Characterization and Edge Sign Prediction in Signed Networks

Tongda Zhang, Haomiao Jiang, and Zhouxiao Bao
Electrical Engineering Department, Stanford University, Stanford, CA 94305 USA
Email: {tdzhang, hjiang36, zhouxiao}@stanford.edu

Yingfeng Zhang
Industry Engineering Department, Northwestern Polytechnical University, Xi'an, Chian
Email: zhangyf@nwpu.edu.cn

Abstract — In this paper, an intensive study on online social networks is studied. Through the presented method, the relationships between entities can be analyzed to be positive or negative. The positive relationship indicates trust or friendship and the negative relationship represents opposition or antagonism. We investigate some basic characteristics of signed networks and make certain extensions to particular features. A modified version of the PageRank algorithm is proposed, which is applicable to signed networks. Based on the creative features, an edge sign predictor using supervised machine learning algorithms is also established. The experimental results show that our model can significantly improve the prediction accuracy and decrease the false positive rate.

Index Terms—signed network, prediction of edge sign, local bias, SNPR, machine learning

I. INTRODUCTION
A. Background
In recent years, social network has become an increasingly important resource to analyze individual behaviors and the embedded communal structures. In most networks, edges simply indicate connections between entities. However, sometimes more information about the connection needs to be represented, which leads to the generation of signed network.

In signed networks, each edge is either positive (trusts, likes and approves) or negative (distrusts, dislikes and denounces). They are generally used to characterize attitudes among a group of people. There are three typical signed networks commonly considered: Epinions, Slashdot and Wikipedia.

With rich information contained in signed network, we can not only make macro analysis on the evolution and structure of real-world social network but also uncover the hidden relationship between two given nodes. Balance and status were proposed in [1] and further studied in [2], [3] with respect to macro analysis. As for edge prediction, Guha et. al proposed an algorithm based on exponentials of the adjacency matrix in [4]. Jure et. al took it one step further by using machine learning scheme. Besides optimizing prediction accuracy, some papers, like [5], put their focus on improving the false positive rate.

B. Our Work
Initially, we explore basic descriptors such as clustering coefficient, together with some variants, e.g. extension from triangle patterns to quadrangle. We intend to catch an intuitive idea about their relative importance and representativeness of corresponding nodes and the entire network. After calculation using real-world datasets, we conclude with a collection of features containing various amounts of information of the network.

The proliferation of signed networks leads researchers to the problem of predicting the sign of an edge [2, 5, 6]. We follow this line to implement an edge sign predictor which integrates the information of local bias and the SN-PageRank values of each node. After experimental simulations, we confirm that our model significantly outperforms previous ones with respect to edge sign prediction. Since the sign of an edge differs much in different datasets, we then use one dataset to train our model and test on the other one, which indicates that our model is generalized and dataset independent without significant deterioration.

With the set of features in hand, it is valuable to find out which of them are significant or negligible when predicting edge signs. We implement forward feature selection on the whole feature set which helps us clearly distinguish between them.

C. Paper Structure
The remainder of this paper proceeds as follows. Part II gives a brief description of a list of related prior work. Part III focuses on analyzing the datasets we use, from
statistical information and basic descriptors. Part IV describes three newly proposed descriptors and elaborates on their practical meaning in signed networks. In part V, we refine the supervised machine learning model for edge sign prediction and perform some experiments on different datasets. We also implement feature selection and cross dataset validation of the model. In the last part, we draw some conclusions based on the experimental results.

II. RELATED WORK

Here, we will give more details and survey further lines of study that are related, with explanation of relationship and difference between their ideas and our work.

Our first goal is to characterize signed social network with a meaningful model and use it to infer the sign of a given edge. Some previous paper did this with prediction accuracy as their goals. Guha et. al introduced belief propagation concept in [4] and used exponential of adjacent matrix as features of their model. They enumerated possible values for the parameters and obtained the optimal prediction accuracy. Years after, in [2], Jure et. al made some extensions to a model based on logistic regression. In our paper, local information, e.g. in-and-out degrees, is added. We are going to follow their steps to predict with brand new features under various machine learning schemes.

