The Computer Journal Advance Access published June 27, 2013

© The British Computer Society 2013. All rights reserved.
For Permissions, please email: journals.permissions@oup.com
doi:10.1093/comjnl/bxt062

A Parallel Approach to Link Sign Prediction in Large-Scale Online Social Networks

JIUFENG ZHOU[1], LIXIN HAN[1,*], YUAN YAO[2,3], XIAOQIN ZENG[1] AND FENG XU[2,3]

[1]*Department of Computer Science and Engineering, Hohai University, Nanjing, Jiangsu, People's Republic of China*
[2]*State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, Jiangsu, People's Republic of China*
[3]*Department of Computer Science and Technology, Nanjing University, Nanjing, Jiangsu, People's Republic of China*
**Corresponding author: lixinhan2002@hotmail.com*

Analyzing the underlying social network is very important for the development of online applications. Owing to the increasingly growing size of these networks, parallel techniques play important roles in many network analysis tasks. In this paper, we explore the link sign prediction problem in large-scale online social networks, and propose a parallel approach, called PLSP, to solve the problem. Specifically, we first extract a set of features that serve as a base for prediction. Experiments on several real datasets show that these features outperform those proposed by existing methods in predictive accuracy. Next, we present two speedup strategies, i.e. dataset division and feature selection, to shorten the training time. Experimental evaluations show that our parallel approach is much faster than the traditional non-parallel method and achieves higher predictive accuracy than other methods at the same time.

Keywords: social network analysis; link sign prediction; parallel training; feature selection

Received 6 June 2012; revised 23 May 2013
Handling editor: Franco Zambonelli

1. INTRODUCTION

Analyzing the underlying social network is very important for the development of many online applications [1, 2]. For example, studying the topology characteristics of social networks could be helpful to detect malicious users [3, 4]. However, many existing data-mining algorithms are proposed for traditional social networks, and these algorithms become very time-consuming or even infeasible, due to the increasingly growing size of online social networks. To this end, some parallel algorithms come into play in many analysis tasks of online social networks [5–7].

In online social networks, users can form links to others in order to maintain connections or express attitude. One of the important functions of online social networks in turn is to suggest potential connections to users (referred to as link prediction [8]). In a general view, link prediction only focuses on predicting positive attitude [9], while negative attitude (such as distrust) is also discovered in many networks [10, 11]. For example, in the online consumer review website Epinions.com, a user can tag others as trustworthy ('trust') or untrustworthy ('block') based on the product reviews written by them. In consequence, researchers propose link sign prediction [12], which is also the focus of this paper, in order to predict the latent binary attitude between two users.

Existing methods for link sign prediction can roughly be categorized into two classes: unsupervised [10, 13, 14] and supervised [12, 15]. In this paper, we use supervised methods based on two reasons: (1) the true signs of the links are available in many social networks, and (2) supervised methods can achieve higher predictive accuracy than unsupervised ones [12, 15]. Despite the success of existing supervised methods, more high-quality features need to be extracted to uncover the underlying principles of generating online signed social networks. More importantly, previous studies tend to ignore the

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

long running time caused by the huge amount of data during the training step. What makes the situation more severe is the necessity of retraining the prediction model frequently, since the available information in online social networks usually grows exponentially [16].

In this paper, a parallel approach (PLSP) is proposed to solve the link sign prediction problem. In PLSP, we first borrow the idea from balance theory [17, 18], status theory [19] and feedback transmission theory [20], and define 12 features consisting of 8 global ones and 4 local ones, and use them as the base for the prediction task. Experiments on real networks show that these 12 features, as a whole, significantly outperform those features of existing methods in predictive accuracy. Next, in order to train the dataset in parallel, the training dataset is divided into several subsets, each of which can be trained separately. To further reduce the training time, we also define several rules to select a subset of features (global ones, local ones or both) for these subsets. Finally, to predict the sign of a link, we choose the subset which this link belongs to and return the sign based on the locally trained model of the chosen subset. Experimental evaluations show that our method is much faster than the traditional non-parallel method and achieves higher predictive accuracy at the same time.

The rest of the paper is organized as follows. Section 2 covers the related work. Section 3 describes the details of our method. Section 4 presents the experimental results. Section 5 concludes the paper.

2. RELATED WORK

There are many hot prediction topics in social networks, and we divide these topics into three categories: network level, community level and individual level. At the network level, several studies have focused on the evolution of network structure [21]. Interestingly, such global structure is usually determined by local behavior: the preferential attachment leads the power-law degree distribution [21], the user reciprocity leads the positive degree correlation [2], etc. At the community level, Palla *et al.* [22] propose four community-based features to extract the overlapping communities from complex networks. With the help of group extraction, many researchers focus on the evolution of groups in the network. For example, Bródka *et al.* [23] study several states of groups and predict the evolution of groups between these states. Richter *et al.* [24] view users under certain mobile service as a group and predict which users are probably going to leave the group. At the individual level, a large body of work focuses on predicting the existence or signs of links between individuals. Such link prediction or link sign prediction problem could be further be divided into homogeneous network [8] such as friendship networks that are composed only by human beings and heterogeneous network [25] such as video-sharing networks that are composed by human beings and videos. In this paper, we put our focus on homogeneous network and further review the literature in the following.

Recommending new friends, which is formalized as the link prediction problem, is an important function of online social networks. Liben-Nowell and Kleinberg [8] summarize many features based on path and common neighbors between node pairs and employ these features for link prediction in an unsupervised way. To deal with dynamics, interdependence, and some other properties of networks, Lichtenwalter *et al.* [9] later propose a supervised machine-learning framework with features derived from degrees and mutual information between nodes. However, link prediction problem only focuses on positive links, whereas negative links have been widely integrated into the features of online social networks.

Taking the negative links into account, link sign prediction has been proposed and partially solved by various authors. Similar to link prediction problem, these proposals can also be categorized into two main classes: unsupervised and supervised. In the unsupervised category, Guha *et al.* [10] develop several propagation schemes based on the paths between a node pair and decide the sign based on several rounding strategies. Similarly, Symeonidis *et al.* [13] also try to find paths between the node pair. In contrast to Guha *et al.* [10], they use the notion of similarity to predict the sign. The above two methods cannot capture interdependency between features, which is very useful for link sign prediction [9]. In this paper, supervised machine-learning method is used to fix this problem.

As to the supervised category, Kunegis *et al.* [26] use various signed spectral similarity measures to predict the sign of the link in Slashdot. They also employ social network analysis techniques to study the characteristics (such as clustering, degree distribution and small-world property) of the network. However, these methods need to calculate the power of the adjacency matrix which is very time-consuming on large-scale online social networks. In contrast, we propose a parallel algorithm by decomposing the dataset into smaller subsets to shorten the overall training time. Recently, Leskovec *et al.* [12] suggest a set of features consisting of degree information and mutual information between node pairs for supervised learning, and they also employ the theory of balance [17, 18] and status [19] to analyze the link sign prediction problem from an unsupervised view. However, the two theories are based on the triad structure, which means that these methods are only effective when two nodes have common neighbors. To overcome this limitation, we extract eight global features, and these features are not affected by the common neighborhood size. While the balance theory is based on the triad structure, Chiang *et al.* [15] extend the theory to longer distance. We leave such extension as future work.

