



Polarity-based graph neural network for sign prediction in signed bipartite graphs

Xianhang Zhang¹ · Hanchen Wang^{2,1}  · Jianke Yu² · Chen Chen² · Xiaoyang Wang² · Wenjie Zhang¹

Received: 5 September 2021 / Revised: 22 November 2021 / Accepted: 21 January 2022 /
Published online: 16 February 2022
© The Author(s) 2022

Abstract

As a fundamental data structure, graphs are ubiquitous in various applications. Among all types of graphs, signed bipartite graphs contain complex structures with positive and negative links as well as bipartite settings, on which conventional graph analysis algorithms are no longer applicable. Previous works mainly focus on unipartite signed graphs or unsigned bipartite graphs separately. Several models are proposed for applications on the signed bipartite graphs by utilizing the heuristic structural information. However, these methods have limited capability to fully capture the information hidden in such graphs. In this paper, we propose the first graph neural network on signed bipartite graphs, namely Polarity-based Graph Convolutional Network (PbGCN), for sign prediction task with the help of balance theory. We introduce the novel polarity attribute to signed bipartite graphs, based on which we construct one-mode projection graphs to allow the GNNs to aggregate information between the same type nodes. Extensive experiments on five datasets demonstrate the effectiveness of our proposed techniques.

Keywords Graph neural network · Signed bipartite graph · Sign prediction

✉ Hanchen Wang

✉ Chen Chen
chenc@zjgsu.edu.cn

Xianhang Zhang
xianhang.zhang@unsw.edu.au

Xiaoyang Wang
xiaoyangw@zjgsu.edu.cn

Wenjie Zhang
zhangw@cse.unsw.edu.au

¹ University of New South Wales, Sydney, Australia

² Zhejiang Gongshang University, Hangzhou, China

1 Introduction

Recently, graph-structured data is becoming increasingly ubiquitous, especially with the spreading popularity of the e-commerce platforms and social networks [32, 33]. These networks can be modeled as signed graphs whose edges have either positive or negative signs. Great research efforts have been spent on the unipartite signed graphs [4, 18, 22, 25, 26, 37]. However, as a common form of signed graphs, the signed bipartite graphs have been overlooked by the research community. Signed bipartite graphs have two independent partitions of nodes and the edges are only formed between the nodes of different types. These graphs are prevalent across many domains. For example, on e-commerce platforms, such as Amazon, the buyers can provide positive or negative reviews towards a product.

Previous works primarily focus on the unipartite signed graphs or bipartite graphs separately [2, 5, 31], which are the graphs without node partition or edge sign information. These methods are unable to handle the complexities brought by both bipartite and signed settings. There have been some methods proposed for signed bipartite graphs based on the balance theory [6, 8]. However, these methods are unable to fully exploit the structural, attribute and high-order information by simply performing conventional random walk, which limits their performance on the sign prediction task. With the growing popularity of graph neural networks (GNNs), a variety of network embedding and GNN-based methods are developed for unipartite signed graphs and unsigned bipartite graphs. These models also lack the capability to fully preserve the information of negative links and node partitions. For example, the graph neural networks for signed graphs cannot capture the similarity between nodes in the same node partition of signed bipartite graphs. Similarly, GNNs designed for unsigned bipartite graphs [9, 21] will aggregate the neighbor information in the same way for both positive and negative edges. As a result, they totally ignore the sign information and cannot be used for the sign prediction task. Unsatisfactory results are observed in our initial attempt to directly apply these models on signed bipartite graphs. Therefore, the graph neural network on the signed bipartite graphs cannot be carried out by simply applying existing models.

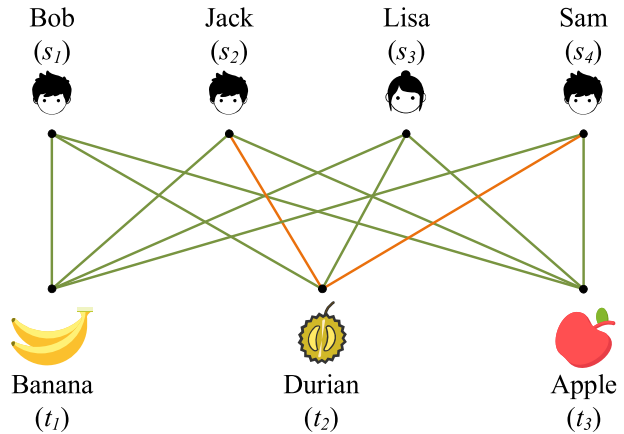
Motivated by the above observations, in this paper, we design a novel graph neural network on the signed bipartite graphs by integrating the proposed polarity attribute, named Polarity-based Graph Convolutional Network (PbGCN). PbGCN first obtains the polarity value for each node, which describes others' opinions towards this node.

For example, in Figure 1, people have polarized opposite opinions toward durian (t_2), even though they may share similar dietary preferences, such as they all love banana (t_1). Consequently, durian (t_2) and banana (t_1) will have small and large polarity values, respectively.

