Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 60 (2015) 332 – 341

Procedia
Computer Science

19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

Transfer AdaBoost SVM for Link Prediction in Newly Signed Social Networks using Explicit and PNR Features

Anh-Thu Nguyen-Thi*, Phuc Quang Nguyen, Thanh Duc Ngo, Tu-Anh Nguyen-Hoang

University of Information Technology, Vietnam National University HCMC, Quarter 6, Linh Trung Ward, Thu Duc Dist, Ho Chi Minh city 700000, Vietnam

Abstract

In signed social network, the user-generated content and interactions have overtaken the web. Questions of whom and what to trust has become increasingly important. We must have methods which predict the signs of links in the social network to solve this problem. We study signed social networks with positive links (friendship, fan, like, etc) and negative links (opposition, anti-fan, dislike, etc). Specifically, we focus how to effectively predict positive and negative links in newly signed social networks. With SVM model, the small amount of edge sign information in newly signed network is not adequate to train a good classifier. In this paper, we introduce an effective solution to this problem. We present a novel transfer learning framework is called Transfer AdaBoost with SVM (TAS) which extends boosting-based learning algorithms and incorporates properly designed RBFSVM (SVM with the RBF kernel) component classifiers. With our framework, we use explicit topological features and Positive Negative Ratio (PNR) features which are based on decision-making theory. Experimental results on three networks (Epinions, Slashdot and Wiki) demonstrate our method that can improve the prediction accuracy by 40% over baseline methods. Additionally, our method has faster performance time.

Keywords: Link Prediction; Signed Social Network; AdaBoost Algorithm

1. Introduction

As well as the development of online social network, user-generated content is created and consumed at impressive rates. With so much user interactions and contents are created, the question of whom and what to trust has become an increasingly important challenge. Fortunately, online social networks have allowed people to indicate whom they trust (*positive links*) and distrust (*negative links*). However, this does not solve the problem, we need a signed link prediction system which predicts the signs of links in online social network. Then, we can algorithmically use that positive and negative information to make suggestions to other users about whom they in turn should trust and help a user make decisions, sort and filter information, receive recommendations.

* Corresponding author. Tel.: +84-090-737-9067
E-mail address: thunta@uit.edu.vn, nguyenthianhthu6789@gmail.com

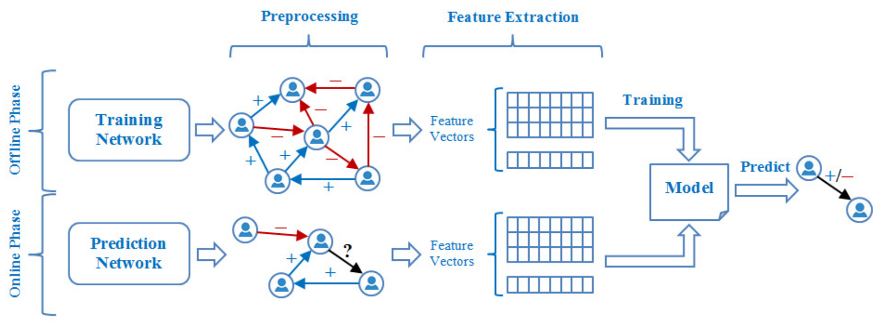

Fig. 1. Framework of a signed link prediction system.

Examples include Epinions[1] whose users can express trust or distrust of others[15], Slashdot[2] whose participants can declare others to be either friends or foes[10] and Wiki[3] whose users can vote for or against the promotion of others to adminship[2]. On Epinions, the trust and distrust information is used to determine the reviews shown, using an undisclosed algorithm. On Slashdot, the posts of users tagged as foes are given a lower score, and may thus be hidden. On Wiki, the voting information is used to automatically search for likely future administrators.

A signed link prediction system works through two phases: the offline and the online phases[19]. The overview of the whole framework is given in the Fig. 1. The purpose of offline phase is to learn a prediction model from the training data with three steps: preprocessing, feature extraction and training. The online phase begins with the step of preprocessing and feature extraction similar to the offline phase. Then, the prediction model (trained in the offline phase) is used to assign an edge to positive or negative.