3. PLSP APPROACH

In this section, we first list the notation that would be used throughout the paper (see Section 3.1). We then explain how the

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

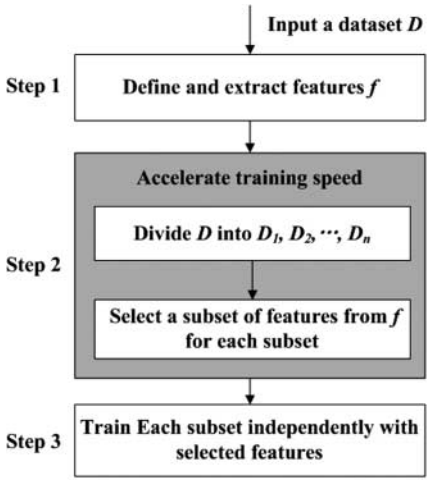

FIGURE 1. The flowchart of PLSP.

PLSP approach could be employed in practice (see Section 3.2). After that, the detail of our PLSP approach is presented, including feature extraction and parallel training for link sign prediction. The brief flowchart of our method is shown in Fig. 1. As we can see, given a dataset D, a set of features are first extracted to represent the relationships between each node pair (see Section 3.3). Two strategies are then employed to accelerate the training speed, i.e. dataset division (see Section 3.4.1) and feature selection (see Section 3.4.2). Finally, each subset is trained independently with the selected features, and the prediction of the link sign is based on the locally trained model.

3.1. Notation

Without loss of generality, the social network is regarded as a directed graph with a binary sign on each edge in this paper. Namely, the node in the graph stands for the participant in the social network, and the edge indicates the attitude (either positive or negative) between two participants.

By considering the dataset D (represented as a graph) in Fig. 2 as an example, some notation and abbreviations are denoted in Table 1.

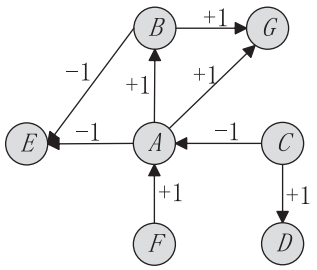

FIGURE 2. Signed social network.

We also adopt the following convention: the operation $|S|$ denotes the size of set S. For example, $|out(A)| = 3$ in the above example.

3.2. Employing PLSP in practice

Figure 3 shows an example that explains how to employ the PLSP algorithm practically.

(1) Convert the graph format into text format. We define '$\langle AB + 1\rangle$' as node A marks +1 on node B. Then the social network in Fig. 3 can be presented as follows:

$$\langle AB + 1\rangle, \langle AE - 1\rangle, \langle AG + 1\rangle, \langle BG - 1\rangle,$$
$$\langle CA - 1\rangle, \langle CF + 1\rangle, \langle DA + 1\rangle, \langle EB + 1\rangle, \langle ED - 1\rangle.$$

(2) Extract features. According to the method of extracting global features ($GF_i, 1 \leq i \leq 8$) and local features (LF_1, LF_2, LF_3, LF_4) suggested in Section 3.3, normalized values (one-tenth as accurate) of these features are presented as follows:

	GF_1	GF_2	GF_3	GF_4	GF_5	GF_6	GF_7	GF_8	LF_1	LF_2	LF_3	LF_4
$\langle AB + 1\rangle$	0.2	1.0	0.7	1.0	1.0	1.0	0.5	0.0	0.0	1.0	0.0	1.0
$\langle AE - 1\rangle$	0.2	0.2	0.7	1.0	0.0	0.0	0.5	0.5	1.0	0.8	1.0	0.0
$\langle AG + 1\rangle$	0.2	0.4	0.7	1.0	0.5	1.0	0.5	0.5	0.5	0.0	0.0	1.0
$\langle BG - 1\rangle$	1.0	0.4	0.0	0.0	0.5	1.0	1.0	0.5	0.5	0.0	1.0	0.0
$\langle CA - 1\rangle$	0.4	0.6	0.5	0.5	1.0	0.0	0.5	0.5	0.5	0.0	1.0	0.0
$\langle CF + 1\rangle$	0.4	0.6	0.5	0.5	1.0	0.0	0.5	0.5	0.5	0.0	1.0	0.0
$\langle EB + 1\rangle$	0.2	1.0	0.5	0.5	1.0	1.0	0.0	0.0	0.5	0.0	1.0	0.0
$\langle ED - 1\rangle$	0.2	0.0	0.5	0.5	0.0	0.0	0.0	0.0	0.5	0.0	1.0	0.0

(3) Divide datasets and select a subset of features. As the real social network are very huge (one of the datasets we use in Section 4 has more than 100 000 nodes and 800 000 edges), it is necessary to accelerate the training speed of the dataset. We simulate the strategy of acceleration here. First, the dataset is divided into subsets, where the node pairs in each of subsets have the same *embeddedness* (we do not use the subset merge strategy suggested in Section 3.4.1 for convenience).

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

TABLE 1. List of notations and abbreviations.

Notation/abbreviation	Description	Example in Fig. 2
$sign(A, B)$	The attitude of user A towards user B	+1
$out^+(A)$	The set of nodes that A has positive attitude towards	$\{B, G\}$
$out^-(A)$	The set of nodes that A has negative attitude towards	$\{E\}$
$out(A)$	The out-degree set of A, i.e., $out(A) = out^+(A) \cup out^-(A)$	$\{B, E, G\}$
$in^+(A)$	The set of nodes that have positive attitude towards A	$\{F\}$
$in^-(A)$	The set of nodes that have negative attitude towards A	$\{C\}$
$in(A)$	The in-degree set of A, i.e., $in(A) = in^+(A) \cup in^-(A)$	$\{C, F\}$
embeddedness [12] of A and B	The size of common neighborhood of a node pair in an undirected sense	2
$intersectedness^+(A, B)$	The intersectino of $out^+(A)$ and $out^+(B)$, i.e., $intersectedness^+(A, B) = out^+(A) \cap out^+(B)$	{G}
$intersectedness^-(A, B)$	The intersectino of $out^-(A)$ and $out^-(B)$, i.e., $intersectedness^-(A, B) = out^-(A) \cap out^-(B)$	{E}
$intersectedness(A, B)$	The intersection of A's out-degree and B's out-degree in directed sense, i.e., $intersectedness(A, B) = out(A) \cap out(B)$	{E, G}
$pos(D)$	Number of positive edges in dataset D	5
$neg(D)$	Number of negative edges in dataset D	3
$balance(D)$	The balance of dataset D, defined as $\max\left(\frac{pos(D)}{neg(D)}, \frac{neg(D)}{pos(D)}\right)$	5/3
size of (D)	The number of links in dataset D	8