The polarity value paves the way for PbGCN to perform one-mode projection [38] by adding edges between nodes of the same type that share similar polarity values. Then, PbGCN is able to aggregate the information via the established edges directly, even if the nodes in the edges have high proximity with each other in the original graph. Based on the above ideas, we build a graph neural network for signed bipartite graphs, which can significantly boost the performance of the sign prediction task compared with the baseline approaches. The main contributions of the paper are summarized as follows.

- To the best of our knowledge, this is the first work that develops a graph neural network model on the signed bipartite graph to solve the sign prediction task.
- Our model combines the advantages of balance theory and polarity information, which describes the controversy of opinions to the node. Thus, it allows the graph neural network to perform aggregation between the nodes of the same type in signed bipartite graphs directly.

Fig. 1 Example for Signed Bipartite Graph



- Extensive experiments conducted on both real-life and synthetic graphs demonstrate that PbGCN outperforms the state-of-the-art methods for sign prediction.

Roadmap. The rest part of the paper is organized as follows: Section 2 introduces highly related work of this paper (Section 2.1), and also given the key definitions and important notations used in this paper (Section 2.2). Section 3 describes our proposed model in detail. Specifically, Section 3.1 gives the overall framework of the model and other subsections describe every aspect of the model thoroughly. Section 4 consist the empirical evaluations of the proposed model, where Section 4.1 introduces the datasets and baseline methods used in experiments, Section 4.2 is the experiment settings and Sections 4.3 and 4.5 illustrate the experiment results including the parameter sensitivity analysis. Finally, the paper concludes in Section 5.

2 Related work and preliminaries

Recently, with the increasing popularity of social networks and e-commerce platforms, the signed bipartite graphs have become much more ubiquitous. With this trend, there are a great number of researches for graph analytics proposed to solve the specific tasks on the signed bipartite graphs, such as link prediction, sign prediction, node classification, etc. Besides, due to the significant success of the graph neural network (GNN) on various graph analysis tasks, some GNN-based models are also proposed for bipartite graphs. In this section, we introduce some related work with important definitions and notations.

2.1 Related work

2.1.1 Analysis on relatively simple graphs

Numerous research works have proposed outstanding methods on graphs with relatively simple structures, i.e., unipartite signed graph and bipartite signed graph [2, 18, 22, 31, 37]. Among them, [18, 22, 37] focus on the analysis of unipartite signed graphs. These studies carefully and effectively analyze the positive and negative relationships among the nodes of signed social networks. They point out the patterns or regularities that frequently appear in signed social networks. As a result, they achieve high accuracy on the sign prediction task, which is useful in

predicting whether users are more likely to have positive or negative relationships with each other in social networks. In addition, the method proposed by [2] can be used on the structure of the bipartite graph and can efficiently reproduce its degree distribution as well as the degree-wise metamorphosis coefficients. What's more, [31] presents a very advanced method for calculating the number of butterflies in a bipartite graph. These methods cannot be directly applied for analysis of signed bipartite graphs due to their unipartite or unsigned settings respectively.

2.1.2 Methods for signed bipartite graph

Aiming to fill the gap mentioned in the previous subsection, a variety of heuristic and theoretical models [1, 3, 6, 8, 11, 13, 17, 24] are proposed for the applications on the signed bipartite graphs such as sign prediction, link prediction and recommendation. In these efforts, [6, 8, 13] analyzed the voting records of representatives from the U.S. Congress by using balance theory and achieved well predictions of the voting situation. In addition, there are other methods that implement the analysis of signed bipartite graphs based on the signed random walk with restart [17], signed multiplicative rank propagation [24], projection method [11] and linear algorithm [1]. Furthermore, [3] develops a heuristic solution and obtains an outstanding performance on the node classification in signed bipartite networks.

2.1.3 GNNs on bipartite graph

Recently, with the increasing popularity and development of network embedding [10, 23] and graph neural networks (GNN) [12, 16, 27], methods for solving problems on bipartite graphs using neural networks continue to be proposed by numerous studies [9, 19, 21, 28, 30, 34]. Among them, [9] proposes a novel framework for cancer survival prediction, [19, 21] propose novel methods to handle large-scale e-commerce tasks by analyzing the hierarchical structures and using GNNs in bipartite graphs. [34] proposes an analysis that provides insights into better extracting and fusing information from the protein–protein interaction network for drug repurposing. Furthermore, [7, 14, 15, 20, 35] designed representation learning models [29, 36] that are able to preserve both positive and negative link information within the signed graphs. These methodologies are lacking the capability to fully exploit the structural and attribute information that lies in the signed bipartite graphs. In this paper, we proposed the first graph neural network for signed bipartite graphs that is able to address the above issues and improve the sign prediction performance.

2.2 Preliminaries

There are two key definitions in this paper and Table 1 summarized the important notations frequently used throughout the paper.

