# Edge prediction in Social Signed Networks

*Department of Computer Science, EPFL, Switzerland*
Paul Moinville, Alexandre Di Piazza, Arman Ehsasi

***Abstract*—This paper proposes two approaches for edge prediction in social signed networks. The first method takes into account the type of triadic structures found between 3 nodes throughout the network. The other approach does not consider triad types, but rather works with averaging edge weights. We find out that our first prediction algorithm does not always outperform baseline prediction scores. However, the second method does outperform the baselines by a substantial margin with respect to binary classification metrics. Additionally, we investigate how tuning parameters increase our prediction scores and propose how future work could additionally improve results for the techniques derived in this paper.**

## I. Introduction

Directed signed networks are implemented as a popular method to analyse the interactions between users on social media platforms. These networks feature nodes (users) with directed links between each other (interactions), which carry an attributed weight that characterizes the sentiment of an interaction. In a signed network using binary weights, one would attribute a weight of "+1" for a positive interaction and "-1" for a negative interaction. One thing to keep in mind is that the definition of "interaction" varies in different contexts and is subject to the type of social network and could be defined as "trust" or "degree of agreement" between two nodes.

A major aspect of signed network analysis is the capability of predicting unknown or missing edge weights between nodes. To this effect, many prediction methods are based on analysing different types of triads of a network, which in fact are an important characteristic in signed networks. The following work, offers an approach for local edge prediction based on "Balance Theory", used to analyse network triads which we call the "Triad Based Method", as well as another approach that we named the "Average Edge" prediction method, which is not based on the triadic structure of the network.

## II. Related Work

Our creative extension is based on the paper "Signed Networks in Social Media" [1], where the authors investigate how signed interactions between users influence network structures on online social platforms. They conduct their analyses and edge predictions by implementing theories from social psychology to signed network theory, namely "Balance Theory" and "Status Theory" [1], which revolve around the analyses of network triads. "Balance Theory" (in this case weak structural balance) explains that certain types of undirected network triads (T3:+++,T1:+ - -,T0:- - -) are balanced and others are not balanced (T2:++-). This means that in the signed network data we should find an overrepresentation of balanced triads, because these types are observed more frequently in society from a psychology viewpoint. For our extension project, we first use the idea of "Balance Theory" and formulate our own approach towards edge prediction using the fractions of balanced and unbalanced triads. On the other hand, we will observe how our other method performs on edge prediction, which is based on analysing all of the edges between 2 nodes as opposed to a triad.

## III. Datasets

We will use the same data as featured in [1] namely, the "epinions", "slashdot", and "wikipedia" data sets. These sets feature positive and negative binary weights and their statistics are summarized in Table I.

| | Epinions | Slashdot | Wikipedia |
|---|---|---|---|
| Nodes | 131828 | 82140 | 7118 |
| Edges | 841372 | 549202 | 103747 |
| p(+edges) in % | 85.3 | 77.4 | 78.8 |
| p(−edges) in % | 14.7 | 22.6 | 21.2 |
| $p(T3), p(T2)$ | 0.872; 0.069 | 0.84; 0.072 | 0.702; 0.207 |
| $p(T1), p(T0)$ | 0.052; 0,007 | 0.077; 0.011 | 0.080; 0.011 |

Table I
STATISTICS FOR DATASETS

## IV. Methods

We will now cover the details behind the implemented prediction algorithms.

### A. Baseline Predictions

To establish a baseline comparison we perform the following 2 trivial classification techniques, where their prediction results can be found in II.

1) Assign $+1$ to each edge
2) Assign $+1/-1$ according to the fraction of pos. and neg. edges

| Method | Metric | Epinions | Slashdot | Wikipedia |
|---|---|---|---|---|
| Base. 1 | Specificity | 0 | 0 | 0 |
| | Accuracy | 0.88 | 0.78 | 0.8 |
| | MCC | 0 | 0 | 0 |
| Base. 2 | Specificity | 0.15 | 0.23 | 0.21 |
| | Accuracy | 0.76 | 0.65 | 0.68 |
| | MCC | 0 | 0 | 0 |

Table II
BASELINE PREDICTION SCORES

### B. "Triad based" Method

For our first method we take a partition of our data set (20%), where we assume that we do not know the weight. Next, we compute the fractions of the different types of triads $(p(T_3), p(T_2), p(T_1), p(T_0))$ of the remaining data set (80%). For an edge where we don't know the sign, we first compute how many triads this edge will create when adding it to the known graph. For each triad that this edge will create, we compute all the signs of the partial triad (we know the signs of the two other edges of the triad that the new unknown edge will create) . The partial triads can have a (++), (+-) or (- -) configuration. Based on this sign and the percentages of each triad type Ti on the known graph (80%), we can thus compute the following conditional probabilities:

**Case 1:**
$p(T_3|++) = \frac{p(T_3)}{p(T_2)+p(T_3)}$ = prob. of assigning +1 to edge
$p(T_2|++) = \frac{p(T_2)}{p(T_2)+p(T_3)}$ = prob. of assigning -1 to edge
**Case 2:**
$p(T_2|+-) = \frac{p(T_2)}{p(T_1)+p(T_2)}$ = prob. of assigning +1 to edge
$p(T_1|+-) = \frac{p(T_1)}{p(T_1)+p(T_2)}$ = prob. of assigning -1 to edge
**Case 3:**
$p(T_1|--) = \frac{p(T_1)}{p(T_0)+p(T_1)}$ = prob. of assigning +1 to edge
$p(T_0|--) = \frac{p(T_0)}{p(T_0)+p(T_1)}$ = prob. of assigning -1 to edge

In the above, $p(T_i)$ refers to the percentage of $T_i$ in the known data. We can thus predict the probability that the sign is a +1 knowing the sign of the partial triad that the edge contributes to. Doing this for each newly created triad, we can then average all the computed probabilities to give a probability that the unknown edge is a +1. We then assign the sign following the computed distribution i.e : if we obtain that the probability of a positive edge is $p = 0.7$, we will give this edge a +1 70 % of the time. Like in "Balance theory", we disregard the direction of the edges in the triad. Lastly, in the case that an edge does not form a triad at all when adding it to the graph, we assign +1 following the percentage of positives edges in the known data, i.e if there are 77 % of positive edges, we will assign a +1 77 % of the time. We note that this case occurs very rarely, and that most of the edges do create new triads.

*Tuning Parameter*

Furthermore, we introduce 3 tuning parameters $(\lambda_1, \lambda_2, \lambda_3)$ that count as additional factors to the before-mentioned conditional probabilities to favor certain triad types over others. For Case 1 we favor $T_3$ over $T_2$, for Case 2, we favor $T_1$ over $T_2$, and for Case 3, we favor $T_0$ over $T_1$. The ideas of structural triadic balance are directly consistent with the favors in Cases 1 and 2 , but, not necessarily with Case 3. However, we find the Case 3 tuning do improve the results. We did not optimize those weights, and only use the same weights for the 3 data sets, after doing a few trials. One idea to optimize the weights would be to perform cross validation on the known data in order to find the weights that perform better. However we realized that the optimization procedure was complex and would require a very long investigation in the context of this project.

### C. "Average Edge" Method

The basic idea behind this method is to determine an unknown edge weight between two nodes based on comparing the mean weights of incoming and outgoing edges between the two nodes. More specifically, we compare the mean of the outgoing edge weights of the *Source* node with the mean of the incoming edge weights of the *Target* node and assign +1 if both means are positive and assign -1 if both means are negative. In the case of a having different signs for the means, we assign the weight based on the higher absolute mean. If both means have the same absolute value, but different signs, we assign the weight randomly. We perform this method in two settings: firstly, the "Leave One Out" setting (LOO), and secondly, the "Remove Partition" setting (RP). LOO method consists in predicting one edge at a time knowing all the remaining graph. RP method consists in removing a whole part of edges at once.

## V. Results

The primary metric we use to evaluate our results is the Mathew's correlation coefficient (MCC). This coefficient gives a measure of the quality of a binary classification and most importantly, takes into account the true negatives which the F-score doesn't do for example. The higher the MCC is, the better the predictions are in both classes. In addition we also list the Specificity and Accuracy.

### A. "Triad based" Method

The table III features the evaluation metrics for our first method. As we can see the "Triad based " method does outperform Base. 2, but not Base. 1 in terms of accuracy. Nonetheless, it does outperform both baselines in specificity and MCC. However, overall compared to "Average Edge" method the prediction performance is much less when we look at the results in the next section. We can see that the performance of the "Triad based" Method doesn't perform well for the task of predicting the signs. However, we found out that it performs well for predicting the number of triads in the Graph. We therefore think that the Algorithm would be more appropriate if the objective is to have a network that would have global structure properties similar to the real unknown network, instead of focusing on predicting the correct sign of each edge.

### B. "Average Edge" Method

Table IV gives a first impression on the average edge algorithm (all edges are known, only averages are done). We find that in terms of all metrics this method outperforms

| Metric | Epinions | Slashdot | Wikipedia |
|---|---|---|---|
| Specificity | 0.22 | 0.32 | 0.46 |
| Accuracy | 0.86 | 0.73 | 0.62 |
| MCC | 0.22 | 0.18 | 0.1 |

Table III
PREDICTION SCORES FOR "TRIAD BASED" METHOD

the baseline predictions and the "Triad Based" method in both settings. But the drawback of this first method is that it overestimates positive weights which leads to a low specificity and indicates that too many negative weights are wrongly predicted positive. In order to remedy to this problem, we add a hyper-parameter $\alpha$ in the range [0;1] which weigh the importance of negative signs. After optimizing this parameter, we manage to balance the classes and increase the MCC scores.