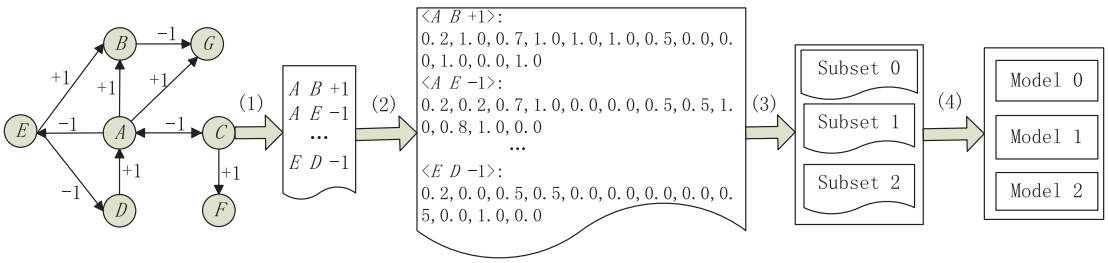

FIGURE 3. An example of employing the PLSP algorithm.

Subset 0: $\langle CA - 1\rangle$, $\langle CF + 1\rangle$,
Subset 1: $\langle AG + 1\rangle$, $\langle BG - 1\rangle$, $\langle DA + 1\rangle$, $\langle EB + 1\rangle$, $\langle ED - 1\rangle$,
Subset 2: $\langle AB + 1\rangle$, $\langle AE - 1\rangle$,

where the *embeddedness* of samples in Subset i is i.

Secondly, the feature selection algorithm proposed in Section 3.4.2 is applied here to further accelerate the training speed. After the selection, the features for each subset are presented as follows:

Subset 0: $GF_1, GF_2, GF_3, GF_4, GF_5, GF_6, GF_7, GF_8$,
Subset 1: $GF_1, GF_2, GF_3, GF_4, GF_5, GF_6, GF_7, GF_8, LF_1, LF_2, LF_3, LF_4$,
Subset 2: LF_1, LF_2, LF_3, LF_4.

(4) Train models on each subset. With the features selected in (3), every subset can train a model for link sign prediction. Specifically, in Fig. 3, Subset 0 trains Model 0, Subset 1 trains Model 1 and Subset 2 trains Model 2. Then given a node pair to be predicted, we only need compute the *embeddedness* of the pair and use the model trained by the subset with the same *embeddedness* for prediction. For example, if we want to predict the sign between C and G, Model 1 is chosen for the prediction as the *embeddedness* of node pair ⟨C G⟩ is 1.

3.3. Feature extraction

As suggested by Leskovec *et al.* [12] and Chiang *et al.* [15], the major improvement of predictive accuracy is achieved by extracting suitable features which can uncover the underlying principles of generating online signed social networks. We take into account both global and local features to uncover the principles. On the one hand, local features can capture the personalized taste and therefore be able to predict the attitude

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

more accurately. On the other hand, global features such as node degree can reflect user's overall attitude and can be used when the available information of local features is scarce.

We now define eight global features GF_i $(1 \leq i \leq 8)$ and four local features LF_i $(1 \leq i \leq 4)$ in terms of each link. In the definitions, the link from node A to B is taken as an example, and the goal is to predict $sign(A, B)$.

3.3.1. *Global features*

Inspired by Leskovec *et al.* [12], our global features are derived from the status theory [19] (GF_1 and GF_2), the feedback transmission theory [20] (GF_7 and GF_8) and several *a priori* estimates from existing links (GF_3, GF_4, GF_5 and GF_6).

(I) GF_1: The status value of A. the computation of status value is based on the status theory [19]. The theory basically says that for any node A and B, if $sign(A, B) = 1$, then node B has a higher status than A. As a result, for a link whose sign is to be predicted, we can simply compare the status values between the two users:

$$sign(A, B) = \begin{cases} +1 & A's\ status < B's\ status, \\ -1 & otherwise. \end{cases} \quad (1)$$

For each isolated node, the initial status of them is defined as zero, i.e. $GF_1 = 0$. When the isolated node forms a positive link towards or receives a negative link from another node, the status value decreases by 1, i.e. $GF_1 = GF_1 - 1$. Conversely, when the node forms a negative link towards or receives a positive link from another node, the status value increases by 1, $GF_1 = GF_1 + 1$. Taking the above cases together, the status value GF_1 of A is calculated as:

$$GF_1 = |out^-(A)| + |in^+(A)| - |out^+(A)| - |in^-(A)|.$$

(II) GF_2: The status value of B. Similar to GF_1, the status value of B, i.e. the other end of the link, is calculated as:

$$GF_2 = |out^-(B)| + |in^+(B)| - |out^+(B)| - |in^-(B)|.$$

Because both GF_1 and GF_2 are available, the sign of a link can be simply decided by Equation (1), which has been evaluated by Leskovec *et al.* [12]. In this work, GF_1 and GF_2 are considered as two separate features to capture more fine-grained information.

(III) GF_3: The *a priori* probability of user A forming positive links towards others. The intuition behind this feature is that some users tend to form positive links generously. Consider the situation when user A reads some content from an unknown user B in Epinions.com. In this situation, the sign can be somehow guessed from the history links that A forms. Namely, if the probability of forming positive links is very high, we may guess that A will probably form a positive link towards B. Similar to Equation (1), Equation (2) shows the prediction of $sign(A, B)$ by feature GF_3.

$$sign(A, B) = \begin{cases} +1 & GF_3 \geq 50\%, \\ -1 & otherwise. \end{cases} \quad (2)$$

In our definition, GF_3 is calculated as follows:

$$GF_3 = \begin{cases} \dfrac{|out^+(A)|}{|out(A)|} & |out(A)| > 0, \\ 50\% & otherwise. \end{cases} \quad (3)$$

Note that in Equation (3), when a node has no out-links, the probability of the node forming a positive link is 50%. This special treatment is also applied to all of the following features.