Definition 1 (Signed Bipartite Graph) A bipartite graph has two separate vertex sets, each of which only has connections with the vertices in the other vertex set. A signed bipartite graph can be denoted as $G = (\mathcal{V}_S, \mathcal{V}_T, \mathcal{E}^+, \mathcal{E}^-)$, where $\mathcal{V}_S = \{s_1, s_2, \dots, s_m\}$, $\mathcal{V}_T = \{t_1, t_2, \dots, t_n\}$ are the mutually exclusive node sets. $\mathcal{E}^+ \subset \mathcal{V}_S \times \mathcal{V}_T$ and $\mathcal{E}^- \subset \mathcal{V}_S \times \mathcal{V}_T$ are the positive and negative edges that connect nodes between two partitions, where $\mathcal{E}^+ \cap \mathcal{E}^- = \emptyset$.

Definition 2 (One-Mode Projection) [38] One-Mode Projection on the bipartite graphs aims to construct a projection graph that creates links between the nodes of the same type, *i.e.*, to build graphs $G_S = (\mathcal{V}_S, \mathcal{E}_S)$, $G_T = (\mathcal{V}_T, \mathcal{E}_T)$ where $\mathcal{E}_S \subset \mathcal{V}_S \times \mathcal{V}_S$ and $\mathcal{E}_T \subset \mathcal{V}_T \times \mathcal{V}_T$.

Definition 3 (Polarity Value) Polarity value is used in our model to measure the disagreement of attitude of the neighbors of one vertex towards it. Formally, in this work, the polarity value is defined as: $r_{v_i} = 1 - \eta_{v_i}^{DS} / \eta_{v_i}^{all}$, where $\eta_{v_i}^{DS} = |\mathcal{N}_{v_i}^+| \times |\mathcal{N}_{v_i}^-|$ is the product of the number of positive and negative neighbors of v_i , and $\eta_{v_i}^{all}$ is the number of all possible neighbor pairs of v_i . We also regularize the polarity value as: $|\tau_{v_i}| = \frac{r_{v_i} - \min(r_{v_i})}{\max(r_{v_i}) - \min(r_{v_i})}$.

Definition 4 (Balance Theory) Balance theory defined that a cycle in signed networks with an even number of negative links is balanced, which is typically stated as “a friend of my friend is my friend” while an “enemy of my friend is my enemy”.

Typically, the one mode projection is performed based on the number of common neighbors of two nodes of the same type.

Example 1 One-mode projection. Figure 2 is an example of a one-mode projection of the signed bipartite graph in Figure 1. In this figure, both s_1 and s_3 are connected with t_1, t_2 and t_3 with positive signed edges, so s_1 and s_3 have 3 neighbors that are connected in the same way. When the threshold used to judge whether a connection should be created is 3, s_1 and s_3 should be connected. Similarly, s_2 and s_4 are connected to both t_1 and t_3 with positive edges and to t_2 with negative edges at the same time, so s_2 and s_4 should also be connected. t_1 and t_3 have 4 common neighbors with the same sign and will also be connected. However, if the number of common neighbors with the same sign is less than 3 like s_1 and s_2 , no edge will be established.

3 Model

In this section, we introduce the details of our proposed Polarity-based Graph Convolutional Network (PbGCN). In Section 3.1, we give the overall introduction of the framework of our proposed PbGCN. Section 3.2 describes how the polarity attribute is captured

Fig. 2 Example for one-mode projection

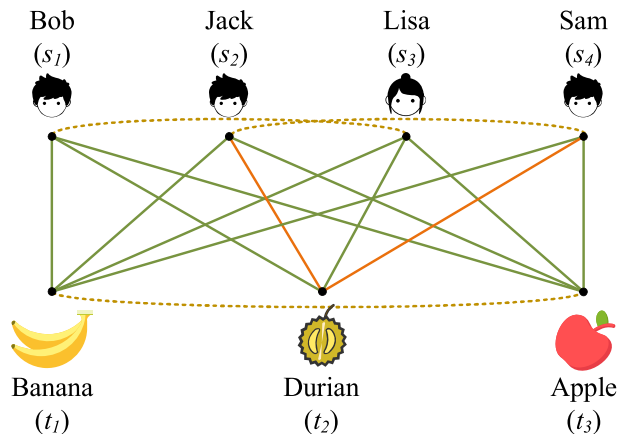


Table 1 Notation Table

Notation	Description
G	Signed bipartite graph.
\mathcal{V}, \mathcal{E}	Nodes and edges in Graph.
A, D	Adjacency matrix and diagonal degree matrix of G .
B	Bi-adjacency matrix.
P	Adjacency matrix of the same type nodes.
I	Identity matrix.
W	Weight matrix in neural network.
H	Hidden feature matrix.
\mathcal{N}	Neighborhood node set.
M	Weighted edge matrix.
c	Random walk restart parameter.
λ	Weight coefficient for bi-adjacency matrix.
τ_{init}, τ	Initial polarity value and polarity attribute.
σ	Nonlinear activation function.
η	Number of neighboring node pairs.
γ, δ	Threshold used to control the addition of connections.
Y	Result of the sign prediction.
\mathcal{L}	Loss function.

in our model, based on which we perform the one-mode projection to establish the links between nodes in the same type. Section 3.3 introduces the graph convolution operations utilized in our model that aggregates the information along the edges in the graph after one-mode projection. Section 3.4 gives the learning objectives used to optimize our model. Finally, the balance theory-based sign prediction is introduced in Section 3.5.