Previous research of signed social networks[14] has shown that the prediction model makes a very strong assumption on the input network: *the signs of all links except the one to be predicted are known in advance*. Thus, we study the edge sign prediction problem with a more realistic setting. Given a newly signed social network, the paucity of available signs makes it difficult to train a good classifier to predict unknown link signs. To solve this problem, we consider leveraging another more mature signed social network, which has the abundant edge sign information. This approach is known as *transfer learning*[16,17].

In this paper, we present a novel transfer learning framework called *TAS* which extends boosting-based learning algorithms and incorporates properly designed *RBFSVM* (SVM with the RBF kernel) component classifiers. With our framework, we use *explicit topological features*[20]. Besides, we propose to use *PNR feature*[19] which is based on the strong theory of decision-making. PNR is a generalizable feature and outperforms most state-of-the-art features in three criteria: accuracy, generalization and speed. Our experimental results on three real signed social networks demonstrate that our method can improve the prediction accuracy and reduces time to extract features, train and test data over baseline methods.

The rest of the paper is organized as follows. We discuss related work in Section 2. Section 3 presents our proposed method which is called TAS. Additionally, we present explicit features and PNR features which are used in our approach in this section. Section 4 shows the experimental results with discussions. Finally, we conclude the paper in Section 5.

2. Related Work

For the edge sign prediction problem, existing studies can be categorized into two major approaches: a matrix kernel approach[9,10] and a machine learning approach[3,11]. Guha et al[9] proposed their leading work on trust propagation in signed social network. Kunegis et al[10] extended the method of graphical spectrum analysis by using kernels taken

[1] www.epinions.com
[2] slashdot.org
[3] www.wikipedia.org

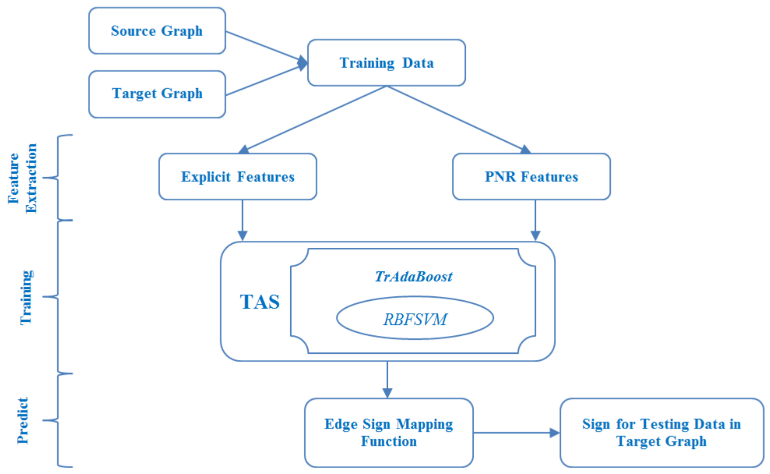

Fig. 2. Our method for signed link prediction in newly signed social networks.

from signed Laplacian matrices of graphs. Leskovec et al[11] proposed *16-dimensional features* corresponding to 16 types of triads in the balance theory and used a logistic regression model to predict the sign of links. Kai-Yang et al[3] tried to improve the quality of feature representation. Instead of using triads, the authors took all cycles of m vertices containing predicted links. The major contribution of their studies is the connections to theories of balance and status in social psychology. However, in their studies, the prediction model makes a very strong assumption on the input network: the signs of all links except the one to be predicted are known in advance, which is not very practical in reality.

Thus, we study the edge sign prediction problem for a newly signed social network whose the edge sign information is very scarce. We try to develop a general framework for transfer learning based on *TrAdaBoost*[4]. We use *RBFSVM* (SVM with the RBF kernel) as component classifier in *TrAdaBoost*. Our method is called *TAS*. The overview of the whole our method is given in the Fig. 2. Our method considers to leverage another more mature signed social network (*Source Graph*) to construct a high-quality classification model for a newly signed social network (*Target Graph*). Thus, we need use generalizable features which can apply to many social networks. In our method, we use *explicit topological features*[20] (node degree, betweenness centrality, triad count and edge embeddedness) which express manifest properties of the edge instances. Instead of using explicit features, we use *PNR features* which are based on decision-making theory.

3. Proposed Method

This section describes our proposed method for edge sign prediction. Firstly, we present our formal definitions for the edge sign prediction problem. Secondly, we propose a TAS which likes learning algorithm in the transfer learning framework. Finally, we describe proposed features for the edge sign prediction problem.