(IV) GF_4: The importance of GF_3. Consider the following two situations: (1) A has one positive out-link and nine negative out-links. (2) A has 1000 positive out-links and 9000 negative out-links. Intuitively, as the values of GF_3 are the same in the two situations, the more out-links A has, the more obvious GF_3 has the influence on the prediction. To capture this difference, it is necessary to consider GF_4, and the calculation of GF_4 is described as follows:

$$GF_4 = |out(A)|.$$

(V) GF_5: The *a priori* probability of user B receiving positive links from others. Similar to GF_3, this feature can be used as

$$sign(A, B) = \begin{cases} +1 & GF_5 \geq 50\%, \\ -1 & otherwise. \end{cases} \quad (4)$$

GF_5 is calculated as follows:

$$GF_5 = \begin{cases} \dfrac{|in^+(B)|}{|in(B)|} & |in(B)| > 0, \\ 50\% & otherwise. \end{cases} \quad (5)$$

(VI) GF_6: The importance of GF_5. The effect of this feature is similar to GF_4. Here is the calculation of GF_6:

$$GF_6 = |in(B)|.$$

(VII) GF_7: The feedback to user A. This feature is derived from the feedback transmission theory: the more positive sentiments one person receives, the more he or she will give positive sentiments to others [20]. Based on this theory, we can expect that user A would probably give positive reviews if he or she receives a lot of positive reviews. Therefore, we use the probability that user A

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

receives positive reviews as a feature in our method. The calculation of this feature (GF_7) is shown as follows:

$$GF_7 = \begin{cases} \dfrac{|in^+(A)|}{|in(A)|} & |in(A)| > 0, \\ 50\% & otherwise. \end{cases}$$

(VIII) GF_8: The feedback from user B. Similar to GF_7, this feature is derived from another aspect of the feedback transmission theory: the more positive sentiments user B gives to others, the more B would receive positive sentiments from others [20]. Here is the calculation of GF_8:

$$GF_8 = \begin{cases} \dfrac{|out^+(B)|}{|out(B)|} & |out(B)| > 0, \\ 50\% & otherwise. \end{cases}$$

In contrast to Leskovec *et al.* [12], who consider the effect of Equations (2) and (4) only in the condition that the *embeddedness* of A and B is bigger than 25, we extend these features to all node pairs by Equations (3) and (5), respectively.

3.3.2. *Local features*

Our local features are derived from the neighborhood information (LF_1 and LF_2) and the balance theory [17] (LF_3 and LF_4).

(I) LF_1: The proportion of common neighbors that A and B have same attitude towards, in all common neighbors. Common neighbors have been considered in link prediction problem [8, 9], and we adapt it for link sign prediction by taking the negative links into account. The intuition behind this feature is that if the attitude of A and B towards the nodes in *intersectedness* (A, B) are similar, then A and B may have positive attitude towards each other since they have similar taste. The calculation of LF_1 is as follows:

$$LF_1 = \begin{cases} \dfrac{|out^+(A) \cap out^+(B)| + |out^-(A) \cap out^-(B)|}{|out(A) \cap out(B)|} & |out(A) \cap out(B)| > 0, \\ 50\% & otherwise. \end{cases}$$

(II) LF_2: The importance of LF_1. Consider the following two situations: (1) A and B have 100 out-links in total while only five out-links go to common neighbors. (2) A and B have five out-links in total and all of them go to common neighbors. Intuitively, the influence of LF_1 is greater in latter situation. To capture this difference, it is necessary to consider LF_2, and the calculation of LF_2 is described as follows:

$$LF_2 = \begin{cases} \dfrac{|out(A) \cap out(B)|}{|out(A) \cup out(B)|} & |out(A) \cup out(B)| > 0, \\ 50\% & otherwise. \end{cases}$$

If $|out(A) \cup out(B)| > 0$, LF_2 is calculated by using a fraction whose numerator is the number of people both A and B having comments on and whose denominator is the total number of people either A or B having comments on.

(III) LF_3: The attitude of A's neighbors towards B. This feature is derived from the theory of structural balance: the friend of your friend could be your friend [19]. Intuitively, A thinks high of all the nodes and all the nodes think high of B, and therefore A probably also thinks high of B. Here is the calculation of LF_3:

$$LF_3 = \begin{cases} \dfrac{|out^+(A) \cap in^+(B)|}{|out(A) \cap in(B)|} & |out(A) \cap in(B)| > 0, \\ 50\% & otherwise. \end{cases}$$

As we can see, if $|out(A) \cap in(B)| > 0$, LF_3 is calculated by using a fraction whose numerator is the number of neighbors that follows the 'the friend of your friend could be your friend' principal and whose denominator is the number of A's out-neighbors($out(A)$) that also have attitude towards B.

(IV) LF_4: The importance of LF_3. The effect of this feature is similar to LF_2. Here is the calculation of LF_4:

$$LF_4 = \begin{cases} \dfrac{|out(A) \cap in(B)|}{|out(A)|} & |out(A)| > 0, \\ 50\% & otherwise. \end{cases}$$

If A makes comments on others, LF_4 calculated by using the proportion of comments A has made on B.

3.3.3. *Comparison of global and local features*

As we can see, GF_i $(1 \leq i \leq 8)$ can be calculated independently of the connecting nodes, and the resulting values are unique for each node. On the other hand, the local features are calculated based on node pairs. For example, when predicting $sign(A, C)$ and $sign(A, B)$, the values of local features may be different.

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

3.4. Parallel training

We now present our parallel training strategies. The dataset is first divided into subsets based on the *embeddedness* of each node pair, and the divided subsets may need to be combined with the other subsets when they are not balanced or their size does not meet the requirements (see Section 3.4.1). In order to speed up the training of each subset, the adapted information gain is used for feature selection, and some subsets are trained with only eight global features or four local features (see Section 3.4.2).

3.4.1. Dividing dataset

Generally, there are two classes of methods to make model training parallel. The first class focuses on making the classifier parallel. For example, Graf *et al.* [27] suggest a cascade SVM and their main idea is to eliminate the non-supporting vectors early from optimization. However, the second class tries to make the dataset independent. For example, Yu *et al.* [28] use the strategies of bagging and boosting to achieve parallel training. In detail, the samples of each subset are chosen randomly by using the bagging strategy or based on the difficulty to be classified by using the boosting strategy.

In this paper, the dataset division method is applied. Our division is based on the *embeddedness* of node pairs. In the PLSP method, the dataset is divided into several subsets each of which only consists of samples with the same *embeddedness*. As a result, for a given link to be predicted, we first compute the *embeddedness* of this link, and the sign is then decided by the local model trained for this *embeddedness*. In contrast to Yu *et al.* [28] whose method needs to conduct the majority vote based on the predicted results of all local models, we only need to choose one model to predict the result.

One problem in dataset division is that the subset may be very small, causing the small sample size problem [29]. In addition to dataset size, there is another issue that may affect predictive accuracy: dataset balance. To be more specific, the dataset balance issue means that if one of the classes (positive or negative) is overwhelming in the dataset, the trained model may be over-fitting. To overcome the small sample size problem and the dataset imbalance problem, we merge the subset with others if its size is too small or if it is very imbalanced. In particular, let us assume the *embeddedness* of the subset which needs to be merged is i, and then it is merged with the subset where the *embeddedness* of all samples is $(i-1)$.