3.1 Framework

Our proposed PbGCN aims to preserve the universal relationship between neighboring nodes, thus achieves higher accuracy in the sign prediction task for signed bipartite graphs compared to the state-of-the-art heuristic methods. PbGCN is a GNN and balance theory-based model which combines their advantages. The framework of PbGCN is shown in Figure 3. We first get the initial structural information of the graph, which helps to summarize the polarity information and add signed weighted edges between the same-type nodes for one-mode projection. The added edges are superimposed into positive and negative adjacency matrices respectively. Afterwards, graph neural network is performed to obtain the prediction τ , which is then combined with the results Y_b obtained based on balance theory for the down-stream prediction task.

3.2 Polarity attribute

Firstly, we need an initial polarity value τ_{init} from the structure of the given graph to perform one-mode projection and guide our training process. The core idea is to capture the distribution of edge signs between each node and its neighbors. For node $v_i \in \mathcal{V}$ in the bipartite

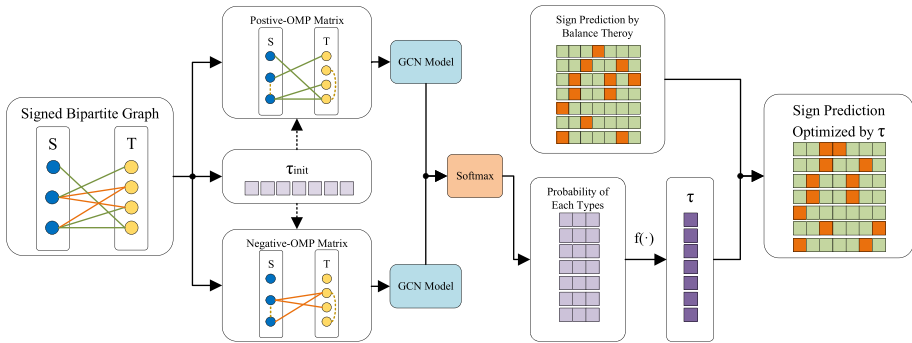


Fig. 3 The Framework of Proposed PbGCN

graph G , we simply compute the possible number of neighboring node pairs connected by different signs as:

$$\eta_{v_i}^{DS} = |\mathcal{N}_{v_i}^+| \times |\mathcal{N}_{v_i}^-|, \tag{1}$$

where $|\mathcal{N}_{v_i}^+|$ and $|\mathcal{N}_{v_i}^-|$ denotes the number of positive and negative neighbors of v_i respectively. For instance, in Figure 1, Jack(s_2) has two positive links and one negative link, thus $|\mathcal{N}_{s_2}^+| = 2$ and $|\mathcal{N}_{s_2}^-| = 1$. We can also compute the number of all possible pairs between the neighbor of v_i using the following equation:

$$\eta_{v_i}^{all} = \frac{|\mathcal{N}_{v_i}|!}{2 \times (|\mathcal{N}_{v_i}| - 2)!}, \tag{2}$$

where $|\mathcal{N}_{v_i}|$ denotes the total number of v_i 's neighbors. Obtained the number of possible pairs, we can get the dissimilar connection rate using the fraction between the numbers of different signed neighbor pairs and all possible pairs: $r_{v_i} = 1 - \eta_{v_i}^{DS} / \eta_{v_i}^{all}$. It is obvious that $|\mathcal{N}_{v_i}|$, r_{v_i} is smaller with more polarized distributed edge signs, i.e., $|\mathcal{N}_{v_i}^+|$ is closed to $|\mathcal{N}_{v_i}^-|$, with the total number of neighbors fixed.

We normalize the dissimilar rate by:

$$|\tau_{v_i}| = \frac{r_{v_i} - \min(r_{\mathcal{V}})}{\max(r_{\mathcal{V}}) - \min(r_{\mathcal{V}})}, \tag{3}$$

where $r_{\mathcal{V}}$ is the rate set of all nodes \mathcal{V} and $|\tau_{v_i}|$ is the normalized value. The sign of τ_{v_i} is determined by the number of positive and negative neighbors of v_i , i.e., $\tau_{v_i} > 0$ if $|\mathcal{N}_{v_i}^+| > |\mathcal{N}_{v_i}^-|$ and vice versa. We denote this initial polarity value τ as τ_{init} in the rest of this paper. τ_{init} directly comes from the neighbor information which captures abundant structural information. However, τ_{init} is hard to be utilized as a training objective or directly used for sign prediction. Therefore, we convert τ_{v_i} to a label y_i by thresholding. The nodes in the given signed bipartite graph are categorized into three classes according to polarity values: *positive tendency nodes*, *negative tendency nodes* which are more likely to have positive or negative links respectively, and *polarized nodes* which have no obvious tendency to form positive or negative links. These labels give a qualitative training target for

GNN, and as a result, our model is able to capture the graph information on a learning basis rather than simply use the values from heuristic statistics.

