Link and Sign Prediction in Signed Networks

CSE291-F Final Project Report

Saicharan Duppati
A53221873
sduppati@ucsd.edu

Håvard Pettersson
U07322227
ax004536@acsmail.ucsd.edu

Shiwei Song
A53206591
shs163@eng.ucsd.edu

Swapnil Taneja
A53219137
swtaneja@eng.ucsd.edu

ABSTRACT

This paper presents the problem of Link Prediction specific to signed networks and proposes solutions that may be used. Link prediction is the task of predicting the potential links that can be formed in graphs over time. While the typical algorithms use a variety of feature selection techniques to explore this question, we want to explore the problem only given the structure of a social graph network, specifically signed social networks.

In specific, in this paper we will study, analyze and compare a number of methodologies that have been proposed in specific to the task of sign/link prediction in signed networks.

KEYWORDS

link prediction, signed networks

1 MOTIVATION

Link prediction is naturally occurring problem in fields of social networks, e-commerce and bioinformatics. Social networks like Facebook have a feature for suggesting new friends, and although they use a variety of features besides just the graph structure, the underlying task of link prediction is important. In the field of e-commerce, link prediction is used to predict and estimate the product demand from the users. In bioinformatics, where it is near impossible to test every combination of proteins, link prediction can help predicting which proteins interact and which don't.

Besides the importance of the link prediction task alone, studying link prediction specific to signed networks has only picked its importance recently and is still relatively uncommonly studied. We believe it is important to include the negative edges as well as the positive in an analysis like link prediction as they are crucial for the stability and structure of the graph.

2 PROBLEM DEFINITION

Link prediction is the problem of predicting non-existing edges between nodes given the current state of the graph.

Signed Networks are the networks with both positive and negative edges. The reasons behind the sign can vary among applications. For example, positive edges could represent the relationship of friend and negative edges could represent the relationship of foe in social networks.

To be more precise, we focus on the link prediction problem in directed, unweighted, signed graph. We formulate the problem as follow. Given a graph $G = \langle V, E, \Sigma \rangle$ at time t where V and E are the sets of nodes and edges in G and Σ is a mapping: $E \rightarrow \{+1, -1\}$, we want to perform two tasks on G:

(1) predict the sign of potential edges (sign prediction)
(2) predict edges that are likely to be formed (link prediction)

3 RELATED WORKS

Classical link prediction is the problem of predicting the existence of a link between two entities (could be treated as a positive link), based on the attributes of the objects and other observed links [6]. In [12], Liben-Nowell and Kleinberg suggested methods based on similarity of nodes based on both node and path properties. The predicting task, such as predicting trust, distrust, friendship, co-authorship and other relationships, could be represented as predicting the link's value in SSNs (Signed Social Networks).

A lot of studies on trust and friendship prediction have been done based on websites that allow users to show opinions to others' contents and comments, such as Epinions, eBay, Wikipedia, Essembly and Slashdot. Guha et al. develop a formal framework of trust propagation schemes and introduce the computational treatment of distrust propagation [7]. Massa and Avesani use the Mole Trust metric, which reduces the prediction error for controversial users, to predict trust between users in Epinions [15].

Burke and Kraut present a model of the behavior of candidates for promotion to administrator status in Wikipedia [2].

Researchers have also worked on signed networks. Kunegis et al. analysed a Slashdot Zoo corpus of about 80 000 nodes and 510 000 edges with signed edges, and discussed properties including clustering, popularity, centrality, and link sign prediction [9]. In [4], Chiang et al. proposed both unsupervised and supervised method based on the measure of social imbalance to do sign prediction.

Recently, researchers started to connect the link prediction problem with social psychology and got good results. Leskovec et al. investigated balance and status theories of SSNs in [11]. They use logistic regression model to predict links' values in signed networks and connected this to the balance and status theory .