The dividing algorithm, called DividingDataset, is shown in Fig. 4. The full dataset D is divided into $D_0, D_1, \ldots, D_n$, where the *embeddedness* of samples in D_i is i, and the time complexity of step 1 is $O(\text{size of } (D))$. If the size of subset D_i is smaller than the threshold M or *balance* (D_i) is greater than $(balance(D)+o)$, where o is the tolerance of the imbalance and we set it as 2 according to the datasets used in our experiments, D_i is merged with D_{i-1}, and the merging procedure is iterative. The time complexity of step 2–8 is $O(n)$. So, the overall time complexity of the DividingDataset algorithm is $O(\text{size of } (D)) + O(n) = O(\text{size of } (D))$, where $n \leq size\ of\ (D)$.

Algorithm DividingDataset(D,M)

Input: Dataset D
Minimum size of the sub-dataset M
Output: Array S storing the divided subsets
1: divide D into $D_0, D_1 \cdots D_n$
2: **for** i **in** n **to** 1
3: **if** size of $(D_i) <$ M || *balance* $(D_i) >$ *balance* (D) +2
4: combine D_i with D_{i-1} as D_{i-1}
5: **continue**
6: **end if**
7: $S \leftarrow D_i$
8: **end for**
9: $S \leftarrow D_0$
10: **return** S

FIGURE 4. The DividingDataset() Algorithm for dividing dataset into subsets.

3.4.2. Reducing the number of features

Despite the speedup from dataset division, the number of samples with zero *embeddedness* in Epinions and Slashdot are still 167 154 and 263 014, respectively. Training on these subsets is still time-consuming. Fortunately, as the values of local features in the subset (the *embeddedness* of all samples in it is zero) are all the same, it can be trained with only eight global features.

Actually, this feature reduction strategy can also be applied to other subsets. Namely, for the subsets that have sufficient local information, they can be trained with only local features. There are also subsets that need both global features and local features. Now the problem is how to select the subset of features.

In this paper, information gain [30] is applied to determine the influence of features. Information gain is usually used as a method for feature selection in decision tree [31]. The larger the information gain is, the better the feature can contribute to the classification accuracy. Here is the calculation of information gain of feature F:

$$gain(F) = E(All) - E(F),$$

where $E(All)$ is the information entropy [32] of the full dataset, and $E(F)$ is the information entropy of the full dataset after being divided by F.

However, the influence of eight global features and that of local features cannot be computed directly by information gain which mainly measures the influence of a single feature. Therefore, based on $gain(F)$, we propose the average information gain (AIG) to measure the overall influence of a group of features, which is calculated as

$$AIG(F_1, F_2, \ldots, F_n) = \frac{\sum_{i=1}^{n} gain(F_i)}{n},$$

where $F_i (1 \leq i \leq n)$ represents the ith feature.

Based on AIG, the rules for feature selection are presented in logic order as follows:

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

Algorithm SelectingFeatures(S, α, β)

Input: Subset array S
Parameter α, β to weight AIG
Output: each subset in S with part of features
1: **for** S_i **in** S
2: **if** *embeddedness* of samples in $S_i = 0$
3: select four global features for S_i by rule 1
4: **else if** α*$AIG(GF_1, GF_2, GF_3, GF_4, GF_5, GF_6, GF_7, GF_8) < \beta$*$AIG(LF_1, LF_2, LF_3, LF_4)$
5: select four local features for S_i by rule 2
6: **else**
7: keep the whole eight features for S_i by rule 3
8: **end if**
9: **end for**
10: **return** S

FIGURE 5. The SelectingFeatures () Algorithm for selecting features from each subset.

Rule 1. If *embeddedness* of all samples in the subset is zero, eight global features are chosen.

Rule 2. If four local features are more important than global features, only four local features are chosen. The importance is measured by the AIG of features. Additionally, the importance can be weighted because the global features and local features should not be taken equally in different online social networks. For example, in Wikipedia.com, people vote others mainly depending on the celebrity of them, which can be considered as global features. In this situation, global features are more important than local ones. As to Epinions.com, the decision of users can be easily affected by their friends, indicating that local features are more important.

Rule 3. If four local features are less important than global features, the whole 12 features are kept. In this rule, we do not only take eight global features, because this situation often happens in the subsets where the *embeddedness* of the samples is small. We believe that the scarce local information is too valuable to be deserted.

The full feature selection algorithm, called SelectingFeatures, is presented in Fig. 5. Note that the parameters α and β are used to weight the AIG of global features and local features, respectively. With this algorithm, many models for the subsets can be trained by only a subset of features (global features or local features). In each loop, calculating the information gain of each feature (i.e. step 4) is the most time-consuming procedure. As the procedure of information gain needs to sort each subset, the total time complexity of information gain is $O(n_i \lg n_i)$, where n_i is the size of (S_i). Then, the overall time complexity of the SelectingFeatures algorithm is $O(n \lg n)$, where n is the size of the whole dataset D.

4. EXPERIMENTAL EVALUATION

In this section, we first report the results of the link sign prediction experiments in a single-threaded environment. We test three different classifiers (logistic regression classifier, C4.5 decision tree and naïve bayes classifier), and then present the results of the PLSP approach with the best classifier in a single-threaded environment. The experiments are conducted on three real datasets, namely, Epinions, Slashdot and Wikipedia (http://snap.stanford.edu). The basic statistics of the datasets is shown in Table 2. In all following experiments, the average accuracy is reported and 10-fold cross-validation is employed to estimate coefficients of classifiers.

TABLE 2. Dataset statistics.

	Epinions	Slashdot	Wikipedia
Nodes	131,828	82,144	6,988
Edges	841,372	549,202	103,507
Positive edges (%)	85.3	77.4	79.2
Negative edges (%)	14.7	22.6	20.8

TABLE 3. Training results of Epinions.

	C4.5	Logistic Regression	Naïve Bayes
Predicting accuracy (%)	97.0	96.7	91.8
ROC area (%)	97.3	98.5	95.6
F-Measure (class: +1) (%)	98.3	98.0	95.0
F-Measure (class: −1) (%)	90.1	88.0	76.3
Training time (unit: second)	7620	1400	340

TABLE 4. Training results of Slashdot.

	C4.5	Logistic Regression	Naïve Bayes
Predicting accuracy (%)	90.8	90.6	79.5
ROC area (%)	93.9	95.4	90.1
F-Measure (class: +1) (%)	94.1	94.0	85.2
F-Measure (class: +1) (%)	79.1	78.0	66.4
Training time (unit: second)	5100	820	250

4.1. Results in a single-threaded environment

First, we use the 12 features suggested in Section 3.3 as the input of classifiers and present the prediction results of the classifiers (logistic regression classifier, C4.5 decision tree and naïve bayes classifier) in Tables 3–5, respectively. From the tables, we can see that the predictive accuracy of C4.5 and logistic regression is higher than that of naïve bayes in all datasets. As to logistic regression classifier and C4.5 decision tree, the performance is competitive while logistic regression requires shorter training time. In the following experiments, we choose the logistic regression classifier, which learns a model

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

TABLE 5. Training results of Wikipedia.