Since the bipartite graphs have no edge connected between nodes within the same type, in order to allow neighbor aggregation between nodes of the same type, we perform one-mode projection to add edges between these same type nodes according to their polarity values. The main idea for creating such edges is to consider the closeness of τ_{v_i}, τ_{v_j} of the same type nodes v_i, v_j . When their gap is less than a threshold γ , we create a weighted edge $M_{ij}^\tau \in (0, 1]$ between them:

$$M_{ij}^\tau = 1 - \left(\left| \tau_{v_i} - \tau_{v_j} \right| \times \frac{1}{\gamma} \right). \tag{4}$$

This weight is determined by the similarity of their polarity value τ_{init} , and the smaller the gap, the greater the weight. After the pair-wise comparison of τ_{init} between nodes in the same partition, we can form the one-mode projection (OMP) graphs M_S^τ and M_T^τ for node sets S and T , respectively. We form an adjacency matrix that involves all nodes in the graph as $M^\tau = \begin{bmatrix} M_S^\tau & \mathbf{0} \\ \mathbf{0} & M_T^\tau \end{bmatrix}$ for the aggregation in graph neural network. The algorithm for one-mode projection is summarized in Algorithm 1, where line 2 is the computation of the number of positive edges, the number of negative edges and the total number of edges connected to node v . Lines 17 to 19 are used to avoid the self-loop.

Algorithm 1 Build one-mode projection matrix M^τ .

```

Input: same type of nodes  $\mathcal{V}$ 
Data: number of different sign combination  $\eta^{DS}$ , number of all combinations  $\eta^{all}$ , rate of same sign connection  $r$ , type of node  $\theta$ , initial polarity value  $\tau_{init}$ 
Output: one-mode projection matrix  $M^\tau$ 
1 for each  $v \in \mathcal{V}$  do
2    $|\mathcal{N}_v^+|, |\mathcal{N}_v^-|, |\mathcal{N}_v| \leftarrow \text{EdgeNum}(v)$ ;
3   if  $|\mathcal{N}_v^+| > |\mathcal{N}_v^-|$  then
4      $\theta_v \leftarrow 1$ ;
5   else
6      $\theta_v \leftarrow -1$ ;
7   end
8    $\eta_v^{DS} \leftarrow |\mathcal{N}_v^+| \times |\mathcal{N}_v^-|$ ;
9    $\eta_v^{all} \leftarrow \frac{|\mathcal{N}_v|!}{2 \times (|\mathcal{N}_v^-| - 2)!}$ ;
10   $r_v \leftarrow 1 - \frac{\eta_v^{DS}}{\eta_v^{all}}$ ;
11 end
12 for each  $v \in \mathcal{V}$  do
13    $\tau_{init}^v \leftarrow \theta_v \times \frac{r_v - \min(r_{\mathcal{V}})}{\max(\mathcal{V}) - \min(\mathcal{V})}$ ;
14 end
15 for each  $v_i \in \mathcal{V}$  do
16   for each  $v_j \in \mathcal{V}$  do
17     if  $v_i$  is  $v_j$  then
18       continue;
19     end
20     if  $|\tau_{init}^{v_i} - \tau_{init}^{v_j}| < \gamma$  then
21        $M_{ij}^\tau \leftarrow 1 - (|\tau_{v_i} - \tau_{v_j}| \times \frac{1}{\gamma})$ ;
22     end
23   end
24 end
25 return  $M^\tau$ 

```

3.3 Graph convolutional layers

In PbGCN, we use the graph convolutional network (GCN) to perform neighbor aggregation. The overall idea of GCN aggregation is to rely on the adjacency matrix to filter out the neighboring nodes of each node, then weight the feature vectors of these neighboring nodes and sum them up. The activation function is then used on them to update feature vectors, so that the similar nodes have similar feature vectors. Besides, in order to retain the feature vectors of the original nodes, a diagonal identity matrix is added to the original adjacency matrix for self-loop. And to ensure that all nodes aggregate the feature vectors of neighboring points at the same scale, a normalization operation is required for each row of the adjacency matrix. As a result, each layer in the original GCN algorithm can be formulated as:

$$H_p^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}), \tag{5}$$

where $H^{(l)}$ is the hidden feature vector at l^{th} layer, \tilde{A} is the adjacency matrix A with self-loop as: $\tilde{A} = A + I_N$, \tilde{D} is a diagonal degree matrix of \tilde{A} where $D_{ii} = \sum_j \tilde{A}_{ij}$, and $W^{(l)}$ is a trainable weight matrix for l^{th} layer. It is clear that the original GCN will suffer from poor performance if it is directly applied to the signed bipartite graph. As a result, there are two major challenges to apply graph convolutional network on the signed bipartite graphs: it is difficult to aggregate the feature vectors of nodes of the same type, on the other hand, the primitive aggregation approach cannot distinguish the sign information of the edges.