Liu et al. achieved community mining by using the links' sign values in an SSN in [14]. Symeonidis and Tiakas use transitive node similarity for predicting and recommending links in an SSN, and they propose the FriendTNS± method, which takes both positive and negative links into account when calculating two connected users' similarity [17].

The above studies mainly focused on computing social members' similarity by different metrics or showing that social psychology

also works in SSNs. For our project, we will work on data from typical social network datasets, Slashdot.

Based on RBMs, Hinton and Salakhutdinov try to represent high-dimensional input vectors by low-dimensional codes [8]. For our project, we use DBNs as one of our possible methodology for performing signed link prediction.

Different from the above research areas, an SSN is a typical complex network, whose nodes' degrees accord with the complex network's degree power law distribution. However, seldom has deep learning research been done on such a kind of data. In this paper, our study focuses on replicating novel methods and improving on them as well as building the proper deep learning models for solving link prediction in datasets from SSNs.

4 METHODOLOGY

Since the majority of the work done in this area is pretty recent, we will be exploring a number of methods. We will be exploring the solutions to the problems of both link prediction and sign prediction. However, we will mainly be focusing on the sign prediction problem in this paper.

4.1 Methods Based on Measure of Social Imbalance

The following method is based on Chiang et al.'s paper on link prediction in signed networks [4].

A signed graph is balanced iff there are no simple cycles with an odd number of negative edges. Based on this theorem, we can define a measure of the graph's imbalance as

$$\mu_k(G) := \sum_{i=3}^{k} \beta_i \sum_{\sigma \in SC_i(G)} \mathbf{1}[\sigma \text{ is unbalanced}]$$

where $SC_i(G)$ is the set of all simple cycles with length i and β_i is an coefficient to weight the contribution of unbalanced simple cycles of different lengths. k represents the maximum order of cycles to be considered.

We will use this definition to predict the signs of edges. In particular, to predict the sign of an edge, we will assign to it the sign that will result in the most balanced graph. We form two new graphs, $G^{+(i,j)}$ and $G^{-(i,j)}$, which are simply G with a positive and negative edge added. Predicting the sign $\Sigma(e)$ of edge $e = (i, j)$ now amounts to scoring the two graphs with $\mu_k(G)$. We let

$$\Sigma(e) = \text{sign}\left(\mu\left(G^{-(i,j)}\right) - \mu\left(G^{+(i,j)}\right)\right).$$

This is, however, inefficient to compute. In fact, finding $SC_i(G)$ is NP-hard. Because finding $SC_i(G)$ is intractable, we instead use $C_i(G)$, i.e. *all* cycles of the graph, not only simple cycles.

Next, as found by Chiang et al., predicting the sign of an edge (i, j) with this method is equivalent to finding the (i, j) entry in the adjacency matrix A of G. This is the method we use, and the full result of Chiang et al. is

$$\text{sign}\left(\mu\left(G^{-(i,j)}\right) - \mu\left(G^{+(i,j)}\right)\right) = \text{sign}\left(\sum_{t=3}^{k} \beta_t A_{i,j}^{t-1}\right).$$

Note that under this definition, it is possible that the sum of adjacency matrix elements is zero, and does not have a sign. When this is the case, we can not meaningfully say anything about the sign of the edge using this method alone. Increasing k, the order of cycles considered, reduces the number of indeterminate edges.

4.2 Jaccard's similarity for Signed networks

Similar to Jaccard's similarity for the unsigned networks, we can define the similarity using the neighbors on the same lines for signed networks as well. All we need to be careful about is to make sure that we treat negative and positive links separately when considering neighbors.

The similarity between user i and user j can be broken down into a vector of three values as follows:

$$S(i,j) = [\frac{|F_i \cap F_j|}{|N_i \cup N_j|}, \frac{|E_i \cap E_j|}{|N_i \cup N_j|}, \frac{|(F_i \cap E_j) \cup (E_i \cap F_j)|}{|N_i \cup N_j|}]$$

***The symbol |x| represents the cardinality of set x.*
In the above formula F_i and E_i stands for the friends (users with a positive edge to or from) and foes (users with a positive edge to or from) of user i and $N_i = F_i \cup E_i$.