3.1. Problem Formulation

A newly signed social network is called *Target Graph* for edge sign prediction. It is a directed graph $G_t = (V_t, E_t^l, E_t^u, S)$. We let V_t denotes the set of vertices, E_t^l denotes the set of edges with sign labels, E_t^u denotes the set of edges whose signs are unknown, and S is a mapping function which denotes the signs of edges (*positive or negative*).

Because the paucity of available signs in newly signed social network makes it difficult to train a good classifier, we need to leverage another more mature signed social network. Thus, we have another directed graph $G_s = (V_s, E_s, S)$ which is called *Source Graph*. We let V_s denotes the set of vertices, E_s denotes the set of edges with sign labels.

Formally, we let $T = T_s \bigcup T_t$ denotes the training data. $T_s = \{e_s, S(e_s)\}, \forall e_s \in E_s$, and $T_t = \{e_t, S(e_t)\}, \forall e_t \in E_t^l$. Let E_t^u denotes the testing data. For an edge instance $e = (u, v) \in T$, we encode the essential information of edge into feature vector for training step. Each feature vector is labeled as positive or negative. Feature vectors which are extracted from training data and their labels are used to train a prediction model.

3.2. Transfer Learning Through TAS

Source graph may have a different joint distribution of the edge instances and the class labels from the target graph. Besides good knowledge, source graph also contains noisy data. The useful knowledge from source graph is advantageous to the process of classification, while the noisy part of the data does not affect the classifier too much. Thus, training data are abundant, but the basic classifiers learn from these data can not classify the testing data well due to different data distributions. We need a prediction model that leverages the labeled instances in both the source and target graphs. Therefore, we construct *Transfer AdaBoost with SVM*. To construct this prediction model, we borrow the AdaBoost idea from Dai et al[4] and AdaBoost with SVM-based component classifiers from Li et al[13].

AdaBoost (*Freund and Schapire,1997*)[8] is a learning framework which aims to boost the accuracy of a weak learner (*component classifier*) by carefully adjusting the weights of training instances and learns a classifier accordingly. But, for source graph training instances, when they are wrongly predicted due to distribution changes by the prediction model, these instances could be those that are the most dissimilar to the target graph instances. Therefore, this learning algorithm doesn't train a good classifier. To solve this problem, we borrow the *Transfer AdaBoost* from Dai et al, which extends AdaBoost (*Freund and Schapire, 1997*) for transfer learning. We should give edge instances in T_s, that are less similar to the target edge instances in T_t, smaller weights to weaken their impacts; conversely, for edge instances in T_s that are more similar to the target edge instances in T_t, we should give larger weights to attach more importance to them.

Previous researches of transfer learning that use Decision Trees[6] or Neural Networks[18] as component classifiers in AdaBoost have been reported. Still, some difficulties remain. What should be the suitable tree size when Decision Trees are used as component classifiers? How could the complexity be controlled to avoid overfitting when Neural Networks are used as component classifiers? To solve this problem, we borrow the *AdaBoostSVM* from Li et al. The AdaBoostSVM is AdaBoost incorporating properly designed RBFSVM (*SVM with the RBF kernel*) component classifiers. From Li et al[13], the distributions of accuracy and diversity over RBFSVM component classifiers by designing parameter adjusting strategies have promising results. Li et al demonstrate AdaBoost approach that uses RBFSVM component classifiers outperforms other AdaBoost approaches using component classifiers such as Decision Trees and Neural Networks. Thus, we incorporate properly designed *RBFSVM* (SVM with the RBF kernel) component classifiers in *Transfer AdaBoost*. RBFSVM uses two regularization parameters: C controls its model complexity and training error; γ is the free parameter of the Gaussian radial basis function. We select proper values of C and γ following Li et al[13].