In order to solve the above problems, we have obtained τ_{init} in the above steps and used it to build a one-mode projection graph M^τ to allow nodes of the same type with potentially similar properties to be connected and provide weights according to the polarity values to ensure better aggregation results. To address the second problem, we divide the original signed adjacency matrix into positive-edge adjacency matrix and negative-edge adjacency matrix. We design the graph neural network on these adjacency matrices separately to ensure that the information of positive and negative edges can each be fully utilized, instead of totally ignoring the sign information or directly using the original adjacency matrices with 1 and -1 weights, which cannot fully consider the sign information during the aggregation process and result in unsatisfactory aggregation results. The aggregation process can be summarized in the following equations:

$$H_p^{l+1} = \sigma(\tilde{D}_p^{\tau-\frac{1}{2}} \tilde{M}_p^\tau \tilde{D}_p^{\tau-\frac{1}{2}} H_p^{(l)} W_p^{(l)}) \tag{6}$$

$$H_n^{l+1} = \sigma(\tilde{D}_n^{\tau-\frac{1}{2}} \tilde{M}_n^\tau \tilde{D}_n^{\tau-\frac{1}{2}} H_n^{(l)} W_n^{(l)}) \tag{7}$$

$$\hat{Y} = softmax(H_p^L + H_n^L), \tag{8}$$

where $\tilde{M}_p^\tau = \begin{bmatrix} M_S^\tau & M_p^\tau \\ M_n^\tau & M_T^\tau \end{bmatrix} + I_M$, $\tilde{M}_n^\tau = \begin{bmatrix} M_S^\tau & M_n^\tau \\ M_n^\tau & M_T^\tau \end{bmatrix} + I_M$, and $\tilde{D}_{pii}^\tau = \sum_j \tilde{M}_{pij}^\tau$, $\tilde{D}_{nii}^\tau = \sum_j \tilde{M}_{nij}^\tau$, σ denotes the activation function, $H_p^{(0)}$ and $H_n^{(0)}$ are the input features of the graph, and $\hat{Y} \in (0, 1)^{|V| \times 3}$ is the output tensor for the label prediction. It is worth noting that the softmax function is used to the sum of two sets of feature vectors in layer L to obtain \hat{Y} , while activation function is applied when aggregating the feature vectors in other layers.

3.4 Learning objectives

With the obtained label prediction, we need to set a loss function as the training target of the model. As mentioned in Section 3.2, we have generated a label \mathbf{y} for each node to describe the probability of this node to form a new positive/negative link. Based on that, we design the cross-entropy loss function as our learning objective:

$$\mathcal{L} = - \sum_{v_i \in \mathcal{V}_{train}} \sum_k^K y_k(v_i) \log(\hat{y}_k(v_i)), \tag{9}$$

where \mathcal{V}_{train} is the node set involved in the training edge set, and K is the number of labels, in our case $K = 3$. The prediction result $\hat{\mathbf{Y}}$ is then mapped as attribute vector τ for all nodes by $f(\cdot) : \mathbb{R}^{|\mathcal{V}| \times 3} \rightarrow \mathbb{R}^{|\mathcal{V}| \times 1}$

$$f(\hat{\mathbf{Y}}) = \frac{\hat{\mathbf{Y}}_{pos} - \hat{\mathbf{Y}}_{neg}}{\hat{\mathbf{Y}}_{pos} + \hat{\mathbf{Y}}_{pol} + \hat{\mathbf{Y}}_{neg}}, \tag{10}$$

where $\hat{\mathbf{Y}}_{pos}, \hat{\mathbf{Y}}_{pol}, \hat{\mathbf{Y}}_{neg}$ represent the predicted probability of a node being of type positive tendency nodes, polarized nodes and negative tendency nodes.

The mapping function f could assign 1 as predicted polarity value to the nodes that are more likely to have positive edges with similar neighboring nodes and assign -1 to the other nodes. We then use the computed polarity value to optimize the balance theory-based prediction.

3.5 Balance theory-based sign prediction

Inspired by the application of balance theory on signed bipartite graphs [6], we design the heuristic prediction process based on the balance theory. According to the balance theory, we add the signed edges between two nodes of the same type with the following equation

$$P_{ij} = \begin{cases} 0 & \delta_n < \eta_{ij}^A - \eta_{ij}^D < \delta_p \\ \eta_{ij}^A - \eta_{ij}^D & otherwise \end{cases} \tag{11}$$

\mathbf{P} is an adjacency matrix storing edges of the same type nodes. Counting the number of neighbors with the same sign edges η_{ij}^A and the number of neighbors with different sign edges η_{ij}^D which are linked to both nodes v_i and v_j to create edges. When $\eta_{ij}^A - \eta_{ij}^D$ is greater or less than a certain threshold, v_i and v_j will be connected by an edge with weight $\eta_{ij}^A - \eta_{ij}^D$. After adding edges between the same type nodes, and then combining the weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{T}|}$ of the original graph, the adjacency matrix $\mathbf{M}^b \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ that can be used for random walk can be constructed:

$$\mathbf{M}^b = \begin{bmatrix} \hat{\mathbf{P}}_S & \lambda \hat{\mathbf{B}} \\ \lambda \hat{\mathbf{B}}^T & \hat{\mathbf{P}}_T \end{bmatrix}, \tag{12}$$