	C4.5	Logistic Regression	Naïve Bayes
Predicting accuracy (%)	89.2	89.1	86.1
ROC area (%)	91.2	93.6	89.9
F-Measure (class: +1) (%)	93.3	93.2	91.0
F-Measure (class: +1) (%)	73.6	72.3	69.7
Training time (unit: second)	500	150	50

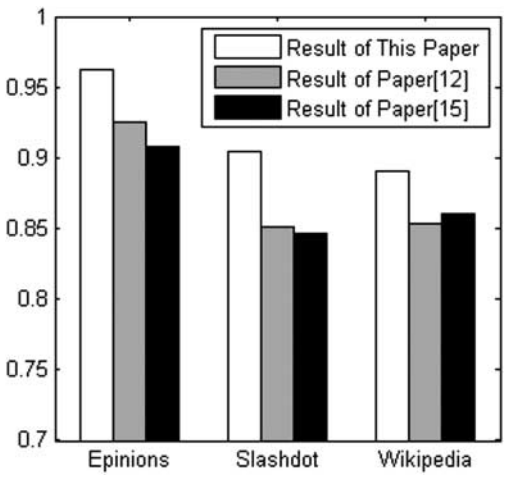

FIGURE 6. The comparison of predictive accuracy.

in the form:

$$P(+1|f) = \frac{1}{1 + e^{-(w_0 + \sum_{i=1}^{n} w_i * f_i)}},$$

where f is a vector of features $(f_1, f_2, \ldots, f_n)$ and $w_0, w_1, \ldots, w_n$ are the coefficients to be estimated based on the training data.

Next, in Fig. 6, we compare the predictive accuracy of our method with that of Leskovec *et al.* [12] and Chiang *et al.* [15], both of whom employ logistic regression as the classifier. We can observe that our predictive accuracy on all three datasets is the highest, indicating that our features are better than theirs in this problem setting. The reasons for this improvement are 2-fold. First, we use eight global features that are based on the well-established status theory (GF_1 and GF_2) and feedback transmission theory (GF_7 and GF_8) as well as the heuristics from out-degree (GF_3 and GF_4) and in-degree (GF_5 and GF_6) information. Secondly, our four local features make use of the transitivity of attitude by considering the common neighborhood (LF_1 and LF_2) and adapt the well-established balance theory (LF_3 and LF_4). On the other hand, Chiang *et al.* [15] ignore the effect of global information, and Leskovec *et al.* [12] only consider the simple degree and triad features. In addition, the PLSP method uses fewer features (the number of features in [12, 15] are 23 and 16, respectively). As to AUC measurement, similar results are observed. For example, The AUC results of Epinions, Slashdot and Wikipedia are 98.5, 95.4 and 93.6%, respectively. The figure is omitted for brevity.

TABLE 6. Generalization of models.

	Data		
Model	Epinions (%)	Slashdot (%)	Wikipedia (%)
Epinions	96.7	89.6	87.2
Slashdot	96.4	90.6	89.1
Wikipedia	96.2	89.6	89.1

The training time on datasets Epinions, Slashdot, and Wikipedia are 1400, 820 and 150 s, respectively. We will further discuss this in Section 4.3.

Next, the generalization of our prediction models is checked: we use the model trained from one dataset and test the model on the other datasets. The results are shown in Table 6. As we can see, the predictive accuracy of each dataset is almost the same no matter which model is used. This result indicates that the extracted features and the trained models can uncover the underlying principles of generating signed social networks [12].

Finally, we compare our features with some basic features that are related to out-degree and in-degree. Again, the link from node A to B is taken as an example, and the features we compared with are: $|out^+(A)|$, $|out^-(A)|$, $|in^+(A)|$, $|in^-(A)|$, $|out^+(B)|$, $|out^-(B)|$, $|in^+(B)|$, $|in^-(B)|$, $|intersectedness^+(A, B)|$ and $|intersectedness^-(A, B)|$. We denote these 10 features by OF_i $(1 \leq i \leq 10)$, respectively. Among our features and OF_i $(1 \leq i \leq 10)$, there exist many features providing the same information for prediction:

$$OF_1 = GF_3 * GF_4; \quad OF_2 = GF_4 - GF_3 * GF_4;$$
$$OF_3 = \frac{(GF_1 - GF_4 + 2GF_4 * GF_3) * GF_7}{2GF_7 - 1};$$
$$OF_4 = \frac{(GF_1 - GF_4 + 2GF_4 * GF_3) * (1 - GF_7)}{2GF_7 - 1};$$
$$OF_5 = \frac{(GF_2 + GF_6 - 2GF_5 * GF_6) * GF_8}{1 - 2GF_8};$$
$$OF_6 = \frac{(GF_2 + GF_6 - 2GF_5 * GF_6) * (1 - GF_8)}{1 - 2GF_8};$$
$$OF_7 = GF_5 * GF_6; OF_8 = GF_6 - GF_5 * GF_6.$$

Thus, we only need to achieve five conditions to prove that our features are better than these basic features: (1) GF_3 and GF_4 are better than OF_1 and OF_2; (2) GF_1, GF_3, GF_4 and GF_7 are better than OF_3 and OF_4; (3) GF_2, GF_5, GF_6 and GF_8 are better than OF_5 and OF_6; (4) GF_5 and GF_6 are better than OF_7 and OF_8; (5) all our features are better than OF_9 and OF_{10}. A feature selection algorithm, called Relief [33], is applied to evaluate this. In the Relief algorithm, 10 000 samples are chosen from each dataset

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

TABLE 7. The importance of every feature in every dataset using the Relief algorithm.

Epinions	$GF_3, GF_5, LF_4, GF_7, GF_8, GF_4, OF_1, LF_2, LF_3, OF_5, LF_1, GF_6, OF_7, GF_1, OF_3, OF_8, GF_2, OF_4, OF_9, OF_2, OF_6, OF_{10}$
Slashdot	$GF_3, GF_5, GF_7, GF_4, OF_1, OF_5, GF_8, OF_2, LF_4, GF_6, OF_8, OF_7, GF_2, GF_1, OF_4, LF_3, LF_1, OF_3, LF_2, OF_6, OF_9, OF_{10}$
Wikipedia	$GF_5, GF_3, GF_8, LF_3, GF_6, OF_8, OF_7, GF_7, GF_4, GF_1, OF_3, LF_2, OF_1, OF_2, LF_4, GF_2, OF_4, OF_5, LF_2, OF_6, OF_9, OF_{10}$

TABLE 8. The status of datasets after applying parallel method.