4.3 Users Reputation and Optimism

The sign of the link from user i to user j is dependent on how likely that the user i on her/his own makes positive/negative connections (*optimism*) and how often does the user j on his/her own is to get positive/negative connection from other users (*reputation*). Below are the formula by [16] to quantify these notions:

$$RBR_i = \frac{|R_{in}^{(+)}(i)| - |R_{in}^{(-)}(i)|}{|R_{in}^{(+)}(i)| + |R_{in}^{(-)}(i)|}$$

$$RBO_i = \frac{|R_{out}^{(+)}(i)| - |R_{out}^{(-)}(i)|}{|R_{out}^{(+)}(i)| + |R_{out}^{(-)}(i)|}$$

Where RBR_i and RBO_i stands for Rank Based Reputation and Rank Based Optimism of the user i. In the formula, $R_{in}^{(+)}(i)$ and $R_{in}^{(-)}(i)$ refer to the sum of the rank values (discussed later) of the nodes that have positive and negative links to user i. Similarly $R_{out}^{(+)}(i)$ and $R_{out}^{(-)}(i)$ as the sum of the rank values of the nodes that have user i have positive and negative links to. Rank of each node can be Page rank, HITS or just a constant. We will later see that just consider rank as a constant. (ie., $R_{in}^{(+)}(i)$ and $R_{in}^{(-)}(i)$ as the number of the positive and input links to user i gives the best performance on our chosen data)

4.4 Community Detection

The task of sign prediction can greatly be influenced if we know what communities that user i and user j belong to. However, the basis for forming a community in the case of signed network differs from the community paradigm of unsigned networks. In the case of unsigned networks, we are concerned with finding the communities that maximize the number of edges within the community.

4.4.1 Balanced Signed network: A network $\boldsymbol{G}$ that can be partitioned into two more subsets (communities) such that every positive arc joins units of the same subset and every negative arc joins units of different subset.

If our data correspond to a balanced signed network and given the algorithm to find such communities, our link prediction problem will be a completely deterministic task. Unfortunately, real life graphs rarely follow the above definition. However, by using the above definition of *balanced signed network* to following objective cost function can be constructed which when minimized gives communities[5].

$$F(C_1, ...C_k) = N + P$$

Where N indicates the total number of negative links within each community C_i and P indicates the total number of positive links across communities.

Solving for the above objective function reduces to a spectral approach according to [1] ie., we will end up solving for s which maximizes $s^T As$ where A is the adjacency matrix of the graph G. We followed this spectral approach for community detection.

4.5 FriendTNS

Symeonidis and Tiakas [17] define FriendTNS (transistive node similarity) and its extension to signed networks is based on a simple similarity measure between neighboring nodes. This similarity is based on the Jaccard coefficient, and is defined as

$$sim\left(v_i, v_j\right) = \frac{\boldsymbol{r_i} \cdot \boldsymbol{r_j}}{||\boldsymbol{r_i}||^2 + ||\boldsymbol{r_j}||^2 - \boldsymbol{r_i} \cdot \boldsymbol{r_j}}.$$

By Theorem 1 of [17], this is equivalent to

$$sim\left(v_i, v_j\right) = \frac{1}{deg\left(v_i\right) + deg\left(v_j\right) - 1}$$

where v_i and v_j are neighbors.

Symeonidis and Tiakas further extend this to arbitrary connected nodes in a graph by defining an extended similarity measure, which is the product of all neighbor similarities in the shortest path between two nodes. They define the extended similarity as

$$esim\left(v_i, v_j\right) = \begin{cases} 0 & \text{no path between } v_i \text{ and } v_j \\ sim\left(v_i, v_j\right) & v_i \text{ and } v_j \text{ are neighbors} \\ \prod_{h=1}^{k} sim\left(v_{p_h}, v_{p_{h+1}}\right) & \text{otherwise} \end{cases}$$

where $v_{p_1} = v_i$ and $v_{p_{h+1}} = v_j$, and intermediate v_{p_h} are the nodes in the shortest path between v_i and v_j.