where \mathbf{M}^b is the matrix constructed according to balance theory, and $\hat{\mathbf{B}}$ is the row normalized bi-adjacency matrix, i.e., $\hat{\mathbf{B}}_{ij} = \mathbf{B}_{ij} / \sum_k |\mathbf{B}_{ik}|$ where $\mathbf{B} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{T}|}$ is the adjacency matrix of the signed bipartite graph, $\lambda = 2$ is the weight coefficient for $\hat{\mathbf{B}}$, and we also normalize matrix \mathbf{P} and \mathbf{M}^b to get matrix $\hat{\mathbf{P}}$ and $\hat{\mathbf{M}}^b$. The relationship between any three nodes is consistent according to balance theory when and only when the number of negative

edges linked to these three nodes is an even number. 4 We then perform two-step random walk to capture more information, and the signs of the edges that connect the starting and ending nodes are determined by the signs of visited edges in these random walk paths. As a result, the prediction matrix Y^b under balance theory can be obtained by performing random walk on the normalized adjacency matrix \hat{M}^b :

$$Y^b = (1 - c)(I - c\hat{M}^b)^{-1}, \quad (13)$$

where $(1 - c)$ is the random walk restart probability.

After obtaining the prediction matrix Y^b that relies on the balance theory, we add it with weighted τ before the prediction. In order to enable τ to add directly to the matrix Y^b , we need to do the expansion operation on τ first, that is to say, copy $\tau \in \mathbb{R}^{|\mathcal{V}|}$ for $|\mathcal{V}|$ times to get $\tau_{expand} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$. To further improve the prediction results with the help of τ , in practice, we will first compute the mean value \bar{y} of Y^b , and then multiply \bar{y} with τ_{expand} . On this basis, the optimized prediction matrix Y_τ is obtained as:

$$Y_\tau = Y^b + \omega \cdot \bar{y} \cdot \tau_{expand}, \quad (14)$$

where ω is the adjustable weight. Finally, we use the sign of the value at the corresponding position in Y_τ as the sign prediction of the given edge.

4 Experiment

In this section, we empirically prove and evaluate the effectiveness of polarity value τ and GNN in the optimization of sign prediction performance. We attempt to answer the following two research questions:

- **Q1.** Has the utilization of polarity property and graph neural network led to improved prediction results?
- **Q2.** How much the proposed method improves the baselines?

To answer the above questions, experiments are conducted on three real-world and two synthetic signed bipartite graphs to evaluate the performance of the baseline methods for sign prediction task. To further investigate the advancement of our proposed method, we also perform a parameter sensitivity analysis on PbGCN. The datasets and baselines used are listed in Section 4.1, the experiment settings are introduced in Section 4.2, the results comparisons are shown in Section 4.3 and the parameter analysis results in Section 4.5.

4.1 Datasets and baselines

The experiments are conducted on the following real-life and synthetic datasets:

- **Senate** and **House** [6] are directed graphs that contain the role call votes records from the 1st to 10th United States Senate and House of Representatives.
- **Gene**¹ is a undirected graph which describes the regulatory effect of the transcription factor on the regulated gene.

¹ http://regulondb.ccg.unam.mx/menu/download/datasets/files/network_tf_gene.txt

Table 2 Statistics of Datasets

Dataset	S Nodes	T Nodes	Edges
Senate	145	1,056	27,083
House	515	1,281	114,378
Gene	211	1,847	4,670
Synth1	1,428	456	15,255
Synth2	7,372	9,747	31,641

- **Synth1** and **Synth2** are synthetic datasets that we generate by randomly adding positive and negative edges between two randomly separated node sets.

The statistics of these datasets are summarized in Table 2.

Two set of synthetic data are generated from two undirected bipartite graphs: Robertson (1929)² and Youtube Group Memberships³. We randomly add positive and negative labels to the edges of these two bipartite graphs to generate symbolic bipartite graphs, where the probability of obtaining a positive label for each edge is 35%. It should be noted that since the youtube dataset is too large, we only take its first 230,000 edges to generate the dataset. The number of edges added by the one-mode projection is usually under five times of the number of original edges in the graph.

We follow the settings in [6] to randomly split 85%, 5% and 10% of the edge labels in the original datasets for training, validation and testing. More specifically, during the training process, only training set is accessible, and when evaluating the sign prediction performance, we compare the ground truth signs of the edges in the test set with the predicted sign suggested by the models.

We compare the proposed framework with the following baselines:

- **SBRW**⁴ [6] is a sign prediction algorithm on signed bipartite graphs by performing random walk on the adjacency matrix constructed according to the balance theory.
- **SBRW+GCN** and **SBRW+GAT** are the optimized models that add the learned predictions by directly performing GCN [16] or GAT [27] in the original signed bipartite graph to the prediction value of SBRW. Please note that these baselines do not consider the sign information.
- **SBRW+ τ_{init}** is the model that utilizes τ_{init} to boost the prediction performance of SBRW by simply adding τ_{init} to the results of SBRW.