*Tuning Parameter*

In order to improve our results and determine $\alpha$, we implement a gradient descent algorithm, where the loss function is based on the Mathew's correlation coefficient (MCC). We want to find the optimal parameter $\alpha$ such that the MCC is maximized. In the end we receive the optimal values of 0.63,0.63, and 0.66 for the Epinions, Slashdot, and Wikipedia data set respectively. Figure 1 left image shows us the plot of the MCC coefficients for different $\alpha$ and different data sets to illustrate the behaviour of the parameter.

Interestingly, a very similar/same ($\approx$ 0.65) parameter optimizes all different data sets which might indicate that it has a broader signification. Intuitively, the optimal coefficient would be based on the ratio of positive and negative edges. For example, if there are 85% of positive edges like in Epinions, the intuition would be to give a weight of 0.85 to the negative weights ($\alpha$=0.85) and 0.15 to the positive ones. But this is not the case here. All $\alpha$ from the different data sets are underestimated with respect to this intuition. The first concrete hypothesis on the meaning of this coefficient would be that negative votes/ratings have more ease to tilt the scales than positive ones. In other words, it confirms the fact that graphs often contain communities of condensed positives links where those communities are separated by negative links. Indeed, it means that positive links are usually very condensed around nodes and that the average of outgoing/incoming edges will definitely be positive (close to 1). On the other hand, negative links are often alone or close to a few positive links which make them a hard opponent against the positive links. So, as positive links are more condensed than negative ones, only a low $\alpha$ coefficient is required to make the negative links confrontable with positive links.

The table V features the evaluation metrics for both prediction settings (LOO and RP). We find that in terms of all metrics this method outperforms the baseline predictions and the "Triad Based" method in both settings. Of course the predictions are not as good as the raw algorithm in

| Metric | Epinions | Slashdot | Wikipedia |
|---|---|---|---|
| Neg. Predictive Value | 0.98 | 0.915 | 0.932 |
| Specificity | 0.684 | 0.573 | 0.396 |
| Accuracy | 0.952 | 0.884 | 0.866 |
| MCC | 0.795 | 0.644 | 0.554 |
| Neg. Predictive Value, $\alpha$ | 0.883 | 0.777 | 0.755 |
| Specificity,$\alpha$ | 0.821 | 0.773 | 0.712 |
| Accuracy,$\alpha$ | 0.958 | 0.898 | 0.890 |
| MCC,$\alpha$ | 0.827 | 0.709 | 0.664 |

Table IV
PREDICTION SCORES FOR AVERAGE EDGE (LEAVE ONE OUT)
TOP: WITHOUT $\alpha$, BOTTOM: WITH $\alpha$

| Metric | Epinions | Slashdot | Wikipedia |
|---|---|---|---|
| Accuracy (LOO) | 0.929 | 0.856 | 0.868 |
| MCC (LOO) | 0.713 | 0.595 | 0.603 |
| Accuracy (RP) | 0.891 | 0.820 | 0.866 |
| MCC (RP) | 0.609 | 0.523 | 0.586 |

Table V
PREDICTION SCORES FOR AVERAGE EDGE
(LEAVE ONE OUT AND REMOVE PARTITION)

Figure IV. But even more interesting, we see that Wikipedia performance is the data set where its performance decreases the less compared to others (0.664 to 0.586) and even surpass Slashdot (see Figure 1, right image). The fact that it highly benefits from this parameter tuning implies that a lot of positive communities are found in the data set and that those communities are most likely separated by less dense communities with more negative votes. In other words, that means that votes are mostly oriented towards the same candidates and all other candidates only have a small impact on electors.

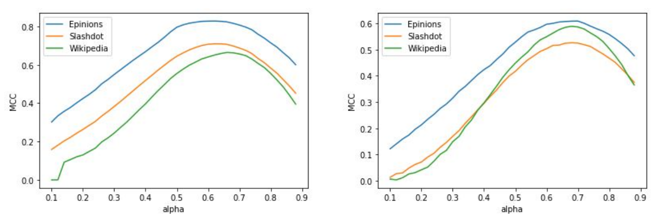


Figure 1. Average Edge Tuning. Left Image: Raw algorithm. Right Image: Remove Partition

## VI. CONCLUSION

The "Triad based" algorithm that we developed for this project can be seen as a different approach towards edge prediction, where in this paper we attempt to lay the basic ground work for this method. On the other hand, we have demonstrated that our "Average Edge" method seems to perform quite well with respect to local edge predictions. We have also seen that we were able to optimize this algorithm to a significant extent. Future work concerning this method could be in further studying the $\alpha$ parameter in understanding its broader significance.

## REFERENCES

[1] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2010, pp. 1361–1370.