Weighted Signed Graph Attention Networks

Bharat Runwal (2018EE30598)
Shreyansh Choudhary (2018MT60794)

September 2021

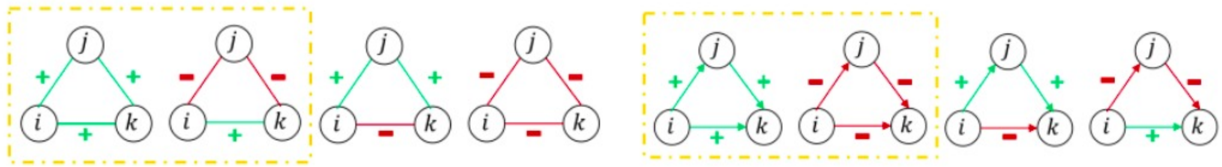
Introduction

The study of networks is a highly interdisciplinary field that draws ideas and inspiration from multiple disciplines including biology, computer science, economics, mathematics, physics, sociology, and statistics. In particular, social network analysis deals with networks that form between people. Online, social networks have been represented as graphs, with nodes representing entities, and edges representing relationships between entities. However, when a network has like/dislike, love/hate, respect/disrespect, or trust/distrust relationships, such a representation is inadequate since it fails to encode the sign of a relationship. Hence, we need signed networks. Signed network em- bedding learning to map nodes in signed network to low-dimensional vector representations. For signed network, SGCN generalizes GCN (Thomas N. Kipf and Welling 2017)to signed networks and designs a new information aggregator based on balance theory. However, it only applies to the undirected signed networks. SiGAT(Huang et al. 2019c) incorporates graph motifs into GAT to capture two well-known theories in signed network research, i.e., balance theory and status theory. In SiGAT, motifs offer us the flexible structural pattern to aggregate and propagate messages on the signed network to generate node embeddings. But the problem in the real world doesn't limits to just finding the sign of the edge, a person may be interested in how much he/she can trust some other person in a relationship or before making some transaction. Hence, in this project, our goal is to incorporates weights in SiGAT architecture to have better representation so that we can have a significant contrast in the embeddings of positively connected nodes and negatively connected nodes. The contrast may depend on the weight of the edge.

Related Work

Signed Network Embedding

Signed social networks are such social networks in signed social relations having both positive and negative signs. To analyse them, many algorithms such as



(a) Illustration of structural balance theory (b) Illustration of status theory

Figure 1:

community detection, node classification, link prediction, spectral graph analysis and many more have been developed. Recently, with the development of network representation learning, researchers begin to learn low-dimensional representations for signed networks. SNE (Yuan, Wu, and Xiang 2017) adopts log-bilinear model and incorporates two signed-type vectors to capture the positive and negative relationship of each edge along the path. For directed signed networks, SIDE (Kim et al. 2018) provides a linearly scalable method that leverages balance theory along with random walks. BESIDE (Chen et al. 2018) mathematically models “bridge” edges based on balance and status theory and achieves state-of-the-art performances.

Signed Network Theory

- **Balance Theory**: Balance theory was developed initially for the undirected signed networks. Triads with an even number of negative edges as balanced. In Figure 1(a), the triangles with three positive signs and those with one positive sign (i.e., the first two triads in Figure 1(a)) are balanced. Balance theory posits that balanced triads are more plausible — and hence should be more prevalent in real-world networks — than unbalanced triads. It exemplifies the principle that the friend of my friend is my friend and the enemy of my enemy is my friend. Balance theory is widely used in the field of signed networks.

- **Status theory**: Status theory is another critical sociological theory in signed network analysis, which provides a different organizing principle for directed networks of signed links. It supposes that a positive directed link “+” indicates that the creator of the link views the recipient as having higher status; and a negative directed link “-” indicates that the recipient is viewed as having lower status. The status may denote the relative prestige, ranking, or reputation. For example, a positive link from A to B means not only B is A’s friend but also B has a higher status than A. For the triangles in Figure 1(b), the first two triads satisfy the status order, but the last two do not satisfy it. For the first triads, when Status(j) > Status(i) and Status(k) > Status(j), we have Status(k) > Status(i).

