp. 538–543.

[10] W. Yuan, K. He, D. Guan, L. Zhou, and C. Li, "Graph kernel based link prediction for signed social networks," *Inf. Fusion*, vol. 46, pp. 1–10, Mar. 2019.

[11] C. Si, L. Jiao, J. Wu, and J. Zhao, "A group evolving-based framework with perturbations for link prediction," *Phys. A, Stat. Mech. Appl.*, vol. 475, pp. 117–128, Jun. 2017.

[12] A. Nocaj, M. Ortmann, and U. Brandes, "Adaptive disentanglement based on local clustering in small-world network visualization," *IEEE Trans. Vis. Comput. Graph*, vol. 22, no. 6, pp. 1662–1671, Jun. 2016.

[13] Y.-L. He, J. N. K. Liu, Y.-X. Hu, and X.-Z. Wang, "OWA operator based link prediction ensemble for social network," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 21–50, Jan. 2015.

[14] J. Tang, S. Chang, C. Aggarwal, and H. Liu, "Negative link prediction in social media," in *Proc. WSDM*, Shanghai, China, Feb. 2015, pp. 87–96.
[15] S. Wang, J. Tang, C. Aggarwal, Y. Chang, and H. Liu, "Signed network embedding in social media," in *Proc. SIAM*, Pennsylvania, PA, USA, Jun. 2017, pp. 327–335.
[16] R. West, H. Paskov, J. Leskovec, and C. Potts, "Exploiting social network structure for person-to-person sentiment analysis," *Trans. Assoc. Comput. Linguistics*, vol. 2, pp. 297–310, Dec. 2014.
[17] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, "SHINE: Signed heterogeneous information network embedding for sentiment link prediction," in *Proc. WSDM*, Marina Del Rey, CA, USA, Feb. 2018, pp. 592–600.
[18] C. Lu, J. Yu, and R.-H. Li, "Exploring hierarchies in Online social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2086–2100, Aug. 2016.
[19] V. Martnez, F. Berzal, and J. Cubero, "A survey of link prediction in complex networks," *ACM Comput. Surv.*, vol. 49, no. 4, Feb. 2017, Art. no. 69.
[20] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proc. SIGCHI*, Atlanta, GA, USA, Apr. 2010, pp. 1361–1370.
[21] A. Shrivastava and P. Li, "A new mathematical space for social networks," in *Proc. NIPS*, Lake Tahoe, NV, USA, 2013, pp. 1–7.
[22] C.-J. Hsieh, K.-Y. Chiang, and I. S. Dhillon, "Low rank modeling of signed networks," in *Proc. SIGKDD*, Beijing, China, Aug. 2012, pp. 507–515.
[23] R. Jin, J. Y. Chai, and L. Si, "An automatic weighting scheme for collaborative filtering," in *Proc. SIGIR*, Sheffield, U.K., Jul. 2004, pp. 337–344.
[24] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.
[25] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proc. SIGIR*, Amsterdam, The Netherlands, Jul. 2007, pp. 39–46.
[26] J. Wang, A. P. Vries, and M. J. T. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proc. SIGIR*, Seattle, WA, USA, Aug. 2006, pp. 501–508.
[27] T. Hofmann, "Collaborative filtering via Gaussian probabilistic latent semantic analysis," in *Proc. SIGIR*, Toronto, ON, Canada, Aug. 2003, pp. 259–266.
[28] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 89–115, Jan. 2004.
[29] R. R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Advances in Neural Information Processing Systems*, vol. 20. Red Hook, NY, USA: Curran Associates, 2008, pp. 1257–1264.
[30] L. Si and R. Jin, "Flexible mixture model for collaborative filtering," in *Proc. ICML*, Washington, DC, USA, 2003, pp. 704–711.
[31] J. D. M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proc. ICML*, Bonn, Germany, Aug. 2005, pp. 713–719.
[32] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. ICML*, Helsinki, Finland, Jul. 2008, pp. 880–887.
[33] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proc. ICML*, Washington, DC, USA, 2003, pp. 720–727.
[34] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. SIGKDD*, San Diego, CA, USA, Aug. 2011, pp. 448–456.
[35] X. Li and J. She, "Collaborative variational autoencoder for recommender systems," in *Proc. SIGKDD*, Halifax, NS, Canada, Aug. 2017, pp. 305–314.
[36] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," in *Proc. CoopIS/DOA/ODBASE*, Napa, Cyprus, 2004, pp. 492–508.
[37] P. Bedi, H. Kaur, and S. Marwaha, "Trust based recommender system for semantic Web," in *Proc. IJCAI*, Hyderabad, India, Jan. 2007, pp. 2677–2682.
[38] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. CIKM*, Napa Valley, CA, USA, Oct. 2008, pp. 931–940.
[39] M. Jamali and L. Lakshmanan, "HeteroMF: Recommendation in heterogeneous information networks using context dependent factor models," in *Proc. WWW*, Rio de Janeiro, Brazil, May 2013, pp. 643–654.
[40] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. RecSys*, Barcelona, Spain, Sep. 2010, pp. 135–142.
[41] R. Salakhutdinov and G. Hinton, "Semantic hashing," *Int. J. Approx. Reasoning*, vol. 50, no. 7, pp. 969–978, Jul. 2009.
[42] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.
[43] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[44] H. Jun and M. Claudio, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *Proc. IWANN*, Malaga Torremolinos, Spain, 1995, pp. 195–201.

DONG LI was born in 1979. He received the master's degree in computer technology from Northeastern University, Shenyang, in 2008. He is currently pursuing the Ph.D. degree. His research interests include social networks analysis and data mining.

DERONG SHEN was born in 1964. She received the Ph.D. degree in computer software and theory from Northeastern University, Shenyang, in 2004, where she is currently a Professor with the School of Computer Science and Engineering. Her research interests include social networks analysis and data integration. She is a Senior Member of the CCF and ACM.

YUE KOU was born in 1980. She received the Ph.D. degree in computer software and theory from Northeastern University, Shenyang, in 2009, where she is currently an Associate Professor with the School of Computer Science and Engineering. Her research interests include social networks analysis and data mining. She is a member of the CCF and ACM.

TIEZHENG NIE was born in 1980. He received the Ph.D. degree in computer software and theory from Northeastern University, Shenyang, in 2009, where he is currently an Associate Professor with the School of Computer Science and Engineering. His research interests include data integration and data mining. He is a member of the CCF and ACM.