A formal description of Transfer AdaBoost with SVM (*TAS*) is given in Algorithm 1. In TAS, we use $w_1, \ldots, w_n$ to denote the weights of edges in T_s, and $w_{n+1}, \ldots, w_{n+m}$ to denote the weights of edges in T_t. For an edge e, $P_t(e) \in [-1, 1]$ is the predicted edge sign for e, and $S(e)$ is the true edge sign. Because the use of source graph is leverage for target graph, the source graph edges will never have a larger influence than the target graph edges. Therefore, the weights of source graph edges would never increase and are always less than those of target graph edges. For any target graph edge $e_t \in E_t^l$, its weight will always get increased by a factor of $\beta_t^{-\frac{|P_t(e_t)-S(e_t)|}{2}} \in [1, +\infty)$. For any source graph edge $e_s \in E_s$, its weight will always get decreased by a factor of $\beta^{\frac{|P_t(e_s)-S(e_s)|}{2}} \in (0, 1]$.

3.3. Proposed Features

Previous study of edge sign prediction problem for a newly signed social network[20] uses *explicit topological features* (node degree, betweenness centrality, triad count and edge embeddedness) which express manifest properties of the edge instances in the source or target graph and *latent topological features* which can capture the common patterns between source graph and target graph. However, with *latent feature*, when the distributional differences between the source and target graphs become larger, the transfer learning performance becomes worse. Thus, we need a feature which is more generalizable than *latent feature*. In this paper, we propose to use *PNR feature*[19] which is based on the strong theory of decision-making.

Algorithm 1: Transfer AdaBoost with SVM (TAS)

```
Input: The two labeled data $T_s$ and $T_t$, the unlabeled data $E_t^u$, a SVM-based component classifiers RBFSVM,
       and the maximum number of iterations $K$.
Output: Edge sign classifier $P$.
begin
    Let $n \longleftarrow |T_s|$, $m \longleftarrow |T_t|$
    The initial weight vector, that $w^1 = (w_1^1, \ldots, w_n^1, w_{n+1}^1, \ldots, w_{n+m}^1)$. We allow the users to specify the initial
    values for $w^1$.
    for $t = 1, \ldots, K$ do
        1. Set $q^t \longleftarrow w^t / (\Sigma_{i=1}^{n+m} w_i^t)$
        2. Call RBFSVM component classifier that is provided the combined training set $T$ with the distribution
        $q^t$ over $T$ and the unlabeled data $E_t^u$ , $P_t$: $F(e) \longrightarrow P_t(e) \in [-1, 1]$.
        3. Calculate the error of $P_t$ on $T_t$:
            $$e_t = \frac{\Sigma_{i=n+1}^{n+m} q_i^t . \frac{|P_t(e_i) - S(e_i)|}{2}}{\Sigma_{i=n+1}^{n+m} q_i^t}$$
        4. If $e_t > 0.5$, select new appropriate parameters ($C$ and $\gamma$) following[13] for an RBF kernel. Then, go to
        (2).
        5. Set $\beta_t = \frac{e_t}{1-e_t}$, $\beta = \frac{1}{1+\sqrt{2\ln(n)/K}}$
        6. Update weight vector $w^t$
            $$w_i^{t+1} = \begin{cases} w_i^t \beta^{\frac{|P_t(e_i)-S(e_i)|}{2}} & 1 \le i \le n \\ w_i^t \beta_t^{-\frac{|P_t(e_i)-S(e_i)|}{2}} & n+1 \le i \le n+m \end{cases}$$
    $$P(e) = \begin{cases} 1 & \text{if } \sum_{t=1}^{K} \log \frac{1}{\beta_t} . P_t(e) \ge 0 \\ -1 & \text{otherwise} \end{cases}$$
```

We present how to construct PNR feature for edge sign prediction. For a directed edge $e = (u, v)$, we use $d_{out}^+(u)$ to denote the number of positive outgoing edges at u and $d_{out}^-(u)$ to denote the number of negative outgoing edges at u. Similarly, $d_{in}^+(v)$ and $d_{in}^-(v)$ are the number of positive and negative incoming edges at v. The term of ε is an extremely small value to ensure that the denominators are nonzero. We are interested in the ratios between four terms at u and v:

$$R_{out}(u) = \frac{d_{out}^+(u)}{d_{out}^-(u) + \varepsilon} \tag{1}$$

$$R_{in}(v) = \frac{d_{in}^+(v)}{d_{in}^-(v) + \varepsilon} \tag{2}$$

where $R_{out}(u)$ is the proportion between positive and negative outgoing edges at u and $R_{in}(v)$ is the proportion between positive and negative incoming edges at v. Additionally, since R_{out} and R_{in} may reach positive infinity, their wide values ranges cause difficulties in the learning step. To overcome this, a threshold t is used to cut the ranges down.