Graph Neural Networks

Majorly all the GNN's today can be summarized as Message Passing Neural Networks. As the graphs are non Euclidean Structures. traditional RNN's and CNN's are not easy to generalise to graph domains. With the help of GNN's, researchers have incorporated Convolutions, attention, LSTM's and other mechanisms into the graph data. Graph auto-Encoders (Thomas N Kipf and Welling 2016) aim to map the nodes of graph to lower dimensional embedding which reconstruct the graph. A lot of GNN models show a better performance than the shadow look-up embeddings. Most of the research have explored GNNs in unsigned network with positive edges only. SGCN (Huang et al. 2019a)designs a new information aggregation and propogation mechanism for the undirected signed networks according to balance theory. SGCN applies a mean-pooling strategy that is close to GraphSAGE to learn node embeddings. SiGAT (Huang et al. 2019c)introduces GAT(Velickovic et al. 2018) to directed signed networks and designs a signed motif based graph neural network model based on social theories(balance theory and status theory),in this work the authors define positive/negative neighbors (2 motifs), positive /negative with direction neighbors (4 motifs) and 32 different triangle motifs in Fig 2.

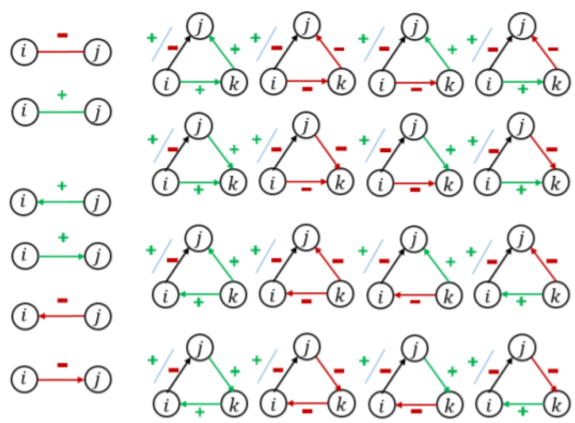


Figure 2: Different motifs from Huang et al. 2019b

Kumar et al. 2016 proposed two novel measures of node behavior: the goodness of a node intuitively captures how much this node is liked/trusted by other nodes, while the fairness of a node captures how fair the node is in rating other nodes' likeability or trust level.

Problem Statement

A Weighted Signed Network (WSN) is a directed, weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ where $\mathcal{V}$ is a set of nodes, $\mathcal{E} \subseteq V \times V$ is a set of edges, and $W : \mathcal{E} \rightarrow [-1, +1]$ is a mapping that assigns a value between -1 and +1 to each edge. $W(u,v)$ can be thought of as assigning a degree of "likes", "agrees" or "trust" score describing how much node u similar to a node v. WSNs, like any network, are incomplete. There may be like/dislike, trust/distrust or agree/disagree relations between people which we don't know about or are yet to form. Our goal is to predict the edge weights of the weighted signed network keeping in mind the two important sociological theories - Balance theory and Status theory.

Methodology

Our base architecture will be based on the above mentioned SiGAT Huang et al. 2019b.The attention mechanism in the graph was introduced by **GAT**,the authors of the SIGAT addapted this work to the signed graphs, specifically they computes the $\alpha_{uv}^{m_i}$ for target node u and node v , where $v \in \mathcal{N}_{m_i}(u)$ i.e. neighbourhood of node u for a particular motif m_i and do this for every motif m_i as follows :

$$[H]\alpha_{uv}^{m_i} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}_{m_i}^T[\mathbf{W}_{m_i}X(u)\|\mathbf{W}_{m_i}X(v)]\right)\right)}{\sum_{k\in\mathcal{N}_{m_i}(u)}\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}_{m_i}^T[\mathbf{W}_{m_i}X(u)\|\mathbf{W}_{m_i}X(k)]\right)\right)}, \tag{1}$$

$$X_{m_i}(u) = \sum_{v\in\mathcal{N}_{m_i}(u)} \alpha_{uv}^{m_i}\mathbf{W}_{m_i}X(v). \tag{2}$$

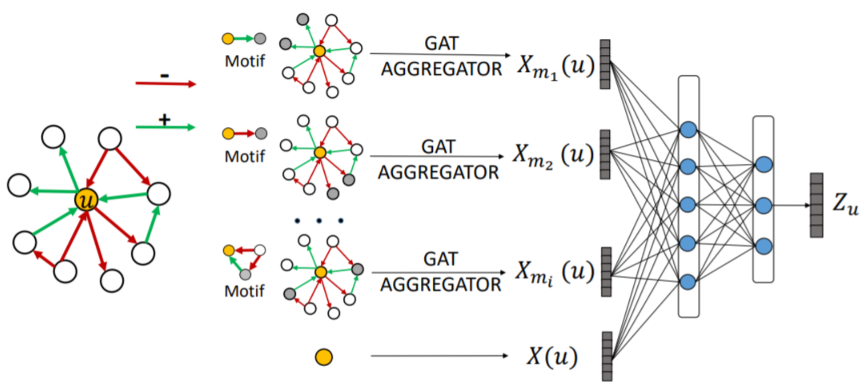


Figure 3: SiGAT- Node Embedding Update Mechanism

Where, X(u) are the embeddings of the node u initialized to the given features of u. $\mathbf{W}_{m_i}$,$\mathbf{a}_{m_i}$ are the parameters being learned.The messages we get from different motifs were concatenated with $\mathbf{X}(\mathbf{u})$ and passed to a two-layer fully connected neural network to obtain the final embedding $\mathbf{Z}_u$ Fig (3) for the