Finally, the authors provide an alternative definition of $sim\left(v_i, v_j\right)$ for use with signed networks. This definition is based on status theory [10, 11]. The signed variant is

$$sim\left(v_i, v_j\right) = \frac{1}{\sigma\left(v_i\right) + \sigma\left(v_j\right) - 1}$$

where $\sigma\left(v\right) = deg^+_{in}\left(v\right) + deg^-_{out}\left(v\right) - deg^+_{out}\left(v\right) - deg^-_{in}\left(v\right)$ and $deg^+_{in}\left(v\right)$ represents the incoming positive links for node v, and so on.

The similarity measure defined falls in the range $[0, 1]$, with 1 being the highest similarity and 0 being the lowest. By calculating the similarity between all nodes in a graph, the highest similarity scores can be used to predict new links.

4.6 Methods Based on Logistic Regression

For the sign prediction task, we can regard it as a binary classification problem. Hence, we can use logistic regression to predict the sign.

$$P(\Sigma(u, v) = +1) = \frac{1}{1 + e^{-(b_0 + \sum b_i x_i)}}$$

The features we used can be divided into two types. The first type of features is the basic graph information like node degree [10]. For an edge $e = (u, v)$, we count the positive out degree $d^+_{out}(u)$. Similarly, we have $d^-_{out}(u)$, $d^+_{in}(v)$ and $d^-_{in}(v)$. We also use the number of common neighbors of u and v as another feature.

The second type of features make use of the balance characteristic of social networks. As proposed in [10], We count the number of triads involving the edge $e = (u, v)$. Since there are two possible signs and directions between w and u, v, there are 16 possible triads in total. We count the number of each possible triads and we get a 16-dimensional feature. We also tested if we ignore the direction of the edges. We will get a 3-dimensional feature in this way. According to [18], the social imbalance features still has a high chance to hold for loops of length four. Our analysis on the dataset proved this (5.1). So we take the same approach to quadrangles. There are 4 possible sign combination for the three edges. So we will get a 4-dimensional feature.

Further to the above features, we also used the social imbalance measure, Jaccard's similarity, users reputation and optimism, communities and FriendTNS as mentioned in the above sections as features.

Since logistic regression gives us a probability, we can extend this to do link prediction. We choose a threshold θ. Then we predict the link as follow

$$L(u, v) = \begin{cases} -1 & P(\Sigma(u, v) = -1) > \theta, \\ +1 & P(\Sigma(u, v) = +1) > \theta, \\ 0 & \text{otherwise} \end{cases}$$

4.7 Neural Networks based feature transformation and sign prediction

In [13] authors conducted a study by transforming features using stacked RBMs (Restricted Boltzmann Machines) on SSNs. The preprocessing involved extracting features that are categorized into two classes - one contains the features based on the node's self-degrees, such as in-degree and out-degree; the other class contains the features based on the node's interactions with its neighbors, such as the common neighbor number and the number of neighbors who share certain opinions. These features are the inputs to the Neural Network.

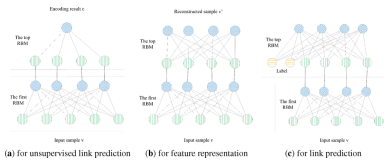

Figure 1

Figure 1 shows the stacked RBMs used in [13] for unsupervised as well supervised approaches. In their paper, their task was to predict the sign of a link. We created a neural structure with the architecture described in this section. We created a 26 length vector input to