Our next step is to consider reliability of each node. Previous papers insisted that all the nodes and their connections were reliable. However, in rating network, the edge sign is highly dependent on individual interests and preferences. The proposed method is similar to PageRank which is introduced and fully developed in paper [7]-[9]. It can iteratively compute the authority and reliability of each node only in directed graph. We will make modifications to PageRank and apply it in signed networks as an important feature in our prediction model.

We will finally step into the generalization problem. To avoid overfitting, several methods have been proposed among which cross validation [10] is most common. Jure et. al applied this to cross-set validation in [2] to find out the common structure in online networks. Feature selection is widely used as well. Forward feature selection proposed in [11], [12] is an aggressive algorithm to find the smallest valuable feature set with a relatively high accuracy. We will combine the two methods to refine our prediction model.

III. DATASET ANALYSIS

We explore two online networks: Epinions and Slashdot. Epinions is a general consumer review site into which the concept of distrust was introduced to avoid junk reviews. Slashdot is a technology-related news website in which users are allowed to tag each other as friends or foes based on their attitude toward others’ comments.

A. Basic Analysis

Some basic statistics of Epinions and Slashdot are listed in Table I.

<table>
<thead>
<tr>
<th>Property</th>
<th>Epinions</th>
<th>Slashdot</th>
</tr>
</thead>
<tbody>
<tr>
<td># Nodes</td>
<td>131828</td>
<td>82140</td>
</tr>
<tr>
<td># Edges</td>
<td>841372</td>
<td>549202</td>
</tr>
<tr>
<td># Pos Edges</td>
<td>717667(85%)</td>
<td>42507(77%)</td>
</tr>
<tr>
<td># Neg Edges</td>
<td>123705(15%)</td>
<td>12410(22%)</td>
</tr>
<tr>
<td>Largest SCC</td>
<td>41441</td>
<td>27382</td>
</tr>
<tr>
<td>Largest WCC</td>
<td>119130</td>
<td>82140</td>
</tr>
<tr>
<td>Cluster Coef</td>
<td>0.128</td>
<td>0.059</td>
</tr>
<tr>
<td>Transitivity</td>
<td>0.081</td>
<td>0.024</td>
</tr>
</tbody>
</table>

B. Degree Distribution

The log-log plots of degree distribution of the two networks are shown in Fig. 1. Apparently, they both obey the power law distribution. After fitting the data, we can get: For Epinions, \(X_{min} = 2\) and \(\alpha=1.72\).; For Slashdot, \(X_{min} = 1\) and \(\alpha=1.49\).

C. Balancing

Balancing is a special case of our trust propagation model in the scenario of triangles. With directed edges, there are \(2^3 = 8\) kinds of triangles. Take Epinions as an example, 88% of the triangles are balanced and 12% are unbalanced, which supports the balancing theory.

<table>
<thead>
<tr>
<th>Loop Prop Pattern</th>
<th>Number of Triangle Patterns in Epinions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppp</td>
<td>6,096,310</td>
</tr>
<tr>
<td>pnp</td>
<td>259,837</td>
</tr>
<tr>
<td>npp</td>
<td>259,837</td>
</tr>
<tr>
<td>ppn</td>
<td>259,837</td>
</tr>
<tr>
<td>pnn</td>
<td>259,837</td>
</tr>
<tr>
<td>npp</td>
<td>259,837</td>
</tr>
<tr>
<td>npn</td>
<td>259,837</td>
</tr>
<tr>
<td>nnn</td>
<td>259,837</td>
</tr>
</tbody>
</table>

In following tables, p stands for a positive edge and n stands for a negative one. Loop differs from Prop in the
direction of the last edge. We list the number of triangles with different patterns in Table II with data from Epinions.

The success rate of trust propagation is quite high (around 88.78%), which shows that trust and distrust can propagate through routines of length 3.

