Improving Link Prediction in Social Networks by User Comments and Sentiment Lexicon

Feng Liu, Bingquan Liu, Chengjie Sun, Ming Liu and Xiaolong Wang

School of Computer Science and Technology,
Harbin Institute of Technology, Harbin, China
{fengliu, liubq, cjsun, mliu, wangxl}@insun.hit.edu.cn

Abstract. In some online Social Network Services, users are allowed to label their relationship with others, which can be represented as links with signed values. The link prediction problem is to estimate the values of unknown links by the information from the social network. A lot of similarity based metrics and machine learning based methods are proposed. Most of these methods are based on the network topological and node states. In this paper, by considering the information from user comment and sentiment lexicon, our methods improved the performances of link prediction for both similarity based metrics and machine learning based methods.

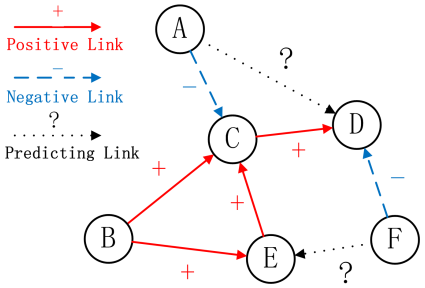
Keywords: Link Prediction, Social Networks, Link Network Structure, User Comments, Sentiment Lexicon

1 Introduction

With the explosive growth of SNS(Social Network Services) websites, there are large scale data of social media. The mass data includes the interactions among social members, such as comments and links. The comment is always a short paragraph with only one or a few sentences, which are sent from one user to another. The link is usually a label with sign value that represent one user's certain kind of opinion to another, such as expressing support or oppose.

Taking social members as vertexes and links as directed edges, the link network can be modeled as graph. The classical link prediction, as shown in Fig. 1(a), is to predict the relation of one user toward another from the evidence provided by their relations with other members from the surrounding social network[10, 15]. The survey[24] introduced that the state-of-art link prediction methods can be roughly divided into two classes: user similarity based metrics and machine learning based methods.

The user similarity based metrics are usually used for recommending tasks, such as co-authorship prediction problem[1, 5, 9], and friendship prediction problem[2, 7, 12]. Their metrics are mainly based on collaborative filtering of users and articles, or analyzing the topology of link network. showed The similarity based metrics is low computing cost and can predict top k recommendations



(a) predicting via link network

Link	Comments
(+)	Great contributions.
(+)	Unconvincing reasons for adminship, but generally good contributions.
(-)	User is too new, too "speedy"-happy.
(+)	Too smart to be an admin.
(-)	I do not trust him with the tools.
(?)	Solid answers, strong contributions, good content editor.
(?)	A good editor, but quite lacking in the areas admins inevitably should work in.

(b) predicting via user comments

Fig. 1. The Link Prediction Problem.

with good performance[21]. But when the k value becomes large, its performance drops sharply.

The machine learning based methods treated the link prediction problem as a classification task. Logistic regression model is used to predict links' values based on features extracted from link network in [13]. Support vector machine is used to analyze how link network structure features effect link's values in [20]. Deep belief network based approaches for link prediction is introduced in [18, 19]. These methods need the detail structure of link network to train a high performance model. As a result, in the condition that only a small part of the link network is observed, these methods would not works well. This problem is named as 'Cold Start'.

The sentiment analysis or opinion mining is the computational study of people's opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes [16, 17]. Taking the 'link' as one user's opinion to another user, the link prediction task is some kind like the sentiment analysis or opinion mining task, as shown in Fig. 1(b). The whole document is modeled as a vector of words, and the support vector machine classifier is learned on word vectors to determine whether a review is positive or negative in [22]. The combination of lexical and syntactic features are used to predict the twitter's sentiment properly [4]. User comments are used in [25] to improve the performance of state theory introduced in [14]. It shows that the link prediction problem can also be solved by using user comments with opinion mining methods, which do not need to know the structure of link network.

The main contributions of this paper are summarized as following:

1. In order to improve the performance for similarity based metrics with large k value, we considered both the topology of link network and information from sentiment lexicon, and proposed the method SentiTNS;
2. To solve the cold start problem, we added features from user comments and sentiment lexicon when learning a model. The experiment results show that our method works well and the performance of link prediction is improved.

2 Information Sources for Link Prediction

2.1 Link Network Structure

The link network could be represented as a directed graph $G = (V, E)$, where V is the set of users and E is the set of edges. The edge directly linking from user u to v is denoted as $e(u, v)$, and the set containing all the paths from u to v is denoted as $Path(u, v)$. And denote $Ne(u)$ as the set of u's neighbour nodes and $CNe(u, v)$ as the common neighbours shared with u and v.