$$R_{out}^t(u) = \min(R_{out}(u), t) \tag{3}$$

$$R_{in}^t(v) = \min(R_{in}(v), t) \tag{4}$$

The PNR feature of the edge $e = (u, v)$ is given by concatenating two limited ratios:

$$PNR(u, v) = (R_{out}^t(u), R_{in}^t(v)) \tag{5}$$

The strength of this feature is that it has a close connection with the decision-making theory in terms of past experience; benefit maximization; herd behaviour; anchoring and adjustment heuristic.

- *Past experience*: Past decisions influence on future decisions. An example is that a person usually gives more positive votes than negative votes. It becomes the habit that this person tends to give positive votes. For an edge $e = (u, v)$, the first element $R_{out}(u)$ indicates the voting history of the voter u.
- *Benefit maximization*: The quality of the object affects directly the decisions of the voters and then, it should be considered for prediction. We can evaluate its quality indirectly by taking its incoming edges. The large value of R_{in} is an evidence for its good quality and otherwise. Generally, a good object tends to receive more positive votes from new people and otherwise.
- *Herd behaviour*: Herd behaviour in human societies is defined as a phenomenon in which independent people observe and mimic the actions of others, even mistaken. An example is that a person with many fans (large value of R_{in}) may usually receive more friend requests than hostile relationships.
- *Anchoring and Adjustment heuristic*: Heuristics are general strategies which can help us make right decisions quickly. In fact, when a person wants to make friend with a stranger, such initial information (likes/dislikes, friends/enemies, etc) will be useful anchors for this person to reach better estimate. In the PNR feature, these anchors are encoded into the term of R_{in} to enrich prediction with necessary information.

While R_{out} represents the past experience of voters, R_{in} implies the principles of benefit maximization, herd behaviour, anchoring and adjustment heuristic. This theoretical foundation helps *PNR feature* significantly outperform the *latent feature* in all aspects: the accuracy, the generalization and the speed. Experimental results demonstrate that *PNR feature* is fitter than *latent feature* when we use them in the transfer learning performance.

We have constructed both explicit and PNR features for edge sign prediction. For an edge instance $e = (u, v) \in E_t^l$ with label $S(e)$, we have 11 features, including node degrees $deg_{out}(u)$ and $deg_{in}(v)$, betweenness centrality $f_{bc}(u)$ and $f_{bc}(v)$, triad counts $f_{FF}(e)$, $f_{FB}(e)$, $f_{BF}(e)$, $f_{BB}(e)$, edge embeddedness $f_{eb}(e)$, PNR features $R_{out}^t(u)$ and $R_{in}^t(v)$. Similarly, we can define features for edge instances in E_s.

4. Experiments

In this section, we present our experiments to evaluate the our method. Firstly, we describe data preparation and evaluation methods. Secondly, we briefly introduce four baseline methods which are compared with our method and present some detailed settings of five methods. Finally, we present experimental results and discussions.

4.1. Data Preparation and Evaluation Methods

We use three online social networks Epinions, Slashdot and Wiki. All networks are downloaded from Stanford Large Network Dataset Collection[4]. Because the original graphs are too large and sparse, we select 19,987 nodes from Epinions, 15,999 nodes from Slashdot and 6,998 nodes from Wiki with the highest degrees[20]. Table 1 shows the statistics of the three extracted networks.

Table 1. Statistics of three extracted networks.

Galleries	**Nodes**	**Edges**	**Positive Edges**	**Negative Edges**	**(%) Positive Edges**
Epinions	19,987	634,209	555,601	78,608	87.6
Slashdot	15,999	371,122	283,993	87,129	76.5
Wiki	6,998	113,844	83,832	30,012	73.6

As the edge signs in all these networks are overwhelmingly positive, we overcome this bias by following the methodology of Guha et al.[9] to generate balanced databases. We consider each pair of networks out of the three. We

[4] http://snap.stanford.edu/data/index.html

use one network as the source graph and the other as the target graph. There are totally 6 pairs to test. In each target graph, we partition the edge instances into four parts. We use one part as the testing data E_t^u and randomly sample 10 percentage of edge instances in the remaining three parts to form the labeled edge set E_t^l. This E_t^l and E_s in the source graph form the training data.