For routines of length four, namely rectangles, similar result is shown in Table III.

<table>
<thead>
<tr>
<th>Loop pattern</th>
<th>Number of Quadrangle Patterns in Epinions</th>
</tr>
</thead>
<tbody>
<tr>
<td>pppp</td>
<td>195,481,458 407,547,491 4,114,491 17,975,843</td>
</tr>
<tr>
<td>npnp</td>
<td>15,869,360 40,589,460 3,170,792 1,664,807</td>
</tr>
<tr>
<td>npnn</td>
<td>19,077,360 24,501,936 3,744,978 14,915,703</td>
</tr>
<tr>
<td>pnpn</td>
<td>15,869,360 9,952,785 672,188 2,137,360</td>
</tr>
<tr>
<td>npnn</td>
<td>19,077,360 16,926,874 898,121 5,864,047</td>
</tr>
<tr>
<td>nnpn</td>
<td>3,744,978 7,552,592 898,121 6,731,991</td>
</tr>
<tr>
<td>nnpp</td>
<td>2,040,988 2,606,922 672,188 2,381,472</td>
</tr>
<tr>
<td>pnnn</td>
<td>4,114,491 4,433,569 341,496 2,181,778</td>
</tr>
</tbody>
</table>

The average success rate is around 76%. This indicates that trust and distrust can still transmit, to a certain extent, through routines with length four.

IV. CHARACTERIZE SIGNED NETWORK

A. Local Bias

Intuitively, the in-coming edges of one node denote the trustiness or distrust it receives from others while the out-coming edges represent the attitude it expresses to others. Therefore, we have the following definitions for any node in a signed network:

Incoming Local Bias (ILB): the percentage of negative reviews it receives in all the incoming reviews;

Out-coming Local Bias (OLB): the percentage of negative reviews it gives in all of its out-coming reviews.

For simplicity of expression, we define four values:

\[
IP(a) = \sum_{b \in (b,a)\in E(G)} 1\{sign(b \rightarrow a) = +\}
\]

\[
IN(a) = \sum_{b \in (b,a)\in E(G)} 1\{sign(b \rightarrow a) = -\}
\]

\[
OP(a) = \sum_{b \in (a,b)\in E(G)} 1\{sign(a \rightarrow b) = +\}
\]

\[
ON(a) = \sum_{b \in (a,b)\in E(G)} 1\{sign(a \rightarrow b) = -\}
\]

Using maximum likelihood estimation, we can get:

\[
ILB(a) = \frac{IN(a)}{IN(a) + IP(a)}
\]

\[
OLB(a) = \frac{ON(a)}{ON(a) + OP(a)}
\]

The histograms in Epinions are shown in Fig. 2.

When using local bias to do prediction, the sign to predict is unknown. So the previous formula should be slightly modified to eliminate the particular information. Two approaches can be adopted. One is to pretend that the objective edge does not exist. The other solution is to assign either positive or negative sign to the objective edge with equal probability. The second approach is proved by experiments to be able to generate better results.

B. SN-PageRank

Essentially, ILB and OLB can be regarded as an attempt to utilize the law of large numbers to estimate the relative trustiness between two nodes. A natural extension is to find out the absolute trustiness of nodes using global information. Resembling to background of PageRank algorithm, power iteration method seems to be applicable for this goal. We name it as SN-PageRank (PageRank for signed network).

We start with definitions. Define the absolute value of trustiness of a node i as \(t_i\), and the stochastic adjacency matrix \(M \in \mathbb{R}^{n \times n}\) where

\[
M_{ij} = \begin{cases} 
\text{sign}(i \rightarrow j) & \text{if } (i, j)\in E(G) \\
0 & \text{otherwise}
\end{cases}
\]

Define the random teleport rate \(1 - \beta\), we have the same iteration algorithm:

\[
t^{(k+1)} = \beta M \cdot t^{(k)} + (1 - \beta) \frac{1}{n} \cdot 1
\]

where \(1 = [1 \ 1 \ldots 1]^T\)

