Signed Networks

Harsh Patel
Department of Computer Science
IIT Gandhinagar
harsh.patel@iitgn.ac.in

Pushkar Mujumdar
Department of Computer Science
IIT Gandhinagar
pushkar.mujumdar@iitgn.ac.in

Shivam Sahni
Department of Computer Science
IIT Gandhinagar
shivam.sahni@iitgn.ac.in

Abstract

The advancements in the field of technology has created an enormous rise in the number of internet users. Studying and moderating social networks is a big challenge. Researchers have come up with various methods of studying relationships between users with the help of network analysis. Signed networks analysis is a hot topic in the field as it gives the ability to connect the computational analysis of social networks to social psychology.

1 Signed Networks?

Modern day problems often require us to capture the relation between multiple entities in a network. Signed networks are a powerful way of data representation that effortlessly reflect the mixture of positive and negative relations between entities.

Signed networks have their applications in various real life problems like social network analysis, bitcoin trust network analysis, etc. Signed networks enable us to carry out our analysis of such networks while upholding the *Social balance theory* which is discussed in section 3.

2 Dataset Information

We primarily use the Bitcoin OTC Trust Weighted Network [9] for our experiments.

1. Positive edges represent trust between users
2. Negative edges represent distrust between users

Statistic	Value
Nodes	5881
Edges	35592
Range of Edge weight	-10 to +10
Percentage of Positive Edges	89%

Table 1: Dataset Statistics

For ease in experimentation processes, we reduce the range of Edge weights to $[-1 \text{ to } +1]$. We also modify the values to form a zero-mean distribution of the edge weights for better interpretation of results. We believe that these heuristics wouldn't affect the true characteristics of the network.

3 Notion of Reputation in Signed Networks

There can be multiple notions of Reputation in a Signed Network. Reputation can be defined in terms of how trustworthy a user is. It can also be defined on the basis of the risk involved in making a transaction with that user. We will try to acknowledge most of these perspectives of Reputation in a Signed Network.

3.1 Why Link Prediction?

Before we move on to our problem statement, we need to acknowledge the fact that the signed network that we often get is an incomplete graph. According to the Social Balance Theory *"A friend of an enemy is also an enemy"*.

Suppose we are given a graph with three nodes A, B and C. Suppose there is positive relation between A and B, and negative relation between B and C, but the relation between A and C is not specified. So without link prediction it would be taken as neutral relation, which might be wrong. Here, we cannot ignore the fact that according to the Social Balance Theory, the relations between A and C would likely be negative.

Link Prediction helps in augmenting the signed network and enables it to capture such social interactions.

3.2 Reputation using Link Prediction

A simple algorithm suggested in one of the works [5] from Stanford defines two statistics, Fairness and Goodness. The Fairness of a node quantifies how fair a node is, at judging the Goodness of other nodes. The Goodness defines how good or reputed a node actually is.

The mutually recursive definitions of Fairness and Goodness of a node are as follows:

$$g(v) = \frac{1}{|in(v)|} \sum_{u \in in(v)} f(u) * W(u,v)$$

$$f(u) = 1 - \frac{1}{|out(u)|} \sum_{v \in out(u)} \frac{|W(u,v) * g(v)|}{R}$$

Now, the weight of the edge from a node u to node v is predicted as:

$$E(u,v) = f(u) * g(v)$$

With this, we define the Reputation of a node in the signed network as the *Goodness* of that node in the complete graph ie. the graph after link prediction.

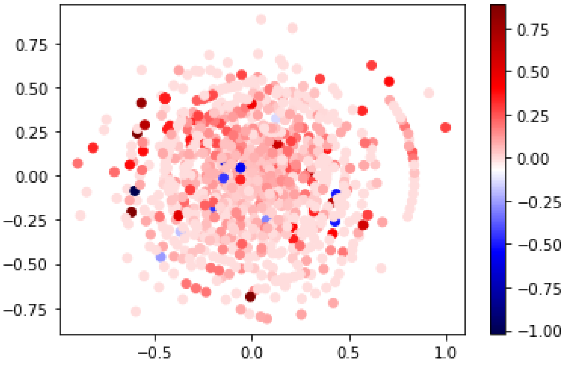

Figure 1: Reputation of the users represented by their goodness values. The plot is shown for the first 10% of the data. Nodes are plotted using Kamada Kawai layout. The nodes in red are more reputed (high goodness) while the ones in blue are less reputed (low goodness).