the neural net. Each vector value corresponds to a new feature. The features we are considering include degree (for outgoing as well as incoming edges each with either a positive or negative sign).They count for 8 features. The next two features are common neighbors based on edges and nodes. Set intersection of neighbors of u and neighbors of v. While counting by nodes, we just count one node only once. While counting by edges, we count the number of edges that connect to a common node. So if u connects to w with 2 edges and w connects to v with 3 edges, we keep 5 as a feature value for common neighbor by edges. The next features are common neighbors having certain property. For example, if there is a positive edge from u to w and a positive edge from w to v , we get one feature for number of nodes having this property. It means we can have 16 such combinations. So total, we have 8 + 2 + 16 = 26 features. We have created samples from the graph data and stored these 26 length vectored data into our file. We built a rbm network and trained it . Let's call this rbm1. rbm1 has 27 visible units and 27 hidden units. One extra unit for the sign (+1 and -1) . For the evaluation section we do not consider sign unit. We train this until convergence and encode our visible vectors. We take the activations and persist it. These activations are then used as inputs to rbm2. rbm2 has 27 visible units and 27 hidden units again. Rmb3 has 27 visible and 2 hidden units. Again for evaluation , we only consider two stacked RBMs. But for the diagrams below we used three stacked RBMs. In order to see if we are heading the right direction we try to linearly separate the results. We do not succeed in doing so without the sign unit. That is we try to plot the activation of first unit with respect to the other . We expect the plot to have clear clusters . So, to get the linear separation we add the sign unit to hold +1 for positive samples and -1 for negative samples. We get the following figure 2 -

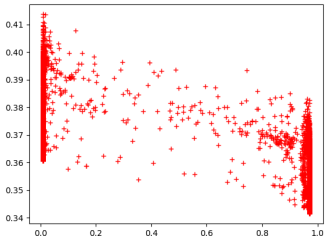

Figure 2

We can see that there is clear separation of data . Now , since -1 and 1 can pretty clearly separate data due to sign , we tried to add 0 for the sign unit instead of -1 for negative samples. We get the following figure -

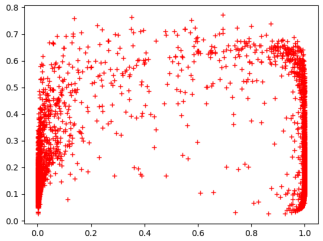

Figure 3

The challenge with this approach is that we can not run additional classifiers on top of these encoders because they require sign as inputs . In the evaluation section , we present the results of the additional classifiers that run on top of the encoded activations and without activations. We clearly observe a decline in the accuracy but the Adaboost classifier gives close results. . We observed above from the figures that the data was not linearly separable without the sign unit. To run additional classifiers we only took 2 stacked RBMs and 26 units.

5 EVALUATION

5.1 Basic Analysis on Dataset

The dataset we will use is the Slashdot Zoo dataset from the Stanford Network Analysis Project (SNAP) [11]. Slashdot allows users to tag other users as "friends" or "foes", representing positively and negatively signed edges. The dataset contains the directed signed relationships between users at 3 different timestamps: November 6th 2008 and February 16th and 21st 2009.

	Slashdot081106	Slashdot090216	Slashdot090221
# nodes	77,350	81,867	82,140
# edges	516,575	545,671	549,202
pos edges	76.7%	77.4%	77.4%
neg edges	23.3%	22.6%	22.6%

Table 1: statistics of dataset at three timestamps

Table 1 shows some basic statistics of the dataset at the three timestamps. There are about 80000 nodes and 500000 edges in the graph. Fig 4 shows the number of positive and negative edges in the dataset. We can see the number of positive edges is about 3 times the number of negative edges.

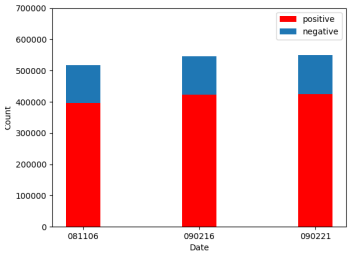

Figure 4: Edge distribution in the dataset

Another interesting finding is about the existence of inverse relation pairs. We found 1888, 2006 and 1949 pairs of nodes that have positive edge in one direction but negative edge in another direction in the three timestamps.