Unfortunately, the situation is different from the original algorithm since the entries of the stochastic adjacency matrix can be negative. This makes this algorithm problematic because the sum of a row might not be unitary anymore. Here we address this problem step by step:

Assume the original adjacent matrix is \(A \in \mathbb{R}^{n \times n}\) :

\[
A = \begin{pmatrix}
0 & sign(1, 2) & \ldots & sign(1, n) \\
\vdots & \vdots & \ddots & \vdots \\
sign(n, 1) & sign(n, 2) & \ldots & 0
\end{pmatrix}
\]

To remove the negative values without impacting the relative trustiness for each pair of nodes, we add 1 to all the entries of \(A\). Therefore, each negative edge is represented by a “0”, no connection is expressed as a “1”, and a positive edge is recorded as a “2”. Every link is increased by one which makes the relative value transmission unchanged.

Now we have modified adjacency matrix as:
To this point, the power iteration can theoretically work. However in real-world, that negative relationship is much less than the number of irrelevant pairs, which makes the original sparse matrix to be a dense one. Power iteration becomes computationally inefficient or even infeasible. So we make a further approximation without significant impact on the result. Our approach is cycle cancellation, in which the redundant cyclic paths are cancelled out. After removing negative edges, a huge number of positive edges and cycles are introduced. So trustiness will be transmitted back and forth through those cycles, which guarantees that the reciprocal relationship is only slightly changed. At the same time, this method eliminates extra linking information, making it more consistent with the original network. We can get a transferred vector as

\[ \mathbf{A} = \left( \begin{array}{cccc} 1 & \text{sign}(1,2) + 1 & \ldots & \text{sign}(1, n) + 1 \\ \text{sign}(2,1) + 1 & 1 & \ldots & \text{sign}(2, n) + 1 \\ \vdots & \vdots & \ddots & \vdots \\ \text{sign}(n, 1) + 1 & \text{sign}(n, 2) + 1 & \ldots & 1 \end{array} \right) \]

The histogram of SN-PageRank (SNPR) obtained using this approach for Epinions is shown in Fig. 3.

![Histogram of SNPR](image)

To validate this approximation approach, we separate the original network into a positive and a negative part. We then perform the power iteration for both graphs. Let \( T = (t_i) \) be the global trustiness for the positive network and \( D = (d_i) \) be the global distrust for the negative network. We can get a transferred vector as \( \mathbf{T}_D = T - D \). To make TD comparable with SNPR, the following region transformation is needed:

\[ (T_D)_i = \alpha \cdot (T_D)_i + \text{bias} \]

where

\[ \alpha = \frac{\max(SNPR) - \min(SNPR)}{\max(T) - \min(T)} \]

\[ \text{bias} = \min(SNPR) - \min(T) \]

The distribution of SNPR is a reasonable estimation of the global trustiness.

The distribution of TD is shown in Fig. 4.

From the two plots, we can see that they have similar distributions with certain bias. Therefore, SNPR is a reasonable estimation of the global trustiness.

![Histogram of TD](image)

C. Weighted Local Bias

For local bias proposed in part IV, our assumption is that each incoming positive edge is equally important in affecting a node’s local bias with incoming negative edge which is based on the law of large numbers. Hence, if the number of neighbors of a node is not large enough, the experimental results using OLB and ILB may not be satisfactory.

With the help of SNPR values, we can recalculate the local bias from a more global aspect by weighting the ratios with SNPR of that node. The following updates are made:

\[ IP(a) = \sum_{b \in \{(a,b) \in E(G)\}} SN_{PR}_b \cdot 1\{\text{sign}(b \to a) = +\} \]

\[ IN(a) = \sum_{b \in \{(a,b) \in E(G)\}} SN_{PR}_b \cdot 1\{\text{sign}(b \to a) = -\} \]

\[ OP(a) = \sum_{b \in \{(a,b) \in E(G)\}} SN_{PR}_a \cdot 1\{\text{sign}(a \to b) = +\} \]

\[ ON(a) = \sum_{b \in \{(a,b) \in E(G)\}} SN_{PR}_a \cdot 1\{\text{sign}(a \to b) = -\} \]

SNPR\(_b\) is the SNPR value for node b. By applying them to the formula of ILB and OLB, we can obtain the weighted local bias, which are named as WILB. The distribution of WILB is shown in Fig.5.