The topology information contains the shortest path $p(u, v)$ calculated from $Path(u, v)$. The node features include the in-degrees and out-degrees with sign values. Denote $d^+_{in}(u)$ for positive in-degree, $d^-_{in}(u)$ for negative in-degree, $d^+_{out}(u)$ for positive out-degrees and $d^-_{out}(u)$ for negative out-degrees. There are totally 8 kinds of node features from node u and v.

The neighbour features includes statistical information from $CNe(u, v)$, such as $c^N_{Ne}(u, v)$ is the number of nodes w in $CNe(u, v)$ and $c^E_{Ne}(u, v)$ is the number of edges between w and u, v. Then select any node w from $CNe(u, v)$, whose edges could have any direction with any sign value connected with u and v, denote $c(u^+_{\rightarrow} w^+_{\leftarrow} v)$ as the number of nodes who get positive links from both u and v. There are 2 directions and 2 kinds of sign values, so the relationships of u, v and w can be divided into 16 kinds. There are totally 18 kinds of neighbour features from node u and v.

2.2 User Comments and Sentiment Lexicon

The text, such as comment, could be represented by the Bag of Words model(BOW). It treats each comment as the bag of its words, and represents text as a vector of words via the word dictionary. The word dictionary Dic contains all the appeared words, and the dictionary size is $lenDic$. The set of words appeared in comment from u to v is denoted as $W(u, v)$. Then build a word vector $w(u, v)$ with dimension $lenDic$, and set the ith position to '1' if the Dic 's $ith\ word \in W(u, v)$, while all other positions are set to '0'.

Sentiment lexicon is a kind of dictionary that contains sentiment scores for words. The scores reflects the degrees of positivity, negativity, or neutrality for the sentiment or opinion which held by words. We extend $w(u, v)$ with sentiment scores. We build a sentiment vector $senti(u, v)$ with dimension $2 \times lenDic$. If the ith position w_i is '1' in $w(u, v)$, we set $senti(u, v)$'s ith position to the w_i's positive sentiment score and set its $(lenDic + ith)$ position to the w_i's negative sentiment score. So $senti(u, v) = (s_1, ...s_{lenDic}, s_{lenDic+1}, ...s_{2\times lenDic})$. It maybe that not all words in Dic appear in the sentiment lexicon, and we use '0' as their sentiment scores.

The details of how to get user dictionary words' positive and negative polarity scores is shown in Algorithm 1. The example uses the sentiment lexicon SentiWordNet, which is based on WordNet. The synset is basic unit in WordNet, so SentiWordNet is also structured by $SynsetTerms$. One word can belongs to many synsets, and each synset has its own positive polarity score and negative

polarity score. We use the average score of all synsets, named as $SynsetTerms$ in SentiWordNet, that a word belongs to as the word's scores.

Input: User word dictionary $WordList$; Sentiment lexicon $SentiWordNet$
Output: Scores of words' positive polarity $PosScoreList$; Scores of words' negative polarity $NegScoreList$

```
for each record in SentiWordNet do
    for each word in record.SynsetTerms do
        if (wordIndex = wordList.index(tempWord)) then
            PosScoreList[wordIndex] += record.PosScore;
            NegScoreList[wordIndex] += record.NegScore;
            WordSynsetCount[wordIndex] += 1;
        end
    end
end
for i in range (0,len(WordList)) do
    if (WordSynsetCount[wordIndex] != 0) then
        PosScoreList[i] = PosScoreList[i]/WordSynsetCount[wordIndex];
        NegScoreList[i] = NegScoreList[i]/WordSynsetCount[wordIndex];
    end
end
```

Algorithm 1: Processing sentiment lexicon

3 Improving Similarity Based Metrics

3.1 SentiSUB

In some conditions, it is difficult to get the structure of link network, even none link could be observed. The similarity based metrics based on link network structure become unavailable at that time. In order to estimate a link's value in such condition, we design a naive method, SentiSUB by calculating the sentiment polarity of the link starting user's comment as:

$$SentiSUB(u,v) = \sum_{i=1}^{l} s_i - \sum_{j=l+1}^{2\times l} s_j \tag{1}$$

where $l = lenDic$ and $s_i \in senti(u,v)$ as defined in Section 2.2.

3.2 SentiTNS

In the condition that the link network structure is observable, the similarity based metrics become available. We also improve its metric by taking the sentiment of comment into account. [23] introduced FriendTNS which takes both positive and negative links into account when calculating user similarity.