To evaluate the performance of this statistic, we use the dynamic nature of the Bitcoin OTC Trust Network. We partition the entire network according to the presence of its entities in a particular time-frame. We evaluate the absolute error between the predicted and the expected edge weights for different split proportions as shown in Table 2.

Train-Test %	No. of Predicted Edges	Mean Absolute Error
50% - 50 %	17796	0.1715892
70% - 30 %	10678	0.1869823
90% - 10 %	3560	0.1336905
	Average MAE:	**0.164087**

Table 2: Link Prediction results using Fairness and Goodness. The mean absolute error is calculated between the predicted edge weights and the true edge weights detected after the dynamic growth of the network. The network is split on a particular time state based on the train %.

3.3 Reputation using SignRank

The previous approach we saw involved link prediction and was a deterministic method of defining reputation in a signed network. Now let us try to apply the probabilistic model - SignRank, similar to Personalized PageRank as proposed in this paper [6] to solve our problem.

SignRank computes the probabilities of a random walker visiting a node while having positive emotion and negative emotion. The random walker starts at a random node with positive emotion and it's emotion flips over whenever it traverses a negative edge. The random walker may also traverse an edge without taking into account its emotion with probability λ. It might also jump to a random node with probability α. These probabilistic jumps help in regularization of the model as we are not doing link prediction here.

The probability of reaching a node i with positive emotion at time $t + 1$ is given by:

$$\pi_i^{+(t+1)} = \alpha(\textstyle\sum_{j \in IN^+_{(i)}} \pi_j^{+(t)} p_{ji} + (1-\lambda) \sum_{j \in IN^-_{(i)}} \pi_j^{-(t)} p_{ji} + \frac{\lambda}{2N} \sum_{j=1}^{N} \pi_j^{-(t)}) + \frac{1-\alpha}{2N}$$

Similarly, the probability of reaching a node i with negative emotion at $t + 1$ is given by:

$$\pi_i^{-(t+1)} = \alpha(\textstyle\sum_{j \in IN^-_{(i)}} \pi_j^{+(t)} p_{ji} + (1-\lambda) \sum_{j \in IN^+_{(i)}} \pi_j^{-(t)} p_{ji} + \frac{\lambda}{2N} \sum_{j=1}^{N} \pi_j^{-(t)}) + \frac{1-\alpha}{2N}$$

Here, $p_j i$ represents the normalized edge weight from j to i, and $IN^+_{(i)}$ and $IN^-_{(i)}$ are the positive and negative incoming edges of node i.

On convergence of this algorithm on the Signed Network, we can define the Reputation of a node as the ratio of the positive emotion probability and the negative emotion probability.

$$R(i) = \frac{\pi_i^+}{\pi_i^-}$$

3.4 Comparing Performance

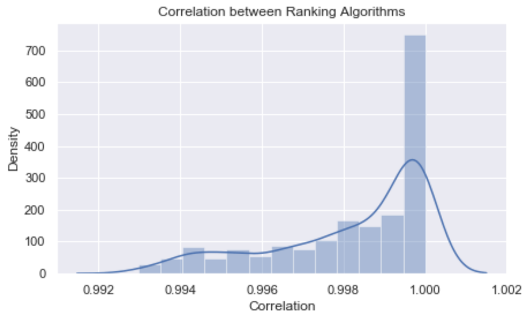

Figure 2: This is a distribution plot of the correlation between the rankings obtained by the above two algorithms. The metric plotted is used in the summation term of Spearman's Rank Correlation calculation.

4 Notion of Communities in Signed Networks

Community detection is a very popular technique for understanding and evaluating the characteristic properties of large complex networks, especially social networks. However, it has been significantly under-explored for signed networks as compared to unsigned ones [7]. Few previous works include modified Spectral Clustering [8], Correlation Clustering [1] for community detection in signed networks. Using graph clustering approaches over the Bitcoin OTC Trust Network, we aim to identify most trust-worthy and conflicting communities in this transaction network. We define the following metric of each individual cluster to estimate their trust-worthiness.

$$Trust_{C_i} = \frac{\sum_{edges} w_{(inside+ve)} + \sum_{edges} |w|_{(outside-ve)}}{N_{edges_{Ci}}}$$

This metric is defined over the basic notion of what is popularly known in these settings as Agreements, which means more no. of positive edges inside and negative edges outside a cluster.