To evaluate the our method, we use three degrees: *accuracy*, *precision* and *recall*[5]. Accuracy, precision and recall are the basic measures used in evaluating a classification model. Besides, we compare performance time of the methods.

4.2. Experimental Settings

For evaluation purpose, we use MATLAB to set up following five methods:

- *Target*: using labeled edge instances in the target graph for training with SVM model (the RBF kernel).
- *Combine+Latent*: using all edge instances in the source graph and labeled edge instances in the target graph for training with SVM model (the RBF kernel).
- *TAS+Latent*: using all edge instances in the source graph and labeled edge instances in the target graph for training with TAS.
- *Combine+PNR*: using both source graph edges and labeled target graph edges for training with SVM model (the RBF kernel).
- *TAS+PNR* (our method): using both source graph edges and labeled target graph edges for training with TAS.

Target, *Combine+Latent* and *TAS+Latent* use the explicit and latent topological features that are proposed in[20]. *Combine+PNR* and *TAS+PNR* use our proposed features (explicit features and PNR features). Besides TAS algorithm, we use SVM model with a RBF kernel because this kernel usually outperforms the different kernels in both accuracy and convergence time[1].

We construct latent features following[20]. However, Ye et al don't publish value of parameters such as the trade-off regularization parameter α and the convergence threshold for an iterative update algorithm. Moreover, they don't also introduce classifier methods initializing latent feature matrices. Thus, after some preliminary test, we set trade-off parameter α is 10, convergence threshold is 10^{-1}. To initialize latent feature matrices, we implement Naive Bayes classifier (*a widely used classifier method*).

4.3. Experiments Results and Discussions

We evaluate our method based on four criteria: accuracy, precision, recall and speed (performance time). We have 6 pairs (source - target) from three networks Epinions, Slashdot and Wiki. Fig 3 shows the accuracies of *Target*, *Combine+Latent*, *TAS+Latent*, *Combine+PNR* and *TAS+PNR* on 6 pairs. This shows that *TAS+PNR* has the best edge sign prediction result. In Fig 4, we present five Precision Recall Curve (PR curve) for methods on 6 pairs. The precision recall area under curve (PR AUC) is just the area under the PR curve. The higher it is, the better the method is. Therefore, our method has the best result on most pairs.

From Fig 3 and Fig 4, we can see that our method (*TAS+PNR*) can improve the prediction accuracy by 40% over baseline methods. In the first group of experiment, we use Epinions as the source graph and Slashdot as the target graph. *Target* has the worst result because these method uses only small amount of edge sign information in target graph. *Combine+Latent* can improve the accuracy over *Target* but the noise in the source edge instances may become more obvious. Thus, *TAS+Latent* has better than *Combine+Latent*. However, the latent feature becomes worse when the distributional differences between the source and target graph become large. The PNR feature that is based on the strong theory of decision-making has high generalization. Therefore, *Combine+PNR* and *TAS+PNR* have high accuracy. With TAS algorithm, we replace latent feature with PNR feature. Our method has the best result. We can observe similar trends in residual pairs.

In the end, we turn to the speed evaluation of methods. We measure the performance time of feature extraction, training and prediction. All our experiments are conducted on the same PCs with 2.90 GHz CPU and 12G RAM. First, we compare the speed of feature extraction. Then, we measure the performance time of training and prediction. Table

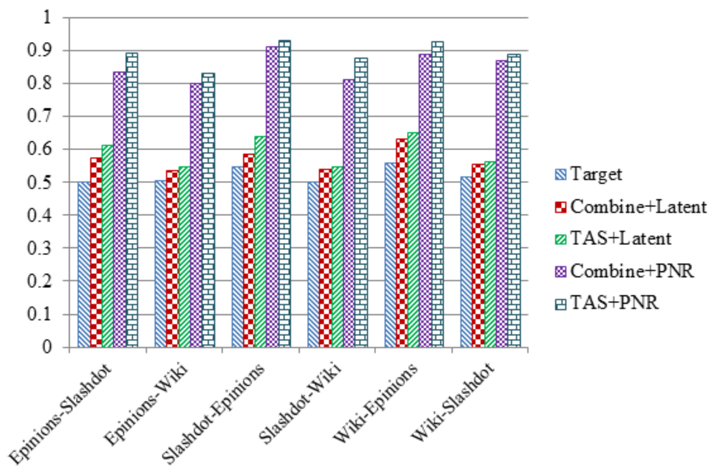

Fig. 3. Prediction Accuracy with 10% of Labeled Target Edge Instances.