target node u. Here, W_1, W_2, b_1, b_2 are the parameters for the 2-layer neural network.
Unlike the work in Huang et al. 2019b,we defined our unsupervised loss function as the following :

$$J(Z_u) = -\sum_{v^+ \in \mathcal{N}(u)^+} \log\left(\sigma(\mathbf{W}_{E^+} Z_u^\top Z_{v^+})\right) - Q \sum_{v^- \in \mathcal{N}(u)^-} \log\left(\sigma(\mathbf{W}_{E^-} Z_u^\top Z_{v^-})\right), \tag{3}$$

where σ is the sigmoid function, $\mathcal{N}(u)^+$ is the set of positive neighborhoods of node u, $\mathcal{N}(u)^-$ is the set of negative neighborhoods of node u and Q is the balanced parameter for the unbalanced positive and negative neighborhoods. $\mathbf{W}_{E^+}$ and $\mathbf{W}_{E^-}$ are the respective positive and negative trust scores, this unsupervised loss function ensures that the friends embedding are similar and the enemy embeddings are dissimilar according to the assigned trust score in the data. This allow us to not just classify the edge as binary score of $+1$ and -1 but also allow us to have sense of how much trust should one put. For example a political party's relation with other political parties can be good but it's not likely that their relation with each of them will be on similar trust score or a country's relation with other maybe good but some countries will have stronger relations compared to other for example India-USA has better relations than India-China.One other difference is that we use Weighted Adjacency matrix during learning of our model whereas Huang et al. 2019b uses Adjacency matrix without edge weights.

Results

Dataset

A number of weighted signed exist in the wild. For this project, we take Bitcoin exchanges-Alpha.

- Bitcoin-Alpha: This is who-trusts-whom network of people who trade using Bitcoin on a platform. Members of Bitcoin Alpha rate other members in a scale of -10 (total distrust) to +10 (total trust) in steps of 1. 93% of the edges are positive-weighted.

  Link : Bitcoin-Alpha

- Wiki-Conflict: The edges in this network represent positive and negative conflicts between users of the English Wikipedia, for example users involved in an edit-war. A node represents a user and an edge represents an edit conflict between two users, with the edge sign representing positive and negative interactions. An example for a negative interaction would be when one user revert the edit of another user. The network is unipartite and undirected, it is signed, with multiple edges. Edges are annotated with time-stamps. Also, it contains 0 weighted edges. Wiki-conflict

Algorithm 1 SiGAT embedding generation (forward) algorithm

Input: Sigend directed Graph $G(V, E, s)$;
Motifs list $\mathcal{M}$; Epochs T; Batch Size B;
Motifs graph extract function $F_{m_i}, \forall m_i \in \mathcal{M}$;
Aggregator GAT-AGGREGATOR$_{m_i}$ with the parameter $\mathbf{W}_{m_i}, \tilde{\mathbf{a}}_{m_i}$, $\forall m_i \in \mathcal{M}$;
Weight matrices $\mathbf{W_1}, \mathbf{W_2}$ and bias $\mathbf{b_1}, \mathbf{b_2}$;
Non-linearity function Tanh
Edge-Weights $\mathbf{W_E}$;

Output: Node representation $Z_u, \forall u \in V$

$X(u) \leftarrow \text{random}(0, 1), \forall u \in V$
$G_{m_i} \leftarrow F_{m_i}(G), \forall m_i \in \mathcal{M}$
$\mathcal{N}_{m_i}(u) \leftarrow \{v|(u, v) \in G_{m_i}\}, \forall m_i \in \mathcal{M}, \forall u \in V$
for $epoch = 1, ..., T$ **do**
  for $batch = 1, ..., |V|/B$ **do**
    $\mathcal{B} \leftarrow V_{(batch-1)\times B+1:batch\times B}$
    for $u \in \mathcal{B}$ **do**
      for $m_i \in \mathcal{M}$ **do**
        $X_{m_i}(u) \leftarrow \text{GAT-AGGREGATOR}_{m_i}(\{X_v, \forall v \in \mathcal{N}_{m_i}(v)\})$
      end for
      $X'(u) \leftarrow \text{CONCAT}(X(u), X_{m_1}(u), ..., X_{m_{|\mathcal{M}|}}(u))$
      $Z_u \leftarrow \mathbf{W_2} \cdot \text{Tanh}(\mathbf{W_1} \cdot X'(u) + \mathbf{b_1}) + \mathbf{b_2}$
    end for
  end for
end for
return Z_u