4.1 Spectral Clustering

Given a graph, spectral clustering allows us to perform dimensionality reduction on the graph using its adjacency and degree matrix, to obtain node embeddings in fewer dimensions. We can subsequently employ clustering techniques on these embeddings to cluster the graph into desired number of clusters. The motivating idea behind Spectral Clustering is Graph Drawing.

4.1.1 Graph Drawing

The objective of Graph Drawing is to generate embeddings for nodes such that a node is close to all of its neighbours. More concretely, it tries to place every node at the mean of the its neighbour positions [8].

For positive weighted edges, we get the following vertex equation:

$$X_i = (\textstyle\sum_{j\sim i} A_{ij})^{-1} \sum_{j\sim i} A_{ij} X_j$$

On simplifying, we obtain,

$$Dx = Ax$$
$$Lx = 0$$

where D is the diagonal matrix, A is the adjacency matrix and $L = D - A$. We call the L matrix as the unsigned Laplacian Matrix.

To extend this equation to Signed Networks, we need to look at some subtleties. First, we want a node to be close to its positive weight neighbours and far from its negative weight neighbours. Second, we need to change the normalising part in the vertex equation.
For Signed Network the equation now becomes:

$$X_i = (\textstyle\sum_{j\sim i} |A_{ij}|)^{-1} \sum_{j\sim i} A_{ij} X_j$$

on simplifying,

$$\bar{D}x = Ax$$
$$\bar{L}x = 0$$

where $\bar{D}$ is defined as $\bar{D}_{ii} = \sum_j |A_{ij}|$ and we obtain the signed Laplacian Matrix [8], $\bar{L} = \bar{D} - A$.

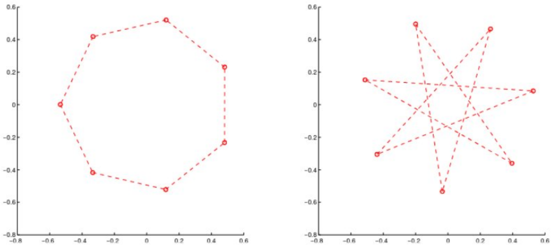

Figure 3: The figure depicts the embeddings for a Graph containing only negative edges by using unsigned Laplacian embeddings and signed Laplacian embeddings respectively from left to right. We see that the nodes are more separated in the right plot. Thus, the signed Laplacian embeddings perform better here.

4.1.2 Signed Spectral Clustering

To cluster the graph, we simply use the signed Laplacian matrix as calculated above to generate embeddings in a lower dimension. Further, we apply K-Means clustering to generate clusters in our Signed Graph. For all the subsequent experiments we use the first 10% of the entire time frame of this dynamic network.

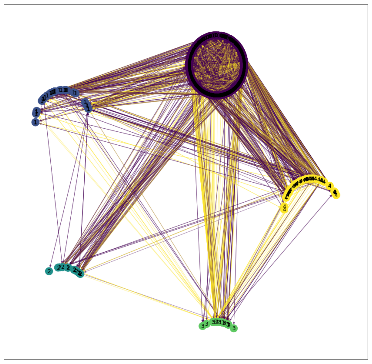

Figure 4: Signed Spectral Clustering over the original graph

Cluster No.	No. of Nodes	Inside +ve edges	Outside -ve edges	Trust Ci
1	660	798	342	0.070695
2	69	23	216	0.022101
3	50	3	132	0.017225
4	20	5	33	0.013065
5	18	0	57	0.000846
Total Nodes:	817	**Total Trust metric of the Network:**		0.123934

Table 3: Signed Spectral Clustering (SSC) over the original graph

4.1.3 Results of SSC

Figure 4 shows the clustering obtained using SSC on the original graph. The selection of the total number of clusters is done using the elbow method. Table 3 shows the reputation of each cluster according to the defined metric. We also show results of SSC on the complete graph obtained using link prediction in Table 4. We see from our results that SSC gives a relatively less skewed clustering for complete graph, and also shows a better Total Trust value obtained in Table 4.