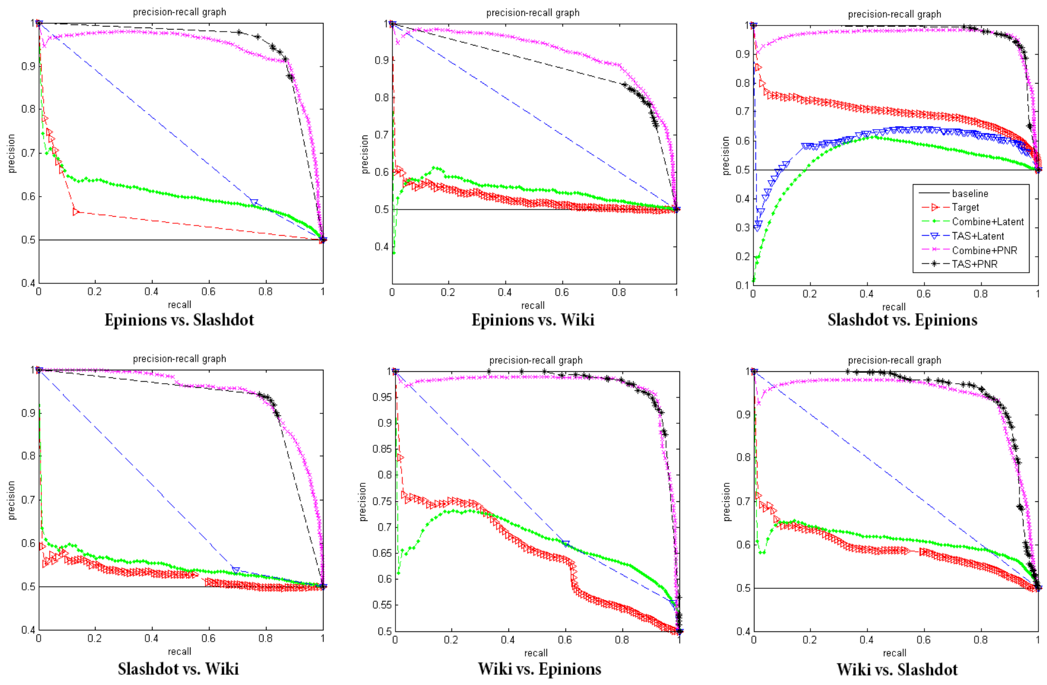

Fig. 4. PR Curve with 10% of Labeled Target Edge Instances.

2 shows the speed of two way feature extraction: explicit features combine latent features, explicit features combine PNR features. In table 3 and table 4, we present the speed of training and prediction with SVM and TAS.

From table 2, table 3 and table 4, when we replace latent feature with PNR feature, the speed of feature extraction, training and prediction is faster. The performance time decrease because PNR has two benefits: low cost feature, simple implementation.

Table 2. Performance time (in minute) of feature extraction (E - Epinions, S - Slashdot, W - Wiki).

	E - S	**E - W**	**S - E**	**S - W**	**W - E**	**W - S**
Explicit+Latent	190.10	144.46	83.66	54.46	38.73	53.53
Explicit+PNR	**143.19**	**115.96**	**50.14**	**34.14**	**24.21**	**35.44**

Table 3. Performance time (in minute) of training and prediction with *SVM* (E - Epinions, S - Slashdot, W - Wiki).

	E - S	**E - W**	**S - E**	**S - W**	**W - E**	**W - S**
Combine+Latent	600.16	1595.10	4390.40	251.36	955.97	75.73
Combine+PNR	**8.84**	**7.22**	**18.19**	**30.16**	**2.17**	**2.37**

Table 4. Performance time (in minute) of training and prediction with *TAS* (E - Epinions, S - Slashdot, W - Wiki).