![Histogram of WILB](image)

V. EDGE SIGN PREDICTION

A. Supervised Method

In this paper, we use two different types of supervised classifiers to predict the edge sign: logistic regression classifier and SVM classifier.

1) Logistic regression

Logistic regression leans a model of the form:

\[ P(r e | x) = \frac{1}{1 + e^{-b_0 + \sum b_i x_i}} \]

In the prediction process, the sign of a given edge e is set to 1 if \( P(r e | x) > 0.5 \) and set to 0 otherwise.

2) SVM algorithm

We then apply the SVM algorithm to find a hyper-plane by solving the following optimization problem:

\[ \min_{\theta} \frac{1}{2} \| w \|^2 \]

\[ \text{s.t.} \ y^{(i)}(w^T x^{(i)} + b) \geq 1, \ i = 1, 2, \ldots, m \]

After successfully finding out the optimal hyper-plane, we are able to give every objective edge a sign according to its relative position with respect to the plane.
B. Feature Extraction

In the two machine learning algorithms, selection of features influences much on the final results. The whole set that we construct contains three kinds of features: commonly adopted dynamics such as node degree and clustering coefficients; extended variants of previous features, e.g. combination the idea of longer cycles in [5] and triangle patterns in [1]; modified version of existing algorithm, which can be successfully applied to signed networks, e.g. SNPR.

In the following parts, we perform a serious of experiments based on this feature set and make comparisons between related features. Also, to catch an idea about their relative significance in edge sign prediction, we also implement the feature selection algorithm.

C. Various Result

1) Prediction accuracy

We use logistic regression and SVM to train and test our model, following the experimental scheme in Guha’s paper and learning models on original as well as sampled datasets with 50% positive edges. The results under 10 fold cross validation is shown in Table IV below.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Epinions Baseline</th>
<th>Epinions Logistic</th>
<th>Epinions SVM</th>
<th>SlashDot Baseline</th>
<th>SlashDot Logistic</th>
<th>SlashDot SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangle Only</td>
<td>50%</td>
<td>83.46%</td>
<td>77.47%</td>
<td>50%</td>
<td>83.33%</td>
<td>77.47%</td>
</tr>
<tr>
<td>Triangle+Quadrangle</td>
<td>87.66%</td>
<td>85.59%</td>
<td>87.66%</td>
<td>93.62%</td>
<td>91.29%</td>
<td>93.62%</td>
</tr>
</tbody>
</table>

The results are encouraging, which successfully supports our ideas of quadrangle counting and SNPR estimation. From the table, we can arrive at the points below:

- When the size of dataset is not large enough, logistic regression generally works better than SVM.
- Quadrangle information can greatly improve the prediction accuracy. Although prediction accuracies when only considering triangles or quadrangles are almost the same, their information is complimentary.
- With SNPR, we can push our prediction accuracy to a quite high level. Our model can reach 91.38% on sampled Epinions dataset, which is significantly better than 86.7% from Guha’s paper. It is also better than Jure’s model on undivided data (i.e., $em = 0$).

2) False positive rate

Besides precision, recall is also a very important criterion in machine learning problems. In our datasets, positive edges is overwhelm larger than the number of negative edges. As a consequence, the false rate tends to be high. Using our model with all features, the false positive rates for the two datasets are 23.97% and 36.26%, both are much better than 44.4% and 50.7% reported in [5].

D. Feature Selection

After training, we have a well-calibrated model, which can be further used to analyze the importance of different features. Observing the weighting coefficients is the most widely used method and it requires carefully normalization on the dataset. However, this requirement cannot be easily fulfilled in our settings. Hence, we used another option, forward feature selection, to analyze the importance of features and simplify the model.