	Wiki Conflict	Bitcoin-alpha
Nodes	118,100	3,783
Edges	2,917,785	24,186

Table 1: Data Statistics

Note: We could not perform experiment on the wiki-conflict dataset because of it's large size as we have limited computational resources, we used freely available Google Colab's NVIDIA Tesla K80 12GB GPU. The unsupervised loss in (3) converges very smoothly for our model as compared to the base model. (Fig. 4)

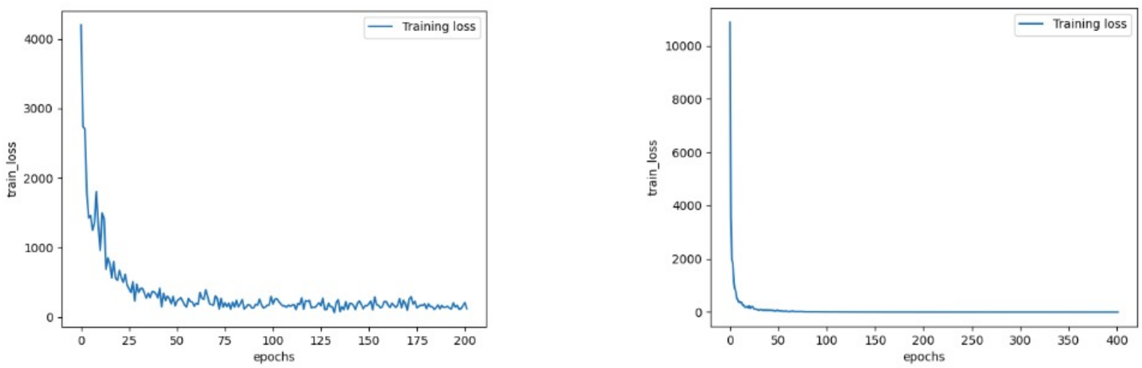


Figure 4: Train Loss difference, Left for the SiGAT and right for our model Modified SiGAT

We compare the performance of our model against the original paper for by comparing the embeddings obtained for the nodes for randomly selected edges from the graph. We performed t-SNE to visualise the embeddings.

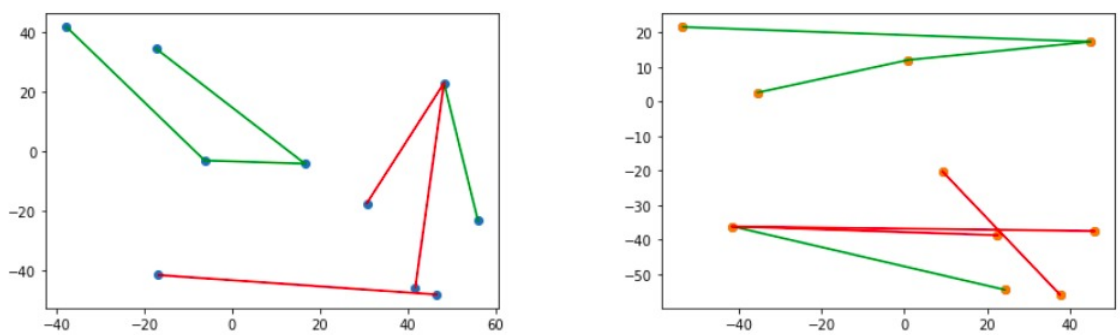


Figure 5: Learned embeddings of the models. Left for the SiGAT and right for our model Modified SiGAT. Note the difference in scale on y axis.

In Fig. (6), we can see the positively related nodes are pulled more closer to each other and the negatively related nodes are pushed apart.

Also when the weights of the models are regressed on the node embeddings, we observe that the base model predicts values near the mode but our model generalized well. The RMSE for the base model is 1.549 vs for the 1.545 for the modified version.