	Global features (%)	Local features (%)	All features (%)	Number of subsets
Epinions	20.4	66.6	13.0	4
Slashdot	47.9	13.8	39.3	9
Wikipedia	8.3	0	91.7	6

and the number of nearest neighbors used to estimate attribute relevance is set as 10 (these two parameters are user-defined [34] and the setting is also applied to the following feature selection experiments by the Relief algorithm). Table 7 shows the importance of every feature in each dataset in descending order. As we can see, all the five conditions are met in every dataset, which means our features are better than the basic features in these datasets under the problem setting.

4.2. Results in a parallel environment

We now study the link sign prediction problem in a parallel environment. In this experiment, the minimum size M (in the DividingDataset algorithm) of each subset is set as 10 000. In addition, (α, β) (in the SelectingFeatures algorithm) is set to (0.35, 0.65) on Epinions and Slashdot (which prefer local features), and (0.65, 0.35) on Wikipedia (which prefers global features), respectively.

Table 8 shows the number of subsets and the proportion of samples trained by global features, local features or all features. In the table, although most data samples of Wikipedia are trained with all features, the size of each subset is no more than 20 000 (about one-fifth of the original dataset). In the other two datasets (Epinions and Slashdot), many data samples can be trained with only global features or local features.

The statistics of training time and predictive accuracy of all subsets are presented by five-number summaries (i.e. minimum, low quartile (Q1), median, upper quartile (Q3) and maximum) in Tables 9 and 10, respectively. The mean value and standard deviation are also presented. As we can see in Table 9, the running time of most subsets is no more than 60 s. Though the longest running time is 450 s (the *embeddedness* of samples between 3 and 3395 in Epinions), it still achieves more than 3× speedup compared with the full dataset. In Table 10, the predictive accuracy of all subsets is comparable to that of the serial training and it is discussed from a statistical view in Section 4.3. Additionally, the minimum predictive accuracy of each subset is still higher than that of Leskovec *et al.* [12] and Chiang *et al.* [15].

To evaluate the performance of our feature selection algorithm, the Relief feature selection algorithm [33] is considered for the comparison. From Table 11, we can see that

TABLE 9. The statistics of training time (unit: second).

	Minimum	Q1	Median	Q3	Maximum	Mean	Standard deviation
Epinions	55	74	140	262	450	196	156
Slashdot	10	20	30	60	300	64	86
Wikipedia	10	11	15	15	40	18	10

TABLE 10. The statistics of predictive accuracy.

	Minimum (%)	Q1 (%)	Median (%)	Q3 (%)	Maximum (%)	Mean (%)	Standard deviation (%)
Epinions	95.7	96.0	96.3	96.4	96.4	96.2	0.2
Slashdot	88.3	88.8	89.0	89.1	92.6	89.6	1.4
Wikipedia	87.6	88.1	88.7	89.6	90.6	88.9	1.0

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

TABLE 11. Accuracy comparison of different feature selection algorithm.

	Algorithm	Mean (%)	Standard deviation (%)	Average time of feature selection (unit: second)
Epinions	SelectingFeatures	96.3	0.1	56
	Relief	96.2	0.3	2760
Slashdot	SelectingFeatures	92.1	0.6	4
	Relief	91.4	0.1	480
Wikipedia	SelectingFeatures	90.6	0	1
	Relief	90.6	0	5

TABLE 12. The statistics of speedup ratio of the subsets.

	Minimum	Q1	Median	Q3	Maximum	Mean	Standard deviation
Epinions	4	7	14	22	26	15	9
Slashdot	3	14	27	41	82	35	27
Wikipedia	4	10	10	14	15	11	4

TABLE 13. The comparison of predictive accuracy of parallel training and serial training.

	Parallel training		
	Mean (%)	Standard variance (%)	Serial training (%)
Epinions	96.2	0.2	96.7
Slashdot	89.6	1.4	90.6
Wikipedia	88.9	1.0	89.1

in every dataset, our feature selection algorithm performs as well as the Relief algorithm does in this condition. However, the average feature selection time of the Relief algorithm is five times more than that of the Selecting Features algorithm we proposed. That is, our feature selection algorithm can perform better in handling the huge amount of data than the Relief algorithm.

4.3. Comparison of the PLSP approach and non-parallel method

We now compare the predictive accuracy and training time of the serial method and PLSP approach. For training time, we define the speedup ratio of subset i as $t(\textit{whole dataset})/t(\textit{subset } i)$, where $t(\textit{whole dataset})$ is the training time on the whole dataset in a single-threaded environment, and $t(\textit{subset } i)$ is the training time of subset i in a parallel environment. Table 12 presents the five-number summaries of speedup ratio of all subsets. The mean value and the standard deviation are also presented. We can observe that more than 10× speedup is achieved in most of the subsets.

For the comparison of predictive accuracy, the mean and variance of predictive accuracy of parallel training and serial training are shown in Table 13. As we can see in the table, the predictive accuracy of our parallel method compares favorably with that of the serial method. Taking the results of Tables 12 and 13 together, we find that the PLSP method performs well in reducing the running time while preserving high predictive accuracy at the same time.

5. CONCLUSIONS

In this paper, we regard the link sign prediction problem of social networks as a classification problem, and suggest twelve features (eight global features and four local features) for training. Experiments show that the predictive accuracy of these features is higher than that of those features proposed by the state-of-the-art method on three real online social networks. We then propose two algorithms to accelerate the training speed in huge datasets, by dividing the dataset into several subsets based on the neighborhood information of node pairs, and by selecting a subset of features for each subset based on the information gain of features. The experimental evaluations show that the training speed of our method is significantly faster than that of the traditional serial training, while little predictive accuracy is lost at the same time.

FUNDING

This work was supported by grants from the National Natural Science Foundation of China (project 60673186, project

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

60571048, project 60873264 and project 60971088), and the Qing Lan Project.

REFERENCES

[1] Wilson, C., Boe, B., Sala, A., Puttaswamy, K.P.N. and Zhao, B.Y. (2009) User Interactions in Social Networks and their Implications. *Proc. 2009 EuroSys Conf., Nuremberg*, Germany, April 1–3, pp. 205–218. ACM, New York.

[2] Yao, Y., Zhou, J.F., Han, L.X., Xu, F. and Lu, J. (2011) Comparing Linkage Graph and Activity Graph of Online Social Networks. *Proc. SocInfo 2011*, Singapore, October 6–8, pp. 84–97. Springer, Berlin.

[3] Mislove, A., Post, A., Druschel, P. and Gummadi, P.K. (2008) Ostra: Leveraging Trust to Thwart Unwanted Communication. *Proc. NSDI 2008*, San Francisco, CA, USA, April 16–18, pp. 15–30. USENIX Association Berkeley, CA, USA.

[4] Yu, H.F., Kaminsky, M., Gibbons, P.B. and Flaxman, A. (2006) SybilGuard: Defending Against Sybil Attacks Via Social Networks. *Proc. ACM SIGCOMM 2006*, Pisa, Italy, September 11–15, pp. 267–278. ACM, New York.