The FriendTNS is based on transitive node similarity, which calculates the similarity of two indirectly connected users by using their shortest path, $p(u,v) = (u, w_2, ..., w_i, ..., w_{l-1}, v)$. The $FriendTNS(u,v)$ is calculated as:

$$FriendTNS(u,v) = \begin{cases} 0 & \text{disconnected} \\ sim(u,v) & \text{neighbors} \\ \prod_{i=1}^{l-1} sim(w_i, w_{i+1}) & \text{otherwise} \end{cases} \tag{2}$$

and

$$sim(u,v) = \frac{1}{\partial(u) + \partial(v) - 1} \tag{3}$$

where $\partial(u) = d^+_{in}(u) + d^-_{out}(u) - d^+_{out}(u) - d^-_{in}(u)$ is defined by the state theory introduced by [14].

We propose the method SentiTNS that considers both kinds of information from link network structure and user comments sentiment. We extent the node's link degrees $d^+_{out}(u), d^+_{in}(u), d^-_{out}(u), d^-_{in}(u)$ to sentiment degrees as:

$$\begin{aligned} sd^+_{(out)}(u) &= \sum_v \sum_{i=1}^{l} s_i \quad (v : v \ belongs \ to \ Ne(u)) \\ sd^+_{(in)}(u) &= \sum_u \sum_{i=1}^{l} s_i \quad (u : u \ belongs \ to \ Ne(v)) \\ sd^-_{(out)}(u) &= \sum_v \sum_{i=l+1}^{2\times l} s_i \quad (v : v \ belongs \ to \ Ne(u)) \\ sd^-_{(in)}(u) &= \sum_u \sum_{i=l+1}^{2\times l} s_i \quad (u : u \ belongs \ to \ Ne(v)) \end{aligned} \tag{4}$$

where $l = lenDic$ and $s_i \in senti(u,v)$.

The state theory assumes that one node's link degrees can reflect the node social state in that social network, and its metric works well. Depending on similar assumption, we assume that the node's sentiment degrees could reflects the node 'sentiment state' in the network. Then we extent SentiSUB by changing the Equation (3) as:

$$sentiment\_sim(u,v) = \partial(u) - \partial(v) \tag{5}$$

where $\partial(u) = sd^+_{in}(u) + sd^-_{out}(u) - sd^+_{out}(u) - sd^-_{in}(u)$ calculated by Equation (4).

Then we could calculate the user similarity $SentiTNS(u,v)$ by Equation (2) with $sentiment\_sim(u,v)$, and make recommendation by the rank of similarities between users.

4 Improving Machine Learning Based Methods

The machine learning based methods for link prediction are mainly based on training a statistics model over the link network structure features. There are two directions to improve them, one is using more powerful models, the other

is taking more features into account. More powerful models could improve the performance over the original feature set, but it is not so suitable for solving the cold start problem. That is because in the condition of cold start, when most part of the link network is unobservable, the representation ability of link structure features becomes weak.

From above analysis, taking more features into account could be an effective method for cold start problem. We added the word vector and sentiment vector as features for training a model. As some other sentiment classification and opinion mining methods, the model Support Vector based Classifier(SVC) is used[6, 22]. For testing the effectiveness of word vector and sentiment vector, we learned models over each source of features individually and tried all possible combinations.

5 Experiments and Analysis

5.1 Experiments Setup

In our experiments, the social media data of Wikipedia RfA [25] is used. Before build the comment word vector, we removed all the words such as 'support', 'supporting', 'supported' and similar words for 'oppose'.

As a sentiment lexicon is needed for our method, the SENTIWORDNET 3.0[3] is used. When calculate the prior polarities for words, we used similar method $mean_m$ introduced in [11]. It calculates the mean of the positive and negative scores for all the senses over synsets, so the part of speech tagging is not needed.

We extracted features and implemented the similarity based metrics by programming in Python, and the toolkit 'LIBLINEAR'[8] is used for learning SVC. In experiments of machine learning based methods, we balanced the data by randomly dropping positive links to avoid the samples imbalance effects, and totally 79660 balanced samples is used. The 5 fold cross-validation is performed to avoid the performance metric variance caused by sample distribution.

5.2 Similarity Based Metrics

The experiment results of similarity based metrics is shown in Fig. 2. We recommend top k links as positive and bottom k links as negative for a user at the same time. If a user does not have more than k positive or negative links, the k value is set to the number of how many links does that user has. We use FriendTNS, inteoduced in [23], as the baseline method. All metrics perform well with small k values, and their accuracies drop when k becomes larger. That is because when k becomes large, the difference between positive link similarity and negative link similarity becomes little, and the metric makes wrong recommendation.