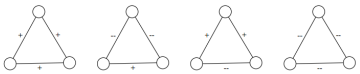

Figure 5: Four types of triangles

As mentioned above the triangles are a good indication for some of the features like social balance in network. There are four kinds of triangles as shown in Fig 5. Based on social balance theory, the first two types are balanced and the latter two are imbalanced. We counted the number of four types of triangles in the dataset. The results are shown in Table 2. As we can see the number of the balanced triangles are much more than the number of imbalanced ones which meet the theory of social balance.

	Slashdot081106	Slashdot090216	Slashdot090221
+++ (bal)	791,246	829,912	837,616
- -+ (bal)	150,580	153,718	161,162
-++ (imb)	130,646	133,358	135,390
- - - (imb)	23,636	24,150	24,962
balance	85.9%	86.2%	86.2%

Table 2: Number of four types of triangles

According to [3], [18], the social balance feature should also hold for longer cycles in the social network graph. So we also checked the quadrangles. The results are shown in Table 3. As we can see, the number of balanced quads are much more than the number of imbalanced ones. We make use of this observation in our work.

	Slashdot081106	Slashdot090216	Slashdot090221
++++ (bal)	141,806,354	157,238,490	158,671,548
-+++ (imb)	15,238,012	15,755,962	16,020,734
- -++ (bal)	38,512,233	39,747,033	40,667,090
- - -+ (imb)	3,644,898	3,739,780	3,884,650
- - - - (bal)	12,737,332	13,611,337	13,947,380
balance	91.1%	91.5%	91.5%

Table 3: Number of five types of quads

5.2 Sign Prediction

For the task of sign prediction, we will begin with the full graph and remove the signs of some edges and use our algorithms to predict the removed signs. Using these results, we can compare the predictions with the ground truth to calculate the accuracy and F1-score to evaluate our algorithms. We use the Slashdot dataset at three timestamps and run 10-fold cross validation to get all the results.

S081106		S090216		S090221	
acc	F1	acc	F1	acc	F1
0.862	0.911	0.866	0.914	0.866	0.918

Table 4: Sign prediction results on Slashdot datasets

Table 4 shows the results on the Slashdot datasets for sign prediction. The accuracy is about 86% and F1-score is about 0.91.

5.2.1 Logistic regression. For the logistic regression, we tried with different features. Table 5 shows the results using single feature. Using only degree features gives us a decent performance. The directed and undirected version of triads give similar results. So we use the undirected with a lower computation consumption. Adding quads features improve the performance a little while adding social imbalance features has no improvement.

feature	accuracy	F1-score
degree	0.830	0.896
triads (undirected)	0.791	0.879
triads (directed)	0.797	0.881
quads	0.840	0.904
social imbalance	0.799	0.881
Jaccard's	0.784	0.877
Optimism/Reputation	**0.859**	**0.910**
Community	0.775	0.873
FriendTNS	0.773	0.871

Table 5: Logistic regression using single features for sign prediction

We also looked into the coefficients fitted by the model. We found that the coefficients for feature of number of balanced triads are positive and the coefficients for feature of number of imbalanced triads are negative in general (14 out of 16). Again, this provides more evidence for the social balance theory.

5.2.2 Optimism/Reputation. We see that optimism/reputation is the feature that works the best with the dataset. Therefore we also analyze the various ranking methods $R^{(+,-)}_{out,in}$ to compute these optimism and reputation

ranking method	accuracy	F1-score
1 (#count)	**0.859**	**0.910**
pageRank (undirected)	0.791	0.879
HITS (undirected)	0.808	0.889

Table 6: Accuracy vs Optimism/Reputation ranking method

We can see from the table 6 that the simplest ranking algorithm of the three gives the best performance on our data.