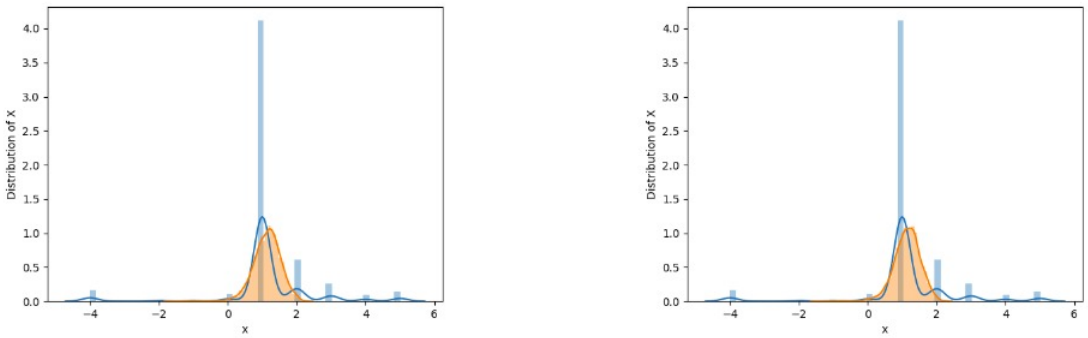


Figure 6: Distribution plots of the true value(blue) vs predicted values(orange) for both the models. Left for the SiGAT and right for our model Modified SiGAT

Future Work

We like to explore other methods which separate the the node embeddings of positive neighbour and negative neighbour better in latent space better i.e. positive is closer and negative are farther. For this purpose we want to work on contrastive learning based method which is used to learn representations such that the similar samples are closer while dissimilar ones are farther away. The contrastive learning in Graph Neural Network is not well explored, one of the difficulty is the data augmentation in the graph domain which is underexplored, the authors in You et al. 2020 proposed a graph contrastive learning approahc with augmentations like Edge perturbation(removal or addition), Node dropping, Attribute masking and Subgraph etc., then proposing a contrastive learning based loss function as :

$$\ell_n = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_{n,i}, \boldsymbol{z}_{n,j})/\tau)}{\sum_{n'=1, n'\neq n}^{N} \exp(\text{sim}(\boldsymbol{z}_{n,i}, \boldsymbol{z}_{n',j})/\tau)}, \tag{4}$$

τ is the temperature parameter,sim is similarity metric for example cosine similarity between, Embeddings $\boldsymbol{z}_{n,i}$ and $\boldsymbol{z}_{n,j}$) are for n^{th} graph in mini batch N(Maximize the similarity between these) and negatives are sampled from remaining all other N-1 augmented graphs from the same mini batch .
This approach cannot be directly applied in case of the Signed Graphs because we have to maintain the Balance theory, so we would like to work on formally defining data augmentation in Signed Graph and instead of using just one positive as done in (4), we would like to include multiple positive here to learn better representations.

References

Chen, Yiqi et al. (2018). “” Bridge” Enhanced Signed Directed Network Embedding”. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 773–782.

Huang, Junjie et al. (2019a). “Signed Graph Attention Networks”. In: *Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions*. Ed. by Igor V. Tetko et al. Cham: Springer International Publishing, pp. 566–577. ISBN: 978-3-030-30493-5.

— (2019b). “Signed Graph Attention Networks”. In: *ArXiv* abs/1906.10958.

— (2019c). “Signed graph attention networks”. In: *International Conference on Artificial Neural Networks*. Springer, pp. 566–577.

Kim, Junghwan et al. (2018). “SIDE: Representation Learning in Signed Directed Networks”. In: *Proceedings of the 2018 World Wide Web Conference*. WWW ’18. Lyon, France: International World Wide Web Conferences Steering Committee, pp. 509–518. ISBN: 9781450356398. DOI: 10.1145/3178876.3186117. URL: https://doi.org/10.1145/3178876.3186117.

Kipf, Thomas N and Max Welling (2016). “Variational graph auto-encoders”. In: *arXiv preprint arXiv:1611.07308*.

— (2017). *Semi-Supervised Classification with Graph Convolutional Networks*. arXiv: 1609.02907 [cs.LG].

Kumar, Srijan et al. (2016). “Edge Weight Prediction in Weighted Signed Networks”. In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 221–230.

Velickovic, Petar et al. (2018). “Graph Attention Networks”. In: *ArXiv* abs/1710.10903.

You, Yuning et al. (2020). “Graph Contrastive Learning with Augmentations”. In: *ArXiv* abs/2010.13902.

Yuan, Shuhan, Xintao Wu, and Yang Xiang (2017). *SNE: Signed Network Embedding*. arXiv: 1703.04837 [cs.SI].