4.2 Correlation Clustering

Correlation clustering is an algorithm majorly used in a scenario where the relations between the entities of our network are known. The algorithm tries to put similar objects into same clusters and

Cluster No.	No. of Nodes	Inside +ve edges	Outside -ve edges	Trust Ci
1	536	158333	64716	0.048947
2	208	21606	53570	0.020009
3	37	921	12954	0.018968
4	9	63	3250	0.008264
5	10	63	3583	0.008117
6	6	20	2151	0.008131
7	5	20	1813	0.006756
8	4	12	1443	0.006718
9	2	2	723	0.006319
Total Nodes:	817	**Total Trust metric of the Network:**		0.132229

Table 4: Signed Spectral Clustering over the Complete Network: Link Prediction

dissimilar into different clusters. A simple definition of similarity completes the major requirements for the algorithm. Although it seems quite simplistic, the domain of its general applicability has been been widening since the initial publication of this work [1] especially for Community Detection in real-world networks like social networks [2] or protein–protein interaction networks [3].

In the paper [1], they describe a constant factor approximation algorithm for minimizing Disagreements i.e., defined as the absolute sum of negatives inside and the sum of positives outside each clusters. This iterative algorithm tries to estimate δ-clean clusters, where δ-clean implies that for each node $v \epsilon C$, the cluster has atleast $(1 - \delta) * |C|$ positive-weighted neighbours inside C and atmost $\delta * |C|$ positive-weighted neighbours outside the cluster.

The fact used behind the correctness of this algorithm is that if the graph contains an erroneous triangle, i.e., a triangle with 2 positive edges and one negative edge, then the clustering can never be perfect. The algorithm involves the following two major steps that indeed help in bounding the number of disagreements by a constant factor of the number of mistakes in the optimal clustering.

4.2.1 Pseudo-Code similar to that in [1]

```
While there are nodes left to cluster in $V$:
    ⇒ Randomly choose a vertex $v$ and find all its positive neighbors, say $A(v)$
    ⇒ (Vertex Removal Step): While $\exists$ $x$ in $A(v)$ such that x is $3 * \delta$-bad wrt $A(v)$, remove $x$ from $A(v)$
    ⇒ (Vertex Addition Step): Add all the vertices from the vertex set $V$ into $A(v)$ that are $7 * \delta$-good wrt $A(v)$
    ⇒ Announce $A(v)$ as one cluster and remove it from the $V$
```

The clusters formed using the above algorithm satisfy the following bound backed by few other lemmas discussed in the paper. The total number of disagreements encountered in the above Cautious algorithm is atmost 8 times the number of disagreements in OPT.

In our initial experiments, to increase the interpretability of the results, we only consider the sign of the trust relations between the Bitcoin users. One major advantage of using the correlation clustering algorithm is that we do not need to find the optimal number of cluster centers for our network. The algorithm iteratively determines the best clustering while minimizing the disagreements as defined before. Although the cleanliness parameter in this algorithm plays a vital role in the clustering performance.

4.2.2 Results of Correlation Clustering

We perform the clustering over two different graph models of the Bitcoin-Trust network stated below:

A) Without the link prediction i.e., An Incomplete Graph:

Figure 5 shows the clustering results using the correlation approach over the original graph of 817 nodes. The clustering looks very even as per the initial observations. We have successfully partitioned

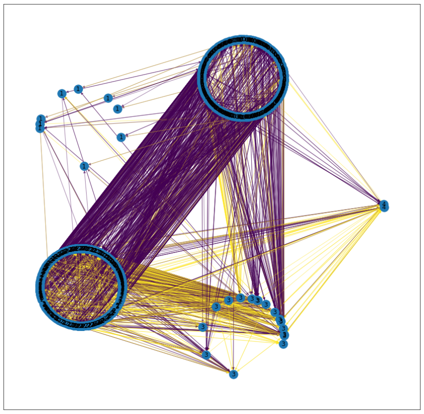

Figure 5: Correlation Clustering over the original graph with Cleanliness parameter = 0.2

the trust-network into 5 conflicting communities. Here, each community has its own trust levels calculated using the above defined metric $Trust_{C_i}$ as shown in Table 5. With this clustering we can see that the community having the maximum no. of agreements has higher trust levels as compared to the other communities.