5.2.3 Running Classifiers on top of 26 features. In the Neural Networks section we described the 26 features we extract . We run the following classifiers on top of these features and make predictions for sign . The results are shown below in the table 7-

Classifier	accuracy	Max	Min
SVM	0.8177	0.8318	0.8006
logistic	0.8108	0.82	0.795
forest	0.765	0.8094	0.6893
Gradient Boosting	0.80955	0.8418	0.7856
Adaboost	0.773055	0.7981	0.7437

Table 7: Sign prediction accuracy without encoding

Since, we are training on fraction 0.9 of the data and 0.1 is for testing, we get variation of results for each classifier (split is random and hence the results may vary). We take average over 5 executions of the program and then show the results. The table also shows maximum and minimum accuracy achieved over these executions of the program. The table 8 below shows the Accuracy after encoding with two stacked RBMs (26 , 26, 2) trained on 15 epochs .

Classifier	accuracy
SVM	0.526
logistic	0.666
Gradient Boosting	0.5525
Adaboost	0.695

Table 8: Sign prediction accuracy with encoding

The table 9 below shows the Accuracy after encoding with two stacked RBMs (26 , 26, 2) trained on 30 epochs .

Classifier	accuracy
SVM	0.685
logistic	0.6931
Gradient Boosting	0.70125
Adaboost	0.7806

Table 9: Sign prediction accuracy with encoding

We see that as we increase the number of epochs, our accuracy increases. And also, Adaboost classifier gives the best accuracy among all other classifiers. For 50 epochs AdaBoost Classifier , we get the accuracy of . We expect the result to surpass on increasing the number of epochs of training.

5.3 Link Prediction

Further, we extend the sign prediction problem into a link prediction problem. Now the problem is given an edge, predict it's sign (-1/0/+1) where 0 stands for no (unlikely) link.

Since we have the Slashdot datasets for three timestamps, our original plan was to predict new links in the later dataset. However, we discovered that the nodeID in different timestamps are assigned differently. So we have to synthesize some edges to test for the link prediction problem.

For this task, we test on the first dataset. First, we randomly select 50,000 edges from the existing edges. Then we randomly generate 50,000 edges that doesn't exist in the original graph (in both direction). We expect our algorithm to predict 0 on these ones. We did the above test 10 times to get the average accuracy rate.

As explained in 4.6, we chose a threshold on the probability calculated by logistic regression to predict the link and the sign. Figure 6 shows the accuracy under different threshold θ. As we can see in the graph, there's a steep jump from $\theta = 0.8$ to $\theta = 0.85$. The accuracy improves significantly. So $\theta \approx 0.85$ is a good separating point for existing links and unknown links. We reached the best accuracy 0.773 at $\theta = 0.9$.

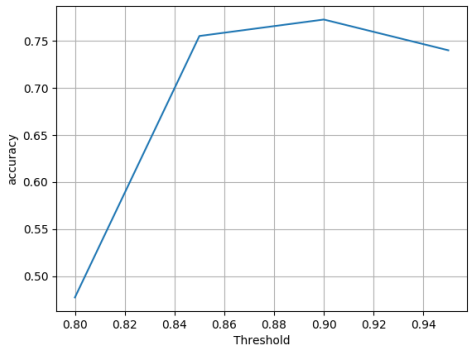

Figure 6: Accuracy under different threshold

6 CONCLUSIONS

In the project, we studied the problem of sign prediction and link prediction in directed signed social networks. We presented various methods that have been used and also devised a few of our own to improve the predictions. Through the project, we have also realized the importance of choosing the features than fine tuning the parameters of the regression model.

We tested our algorithms on the Slashdot datasets. We reached 86.6% accuracy and 0.91 F1-score for sign prediction. Similarly, we also achieved 77% accuracy for link prediction (Slashdot090221).

7 ACKNOWLEDGEMENTS

We would like to thank Prof. Fragkiskos D. Malliaros and the teaching assistant(s) for their dedication and support in helping us understand the material for the course.