	E - S	**E - W**	**S - E**	**S - W**	**W - E**	**W - S**
TAS+Latent	4273.20	4865.50	8185.40	5016.60	852.42	978.84
TAS+PNR	**1070.40**	**1073.70**	**1772.00**	**1055.20**	**214.71**	**302.63**

5. Conclusion

In this paper, we studied the problem of signed link prediction in social networks that have both positive and negative links. We focus how to effectively predict sign links in newly signed social network whose the edge sign information is very scarce. We propose a novel transfer learning framework called *TAS* that extends boosting-based learning algorithms and incorporates properly designed RBFSVM component classifiers. *TAS* can select the most useful source graph instances as additional training data for predicting the labels of target graph techniques when the noise in the source graph instances cause the model to predict wrongly on the test edges from the target graph. Besides, we replace *latent feature* with *PNR feature* that is low cost feature and has a close connection with the decision-making theory in terms of past experience; benefit maximization; herd behaviour; anchoring and adjustment heuristic. The results of experiments on three networks Epinions, Slashdot and Wiki show that our method really improves on previous methods in two criteria accuracy and speed significantly.

References

1. Ben-Hur A, Weston J. A Users Guide to Support Vector Machines. *Methods in Molecular Biology*; 2010. p. 223-239.
2. Burke M, Kraut R. Mopping up: Modeling wikipedia promotion decisions. *In Proc. CSCW*; 2008.
3. Chiang K, Nagarajan N, Tewari A, Dhillon IS. Exploiting longer cycles for link prediction in signed networks. *In Proceedings of the 20th ACM Conference on Information and Knowledge Management*;2011. p. 1157-1162.
4. Dai W, Yang Q, Xue G, Yu Y. Boosting for Transfer Learning. Proc. 24th Intl Conf. *Machine Learning*; 2007. p. 193-200.
5. Davis J, Goadrich M. The relationship between precision-recall and roc curves. *Technical report 1551, University of Wisconsin Madison*; 2006.
6. Dietterich TG. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning 40 (2)*; 2000. p. 139-157.
7. DuBois T, Golbeck J, Srinivasan A. Predicting trust and distrust in social networks. *In Proceedings of the third IEEE International Conference on Social Computing, SocialCom*; 2011.
8. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences 55 (1)*; 1997. p. 119-139.
9. Guha RV, Kumar R, Raghavan P, Tomkins A. Propagation of trust and distrust. *In Proc. 13th WWW*; 2004.
10. Kunegis J, Lommatzsch A, Bauckhage C. The Slashdot Zoo: Mining a social network with negative edges. *In Proc. 18th WWW*; 2009. p. 741-750.
11. Leskovec J, Huttenlocher D, Kleinberg J. Predicting positive and negative links in online social networks. *In WWW*; 2010. p. 641-650.
12. Leskovec J, Huttenlocher D, Kleinberg J. Signed networks in social media. *In Proc. 28th CHI*; 2010.
13. Li X, Wang L, Sung E. Adaboost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence, 21*; 2008.
14. Liben-Nowell D, Kleinberg J. The link prediction problem for social networks. *In CIKM*; 2003. p. 556-559.
15. Massa P, Avesani P. Controversial users demand local trust metrics: an experimental study on epinions.com community. *In AAAI 05, AAAI Press*; 2005. p. 121-126.

16. Pan SJ, Yang Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*; 2010. p. 1345-1359.
17. Raina R, Battle A, Lee H, Packer B, Ng AY. Self-taught learning: Transfer learning from unlabeled data. *In ICML*; 2007. p. 759-766.
18. Schwenk H, Bengio Y. Boosting neural networks. *Neural Computation 12*; 2000. p. 1869-1887.
19. Tuyen H, Thanh V, Bac L. A Decision-making based Feature for Link Prediction in Signed Social Networks. *In IEEE RIVF*; 2013. p. 169-174.
20. Ye J, Cheng H, Zhu Z, Chen M. Predicting positive and negative links in signed social networks by transfer learning. *In Proceedings of the 22nd international conference on World Wide Web, International World Wide Web Conferences Steering Committee*; 2013. p. 1477-1488.
21. Zhuang F, Luo P, Xiong H, He Q, Xiong Y, Shi Z. Exploiting associations between word clusters and document classes for cross-domain text categorization. *In SDM*; 2010.