Feature selection process on Epinions dataset is shown in Fig. 6 below.

From the figure, we can see that OLB, ILB and SN-PageRank are important. Also, quadrangle information is also useful in edge sign prediction. Similarly, OLB, ILB and Quadrangle information stands out in process of feature selection on Slashdot dataset.

E. Cross Dataset Validation

Since online social networks share similar structures, we can generalize our model and fit it to the whole online social network. To evaluate this kind of generalization, we train our model on one dataset and test it one the other one. The generalization accuracy for the 50%-positive sampled data is shown in Table 5 below.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Epinions Baseline</th>
<th>Epinions Logistic</th>
<th>Epinions SVM</th>
<th>SlashDot Baseline</th>
<th>SlashDot Logistic</th>
<th>SlashDot SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangle Only</td>
<td>50%</td>
<td>83.46%</td>
<td>77.47%</td>
<td>50%</td>
<td>83.33%</td>
<td>77.47%</td>
</tr>
<tr>
<td>Triangle+Quadrangle</td>
<td>87.66%</td>
<td>85.59%</td>
<td>87.66%</td>
<td>93.62%</td>
<td>91.29%</td>
<td>93.62%</td>
</tr>
</tbody>
</table>

The results are encouraging, which successfully supports our ideas of quadrangle counting and SNPR estimation. From the table, we can arrive at the points below:

- When the size of dataset is not large enough, logistic regression generally works better than SVM.
- Quadrangle information can greatly improve the prediction accuracy. Although prediction accuracies when only considering triangles or quadrangles are almost the same, their information is complimentary.
- With SNPR, we can push our prediction accuracy to a quite high level. Our model can reach 91.38% on sampled Epinions dataset, which is significantly better than 86.7% from Guha’s paper. It is also better than Jure’s model on undivided data (i.e., $em = 0$).

F. Conclusion

In this paper, we investigated several different sorts of characteristics of signed networks, including some commonly used features, extension of triangle patterns, local descriptors and a modified version of PageRank value. By feeding some of the selected features to the edge sign prediction model, we successfully improve the
performance of previous work to a large extent, in terms of both the prediction accuracy and the false positive rate. By conducting the feature selection algorithm, we find that the local bias, quadrangle and the SN-PageRank can make some contribution in predicting edge signs.

There are a number of further directions can be investigated starting from this paper. The first one is to explore other creative and effective methods that might yield still better performance than the edge sign prediction model used in this work. And then, based on current feature set, it is possible to explore more representative features or descriptors that might work well specifically for signed networks. Based on the results, it is fairly valuable to combine them with the social theories of signed links and build up a thorough understanding on why the features are significant. Another important aspect is that, as is mentioned in part IV, the meaning of edge signs varies much in different social network; it is worthy to explore our methods and the extensions using other datasets.

REFERENCES


Tongda Zhang was born in China, 1989. He graduated with the outstanding graduate award from Tsinghua University in 2011, and earned his Bachelor’s degree of Automation. He is currently a GRADUATE STUDENT in Electrical Engineering at Stanford University with a focus of data mining, optimization and Human Computer Interaction.

Zhouxiao Bao was born in China, 1991. She received her bachelor degree in electronic information engineering from University of Science and Technology of China in 2011. In 2012, she received her Master’s degree in electrical engineering from Stanford University.

She currently works for Google Inc. as a SOFTWARE ENGINEER in California.

Haomiao Jiang was born in China, 1989. He earned bachelor degree in computer science from Shanghai Jiao Tong University in 2011. He got his master degree from department of electrical engineering at Stanford University.

Mr. Jiang acted as engineering INTERN at Intel and Qualcomm, concentrating on algorithm and software development for gesture recognition and image processing.

Yingfeng Zhang was born in China, 1979. He received his Dr. degree in mechanical engineering from Xi’an Jiaotong University of China in the year of 2005.

He currently is a Professor in Northwestern Polytechnical University, China. His research interest is data mining for production process.