REFERENCES

[1] Pranay Anchuri and Malik Magdon-Ismail. 2012. Communities and Balance in Signed Networks: A Spectral Approach. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012) (ASONAM '12)*. IEEE Computer Society, Washington, DC, USA, 235–242. DOI: http://dx.doi.org/10.1109/ASONAM.2012.48

[2] Moira Burke and Robert Kraut. 2008. Mopping up: modeling wikipedia promotion decisions. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*. 27–36. http://dl.acm.org.proxy.lib.umich.edu/citation.cfm?id=1460571

[3] Kai-Yang Chiang, Cho-Jui Hsieh, Nagarajan Natarajan, Ambuj Tewari, and Inderjit S. Dhillon. 2013. Prediction and Clustering in Signed Networks: A Local to Global Perspective. *CoRR* abs/1302.5145 (2013). http://arxiv.org/abs/1302.5145

[4] Kai-Yang Chiang, Nagarajan Natarajan, Ambuj Tewari, and Inderjit S. Dhillon. 2011. Exploiting Longer Cycles for Link Prediction in Signed Networks. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM '11)*. ACM, New York, NY, USA, 1157–1162. DOI: http://dx.doi.org/10.1145/2063576.2063742

[5] Patrick Doreian and Andrej Mrvar. 2009. Partitioning signed social networks. *Social Networks* 31, 1 (2009), 1 – 11. DOI: http://dx.doi.org/10.1016/j.socnet.2008.08.001

[6] Lise Getoor and Christopher P. Diehl. 2005. Link Mining: A Survey. *SIGKDD Explor. Newsl.* 7, 2 (Dec. 2005), 3–12. DOI: http://dx.doi.org/10.1145/1117454.1117456

[7] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2004. Propagation of Trust and Distrust. In *Proceedings of the 13th International Conference on World Wide Web (WWW '04)*. ACM, New York, NY, USA, 403–412. DOI: http://dx.doi.org/10.1145/988672.988727

[8] G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 5786 (2006), 504–507. DOI: http://dx.doi.org/10.1126/science.1127647

[9] Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. 2009. The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*. ACM, 741–750.

[10] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*. ACM, 641–650.

[11] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 1361–1370.

[12] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *journal of the Association for Information Science and Technology* 58, 7 (2007), 1019–1031.

[13] Feng Liu, Bingquan Liu, Chengjie Sun, Ming Liu, and Xiaolong Wang. 2015. Deep Belief Network-Based Approaches for Link Prediction in Signed Social Networks. *Entropy* 17, 4 (2015), 2140–2169. DOI: http://dx.doi.org/10.3390/e17042140

[14] Jiming Liu, Bo Yang, and William Cheung. 2007. Community Mining from Signed Social Networks. *IEEE Transactions on Knowledge and Data Engineering* 19 (2007), 1333–1348. DOI: http://dx.doi.org/doi.ieeecomputersociety.org/10.1109/TKDE.2007.1061

[15] Paolo Massa and Paolo Avesani. 2005. Controversial Users Demand Local Trust Metrics: An Experimental Study on Epinions.Com Community. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 1 (AAAI'05)*. AAAI Press, 121–126. http://dl.acm.org/citation.cfm?id=1619332.1619354

[16] Mohsen Shahriari, Omid Askari Sichani, Joobin Gharibshah, and Mahdi Jalili. 2016. Sign prediction in social networks based on users reputation and optimism. *Social Network Analysis and Mining* 6 (2016), 1–16.

[17] Panagiotis Symeonidis and Eleftherios Tiakas. 2014. Transitive node similarity: predicting and recommending links in signed social networks. *World Wide Web* 17, 4 (2014), 743–776. DOI: http://dx.doi.org/10.1007/s11280-013-0228-2

[18] Tongda Zhang, Haomiao Jiang, Zhouxiao Bao, and Yingfeng Zhang. 2013. Characterization and edge sign prediction in signed networks. *Journal of Industrial and Intelligent Information Vol* 1, 1 (2013).