Although the above clustering shows decent results, in the subsequent section, we show that how link prediction can be used to enhance the community detection approach with a better representation of the network.

Cluster No.	**No. of Nodes**	**Inside +ve edges**	**Outside -ve edges**	**Trust Ci**
1	380	598	582	0.095065
2	19	47	260	0.055009
3	2	2	20	0.036433
4	406	0	520	0.024462
5	10	0	15	0.000722
Total Nodes:	817	**Total Trust metric of the Network:**		0.211691

Table 5: Correlation Clustering over the original graph

B) With link prediction i.e., a Complete Graph: In this experiment, we firstly predict all the unknown weights of the network and then perform correlation clustering over the complete graph. Figure 6 shows the clustering results. Table 6 shows a summary of both the number of nodes and

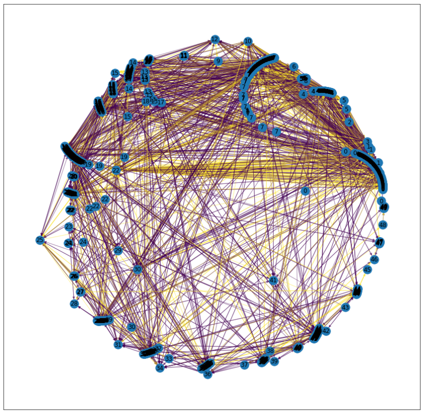

Figure 6: Correlation Clustering over the Complete graph: Link Prediction is done but all the edges could not be shown. Cleanliness parameter = 0.05

the agreement features of the individual clusters. Using this approach, we observe that the Trust metric value of the entire graph is higher than what we observed with initial approach of using an incomplete graph for clustering. We also observe that the best community of 125 nodes detected from the complete graph and the community of 380 nodes detected using the incomplete graph clustering have more than 60% of common users, thus validating that both the approaches detect a set of nodes having similar features, in this case, maximising the number of agreements. We also infer that the no. of clusters with this network model are more than the previous model showing how erroneously the previous model captures the relations between unknown edges among the users. This backs up our initial hypothesis that link prediction is an essential component for the network model to follow the most basic Social Balance Theory.

Cluster No.	**No. of Nodes**	**Inside +ve edges**	**Outside -ve edges**	**Trust Ci**
1	125	10245	39788	0.020745
2	148	14136	45778	0.01974
3	52	1914	18040	0.011124
4	76	2691	24335	0.010605
5	36	594	12270	0.009615
.	.	.	.	.
.	.	.	.	.
50	1	0	253	0.005589
Total Nodes:	817	**Total Trust metric of the Network:**		0.38201117

Table 6: Correlation Clustering over a Complete Network: Link Prediction done

5 Why should we even think of Resiliency of these real-world networks?

To give an analogy, Silk Route was a large drug marketplace which used primarily bitcoins as a medium of exchange. When the people behind the marketplace got arrested, a lot of bitcoin users lost trust in the crypto-currency and fled.

Disruptions like these are quite common in real-world networks. These disruptions have a severe affect on both the life and strength of the network. Thus, resiliency estimation is an important task while analysing any large realistic network. We attempt to perform a disruption event into the Bitcoin Trust-Network to quantize and analyse how the network gets affected and what possible measures can be taken to reduce the strength of impact due to such events. In the previous section, using the correlation clustering approach, we were able to detect the best community in the network on the basis of its Trustworthiness. In this experiment, we remove the most trustworthy community from the network and then recalculate all the metrics as defined in the previous experiments. Firstly, we should understand how would an emotional sentiment like Trust, affect such a large transaction network. As per [4], Trust is an emotional brain state that leads to behaviors like acting in ways that depend on an another individual. Since we are removing the most trusted community from the network, we hypothesize that this could severely affect the strength i.e., the Trust level of the entire network. Figure 7 shows the clustering results over the disrupted network.