When k is less than 10, the performance of SentiSUB is best. It shows that comments with strong sentiment polarity could represent the link's value properly. When the value of k becomes larger than 20, SentiTNS performs better than

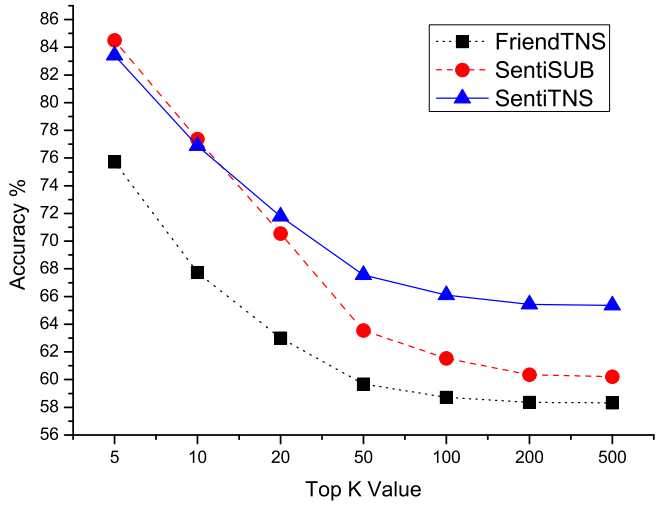


Fig. 2. Similarity Based Metrics

the other two. Even when k = 500, nearly all links are recommended as positive or negative, SentiTNS still has accuracy over 65%. It means the sentiment state theory is suitable for recommending most links in the network. FriendTNS also has good performance when k is little, and it only uses the information from link network.

5.3 Machine Learning Based Methods

The experiment results of machine learning based metrics is shown in Fig. 3. 'LNS' is link network structure feature, 'BOW' is comment word feature, and 'SEN' is the sentiment score feature. The '+' means using the two sources of features together, and 'All' is using all the three sources of features. The X axis is the observed scale of link network by randomly lost none, 10%, and up to 90% links in the network when collecting LNS.

The performance of model learned over LNS drops with the decrease of observed scale. This phenomenon is caused by the cold start problem. The performances of models learned over BOW, SEN and BOW+SEN are 3 parallel lines with X axis, Because they do not effect by the link network structure. When add other features with LNS, the curves do not drop fast with the observed scale. It means that the cold start problem could be solved properly by this method. We get the best accuracy with model learned over all features. It means that the performance of link prediction is improved than only using LNS.

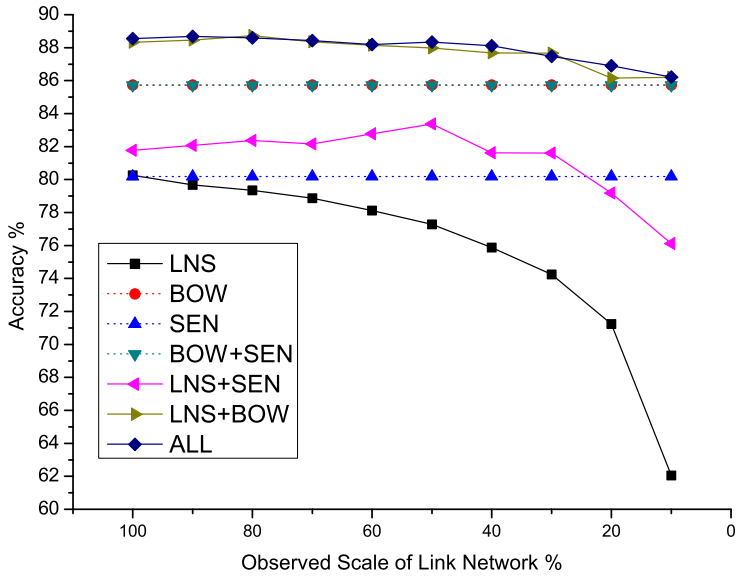


Fig. 3. Machine Learning Based Methods

The cure of BOW+SEN is nearly the same as BOW, and cure of ALL(LNS+BOW+SEN) is nearly the same as LNS+BOW. And the curve of LNS+SEN shakes the most over all curves. It shows that the sentiment vector features are not so suitable for combination with other features, when training a SVC.