[5] Kang, U., Papadimitriou, S., Sun, J.M. and Tong, H.H. (2011) Centralities in Large Networks: Algorithms and Observations. *Proc. SDM 2011*, Mesa, AZ, USA, April 28–30, pp. 119–130. SIAM, Philadelphia, PA.

[6] Kang, U., Chau, D.H. and Faloutsos, C. (2011) Mining Large Graphs: Algorithms, Inference, and Discoveries. *Proc. ICDE 2011*, Hannover, Germany, April 11–16, pp. 243–254. IEEE Press, Piscataway, NJ, USA.

[7] Papadimitriou, S. and Sun, J.M. (2008) DisCo: Distributed Co-clustering with Map-Reduce: A Case Study towards Petabyte-Scale End-to-End Mining. *Proc. ICDM 2008*, Pisa, Italy, December 15–19, pp. 512–521. IEEE Press, Piscataway, NJ, USA.

[8] Liben-Nowell, D. and Kleinberg, J.M. (2003) The Link Prediction Problem for Social Networks. *Proc. 2003 ACM CIKM*, New Orleans, LA, USA, November 2–8, pp. 556–559. ACM, New York.

[9] Lichtenwalter, R., Lussier, J.T. and Chawla, N.V. (2010) New Perspectives and Methods in Link Prediction. *Proc. 16th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Washington, DC, USA, July 25–28, pp. 243–252. ACM, New York.

[10] Guha, R.V., Kumar, R., Raghavan, P. and Tomkins, A. (2004) Propagation of Trust and Distrust. *Proc. WWW 2004*, New York, NY, USA, May 17–20, pp. 403–412. ACM, New York.

[11] Brzozowski, M.J., Hogg, T. and Szabó, G. (2008) Friends and Foes: Ideological Social Networking. *Proc. CHI 2008*, Florence, Italy, April 5–10, pp. 817–820. ACM, New York.

[12] Leskovec, J., Huttenlocher, D.P. and Kleinberg, J.M. (2010) Predicting Positive and Negative Links in Online Social Networks. *Proc. WWW 2010*, Raleigh, NC, USA, April 26–30, pp. 641–650. ACM, New York.

[13] Symeonidis, P., Tiakas, E. and Manolopoulos, Y. (2010) Transitive Node Similarity for Link Prediction in Social Networks with Positive and Negative Links. *Proc. RecSys 2010*, Barcelona, Spain, September 26–30, pp. 183–190. ACM, New York.

[14] Kunegis, K., Schmidt, S., Lommatzsch, A., Lerner, J., Luca, E.W.D. and Albayrak, S. (2010) Spectral Analysis of Signed Graphs for Clustering, Prediction and Visualization. *Proc. SDM 2010*, Columbus, OH, USA, April 29–May 1, pp. 559–570. SIAM, Philadelphia, PA.

[15] Chiang, K.Y., Natarajan, N., Tewari, A. and Dhillon, I.S. (2011) Exploiting Longer Cycles for Link Prediction in Signed Networks. *Proc. CIKM 2011*, Glasgow, UK, October 24–28, pp. 1157–1162. ACM, New York.

[16] Leskovec, J., Kleinberg, J.M. and Faloutsos, C. (2005) Graphs Over Time: Densification Laws, Shrinking Diameters and Possible Explanations. *Proc. 11th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Chicago, IL, USA, August 21–24, pp. 177–187. ACM, New York.

[17] Heider, F. (1946) Attitudes and cognitive organization. *J. Psychol.*, **21**, 107–112.

[18] Cartwright, D. and Harary, F. (1956) Structural balance: a generalization of heider's theory. *Psychol. Rev.*, **63**, 277–293.

[19] Leskovec, J., Huttenlocher, D. and Kleinberg, J. (2010) Signed Networks in Social Media. *Proc. CHI 2010*, Atlanta, GA, USA, April 10–15, pp. 1361–1370. ACM, New York.

[20] Fisher, C.D. (1979). Transmission of positive and negative feedback to subordinates: a laboratory investigation. *J. Appl. Psychol.*, **64**, 533–540.

[21] Chakrabarti, D. and Faloutsos, C. (2006) Graph mining: laws, generators, and algorithms. *ACM Comput. Surv.*, **38**, 52–120.

[22] Palla, G., Derényi, I., Farkas, I. and Vicsek, T. (2005) Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, **435**, 814–818.

[23] Bródka, P., Kazienko, P., and Koloszczyk, B. (2012) Predicting Group Evolution in the Social Network. *Proc. SocInfo 2012*, Lausanne, Switzerland, December 5–7, pp. 54–67. Springer, Berlin.

[24] Richter, Y., Yom-Tov, E. and Slonim, N. (2010) Predicting Customer Churn in Mobile Networks through Analysis of Social Groups. *Proc. SDM 2010*, Columbus, OH, USA, April 29–May 1, pp. 732–741. SIAM, Philadelphia, PA.

[25] Davis, D.A., Lichtenwalter, R. and Chawla, N.V. (2011) Multi-relational Link Prediction in Heterogeneous Information Networks. *Proc. ASONAM 2011*, Kaohsiung, Taiwan, July 25–27, pp. 281–288. IEEE Computer Society Press, Washington, DC and Los Angeles, CA.

[26] Kunegis, J., Lommatzsch, A. and Bauckhage, C. (2009) The slashdot Zoo: Mining a Social Network with Negative Edges. *Proc. WWW 2009*, Madrid, Spain, April 20–24, pp. 741–750. ACM, New York.

[27] Graf, H.P., Cosatto, E., Bottou, L., Dourdanovic, L. and Vapnik, V. (2004) Parallel support vector machines: the Cascade SVM. *Adv. Neural Inf. Process. Syst.*, **17**, 521–528.

[28] K7L3N6 (2001) *Parallelizing Boosting and Bagging*. Queen's University, Kingston, ON, Canada.

[29] Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition*. Academic Press, New York.

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013

[30] Han, J.W. and Kamber, M. (2001) *Data Mining: Concepts and Techniques*. Academic Press, London.

[31] Quinlan, J.R. (1986) Induction of decision trees. *Mach. Learn.*, **1**, 81–106.

[32] Shannon, C.E. and Weaver, W. (1949) *The Mathematical Theory of Communication*. The University of Illinois Press, Urbana, IL.

[33] Kira, K. and Rendell, L.A. (1992) The Feature Selection Problem: Traditional Methods and a New Algorithm. *Proc. 10th Nat. Conf. on Artificial Intelligence*, San Jose, CA, July 12–16, pp. 129–134. AAAI Press, Menlo Park, CA.

[34] Robnik-Sikonja, M. and Kononenko, I. (2003) Theoretical and empirical analysis of ReliefF and RReliefF. *Mach. Learn.*, **53**, 23–69.

Downloaded from http://comjnl.oxfordjournals.org/ at Nanjing University on June 28, 2013