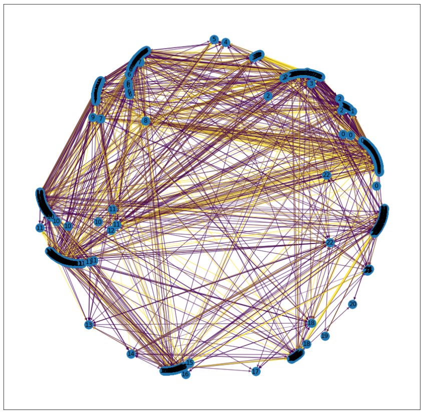

Figure 7: Correlation Clustering results over the Disrupted Network: Removal of the most-trustworthy community

If we see the results from Table 7, it can be very well inferred that the Total Trust Metric of the Disrupted network reduces to almost half i.e, 0.19 from 0.38. With this results we can see how such a popular network can be affected by natural disruptions. It can be seen that there is a lot of scope in improving the resiliency of such real-world networks. One interesting open-ended approach could be to enhance the network by identifying and flagging fraudulent users which could possibly hinder the strength of such large networks.

Cluster No.	**No. of Nodes**	**Inside +ve edges**	**Outside -ve edges**	**Trust Ci**
1	96	6672	29403	0.01932
2	90	5118	26974	0.014159
3	80	1971	22066	0.012576
4	56	1373	16783	0.011443
5	86	3207	24379	0.011293
.	.	.	.	.
.	.	.	.	.
23	1	0	352	0.004836
Total Nodes:	692	**Total Trust metric of the Network:**		0.1918919

Table 7: Correlation Clustering over the Disrupted Network

6 Conclusion

In this project we analyzed signed networks in the context of Bitcoin OTC Trust Network. We first explored the notion of reputation of a node defined using both deterministic and probabilistic approaches i.e., Goodness and Sign Rank respectively. We then defined our own metric to quantify the trustworthiness of a community based on the notion of agreements in the network. We then looked at the ways to detect trustworthy or say, healthy communities using the popular clustering approaches - Spectral and Correlation Clustering. We finally perform resiliency analysis on this signed network by exposing the network to a custom disruption event.

References

[1] Bansal, N., Blum, A. & Chawla, S. Correlation Clustering. Machine Learning 56, 89–113 (2004). https://doi.org/10.1023/B:MACH.0000033116.57574.95

[2] Santo Fortunato,Community detection in graphs,Physics Reports,Volume 486, Issues 3–5,2010,Pages 75-174,ISSN 0370-1573,https://doi.org/10.1016/j.physrep.2009.11.002.

[3] G. Palla, I. Derényi, I. Farkas, T. Vicsek Uncovering the overlapping community structure of complex networks in nature and society Nature, 435 (2005), pp. 814-818

[4] Paul Thagard, “What Is Trust?” Psychology Today, Sussex Publishers, www.psychologytoday.com/us/blog/hot-thought/201810/what-is-trust.

[5] S. Kumar, F. Spezzano, V. S. Subrahmanian and C. Faloutsos, "Edge Weight Prediction in Weighted Signed Networks," 2016 IEEE 16th International Conference on Data Mining (ICDM), 2016, pp. 221-230, doi: 10.1109/ICDM.2016.0033.

[6] Cong Wan, Yanhui Fang, Cong Wang, Yanxia Lv, Zejie Tian, Yun Wang, and Huaming Wu. 2019. SignRank: A Novel Random Walking Based Ranking Algorithm in Signed Networks. Wirel. Commun. Mob. Comput. 2019 (2019). DOI:https://doi.org/10.1155/2019/4813717

[7] Esmailian, P., Jalili, M. Community Detection in Signed Networks: the Role of Negative ties in Different Scales. Sci Rep 5, 14339 (2015). https://doi.org/10.1038/srep14339

[8] Kunegis, Jérôme & Schmidt, Stephan & Lommatzsch, Andreas & Lerner, Jürgen & De Luca, Ernesto & Albayrak, Sahin. (2010). Spectral Analysis of Signed Graphs for Clustering, Prediction and Visualization. Proc SDM. 559-. 10.1137/1.9781611972801.49.

[9] S. Kumar, B. Hooi, D. Makhija, M. Kumar, V.S. Subrahmanian, C. Faloutsos. REV2: Fraudulent User Prediction in Rating Platforms. 11th ACM International Conference on Web Searchand Data Mining (WSDM), 2018.