6 Conclusion

In this paper, by taking information from user comment and sentiment lexicon into account, methods for link prediction based on similarity metrics and machine learning models are proposed. The method SentiTNS considers both the topology of link network and user comment's sentiment polarity, and it has good performance for recommending links with large top k value. The feature combination method can solve the cold start problem properly and it also improves the link prediction performance in conditions that most part of the link network is observable.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (61272383 and 61300114), Specialized Research Fund for the Doctoral Program

of Higher Education (No. 20132302120047), the Special Financial Grant from the China Postdoctoral Science Foundation (No.2014T70340), China Postdoctoral Science Foundation (No.2013M530156), and Natural Science Foundation of Heilongjiang Province(F201132).

References

1. Al Hasan, M., Chaoji, V., Salem, S., Zaki, M.: Link prediction using supervised learning. In: SDM?06: Workshop on Link Analysis, Counter-terrorism and Security. pp. 1–10. SIAM, Philadelphia, PA, USA (2006)
2. Anderson, A., Huttenlocher, D., Kleinberg, J., Leskovec, J.: Effects of user similarity in social media. In: Proceedings of the fifth ACM international conference on Web search and data mining. pp. 703–712. ACM (2012)
3. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: LREC. vol. 10, pp. 2200–2204 (2010)
4. Becker, L., Erhart, G., Skiba, D., Matula, V.: Avaya: Sentiment analysis on twitter with self-training and polarity lexicon expansion. Atlanta, Georgia, USA p. 333 (2013)
5. Brzozowski, M.J., Hogg, T., Szabo, G.: Friends and foes: ideological social networking. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 817–820. ACM (2008)
6. Chalothom, T., Ellman, J.: Simple approaches of sentiment analysis via ensemble learning. In: Information Science and Applications, pp. 631–639. Springer (2015)
7. Chelmis, C., Prasanna, V.K.: Social link prediction in online social tagging systems. ACM Transactions on Information Systems (TOIS) 31(4), 20–46 (2013)
8. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. The Journal of Machine Learning Research 9, 1871–1874 (2008)
9. Freno, A., Garriga, G., Keller, M.: Learning to recommend links using graph structure and node content. In: Neural Information Processing Systems Workshop on Choice Models and Preference Learning. pp. 1–7. NIPS (2011)
10. Getoor, L., Diehl, C.P.: Link mining: a survey. ACM SIGKDD Explorations Newsletter 7(2), 3–12 (2005)
11. Guerini, M., Gatti, L., Turchi, M.: Sentiment analysis: How to derive prior polarities from sentiwordnet. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1259–1269. Association for Computational Linguistics (2013)
12. Kunegis, J., Lommatzsch, A., Bauckhage, C.: The slashdot zoo: mining a social network with negative edges. In: Proceedings of the 18th international conference on World wide web. pp. 741–750. ACM (2009)
13. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: Proceedings of the 19th international conference on World wide web. pp. 641–650. ACM (2010)
14. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Signed networks in social media. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1361–1370. ACM (2010)
15. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American society for information science and technology 58(7), 1019–1031 (2007)

16. Liu, B.: Sentiment analysis and subjectivity. Handbook of natural language processing 2, 627–666 (2010)
17. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Mining Text Data, pp. 415–463. Springer (2012)
18. Liu, F., Liu, B., Sun, C., Liu, M., Wang, X.: Deep learning approaches for link prediction in social network services. In: Neural Information Processing. pp. 425–432. Springer (2013)
19. Liu, F., Liu, B., Sun, C., Liu, M., Wang, X.: Deep belief network-based approaches for link prediction in signed social networks. Entropy 17(4), 2140–2169 (2015)
20. Liu, F., Liu, B., Wang, X., Liu, M., Wang, B.: Features for link prediction in social networks: A comprehensive study. In: Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on. pp. 1706–1711. IEEE (2012)
21. Lü, L., Zhou, T.: Link prediction in complex networks: A survey. Physica A: Statistical Mechanics and its Applications 390(6), 1150–1170 (2011)
22. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 115–124. Association for Computational Linguistics (2005)
23. Symeonidis, P., Tiakas, E.: Transitive node similarity: predicting and recommending links in signed social networks. World Wide Web 17(4), 743–776 (2014)
24. Wang, P., Xu, B., Wu, Y., Zhou, X.: Link prediction in social networks: the state-of-the-art. Science China Information Sciences pp. 1–38 (2014)
25. West, R., Paskov, S.H., Leskovec, J., Potts, C.: Exploiting social network structure for person-to-person sentiment analysis. Transactions of the Association of Computational Linguistics – Volume 2, Issue 1 pp. 297–310 (2014), http://aclweb.org/anthology/Q14-1024