4.2 Experimental settings

We designed the polarity-based graph neural network for edge sign prediction on signed bipartite graphs to improve the prediction accuracy compared to the conventional balance theory-based methods. In our proposed model, we use two-layer GCN model, choose *Adam* as the optimizer with the learning rate $lr = 0.01$, and we use the randomly generated features

² https://iwdb.nceas.ucsb.edu/html/robertson_1929.html

³ <http://socialnetworks.mpi-sws.org/data-imc2007.html>

⁴ <https://github.com/DSE-MSU/signed-bipartite-networks>

as the input of GCN. The dimension of input feature is 1000, the representation feature's dimension is 64, and the final output dimension for prediction is 3. The random walk restart probability $1 - c$ is set to 0.15. Since the sign prediction problem can be formulated as a binary classification task, we select the widely-used Area Under Curve (AUC) and F1-score as the evaluation metrics. The bold entries in Tables 3 and 4 indicate the best results.

4.3 Sign prediction results

The experiment results are illustrated in Table 3. Our proposed PbGCN outperforms all baseline methods across all datasets. Compared to the state-of-the-art sign prediction model SBRW, PbGCN could significantly improve AUC and F1-score by up to 6.54% and 16.71% respectively. It is worth mentioning that SBRW guesses the sign based on the proportion of positive and negative edges in the dataset if it encounters a result that cannot be distinguished to make a decision. Our proposed PbGCN could avoid this situation, thus achieves better performance. The improvement compared to SBRW+GCN and SBRW+GAT indicates that the edges constructed by the polarity attribute and consideration of positive and negative links could consistently enhance the prediction accuracy. These results provide a positive answer to our first research question: the utilization of polarity value and GNNs indeed provides an empirically verifiable improvement.

Moreover, PbGCN reaches better performance in comparison to SBRW+ τ_{init} . The reason is that τ_{init} only contains the node attribute rather than its neighbor information. Therefore, PbGCN could easily make a more accurate prediction by the aggregation of neighbor information in the signed bipartite graph with added τ_{init} -based edges.

Overall, it can be summarized that PbGCN achieves the superior performance because of the utilization of polarity attribute, one-mode projection and GNN which aggregates the neighbor information with consideration of both positive and negative links.

4.4 Ablation study

We also perform the ablation study to show the effectiveness of our model. We replace the graph neural network in the paper with the basic neural network (*i.e.*, multilayer perceptron MLP), denoted as NN-Baseline. The results are illustrated in Table 4. We can find in the results that, in most datasets, PbGCN outperforms the NN-Baseline, which indicate the effectiveness of the design of PbGCN.

4.5 Parameter analysis

We conduct the parameter analysis experiments on the following parameters: weight coefficient ω of learned τ , threshold γ , number of GCN layers L and learning rate lr . We perform all the parameter analysis experiments on the Gene dataset. The analysis results are shown in Figure 4.

For PbGCN, the most important parameter is the weight ω of τ . We tested seven ω values varying between 0 and 0.3 with an interval of 0.05 as our weight. The result of tests can be seen in Figure 4(a) for AUC and Figure 4(b) for F1. When the weight ω is set to 0, the prediction is made fully by the heuristic method based on balance theory. The best AUC value is achieved when the weight is 0.15. At the same time, F1-score reaches the salient point with a weight of 0.15. The performance of PbGCN worsens with the increment of ω . Therefore, for the Gene dataset, ω was set to be around 0.15.

Table 3 Experiment results for sign prediction

Dataset	Algorithm	AUC	F1	#Guess
Senate	SBRW	79.398%	81.077%	0
	SBRW+GCN	79.406%	81.388%	0
	SBRW+GAT	79.365%	81.037%	0
	SBRW+ τ_{init}	79.662%	81.551%	0
	PbGCN	80.426%	82.282%	0
House	SBRW	81.143%	82.504%	0
	SBRW+GCN	81.315%	82.868%	0
	SBRW+GAT	81.261%	82.653%	0
	SBRW+ τ_{init}	81.216%	82.637%	0
	PbGCN	81.579%	83.159%	0
Gene	SBRW	67.238%	69.048%	84
	SBRW+GCN	70.122%	68.046%	0
	SBRW+GAT	69.419%	68.151%	0
	SBRW+ τ_{init}	68.963%	68.670%	31
	PbGCN	71.633%	69.725%	0
Synth1	SBRW	50.320%	59.923%	9
	SBRW+GCN	50.519%	69.034%	0
	SBRW+GAT	49.826%	62.297%	0
	SBRW+ τ_{init}	51.466%	66.433%	2
	PbGCN	51.587%	69.939%	0
Synth2	SBRW	50.660%	64.920%	639
	SBRW+GCN	51.043%	66.965%	0
	SBRW+GAT	50.756%	65.550%	0
	SBRW+ τ_{init}	51.653%	64.287%	245
	PbGCN	52.154%	67.163%	0

In addition, the threshold value γ that determines whether an edge is established between two nodes of the same type is also an important parameter. We tested six thresholds $\{0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$, and the results are demonstrated in Figures 4(c) for AUC and Figure 4(d) for F1. We can make a conclusion from the experiment results that PbGCN is robust for F1-score with regards to varying thresholds γ . The best point appears at 0.1 for AUC. As a result, for the Gene dataset, we choose 0.1 as the threshold value to determine whether an edge should be added between the same type nodes.

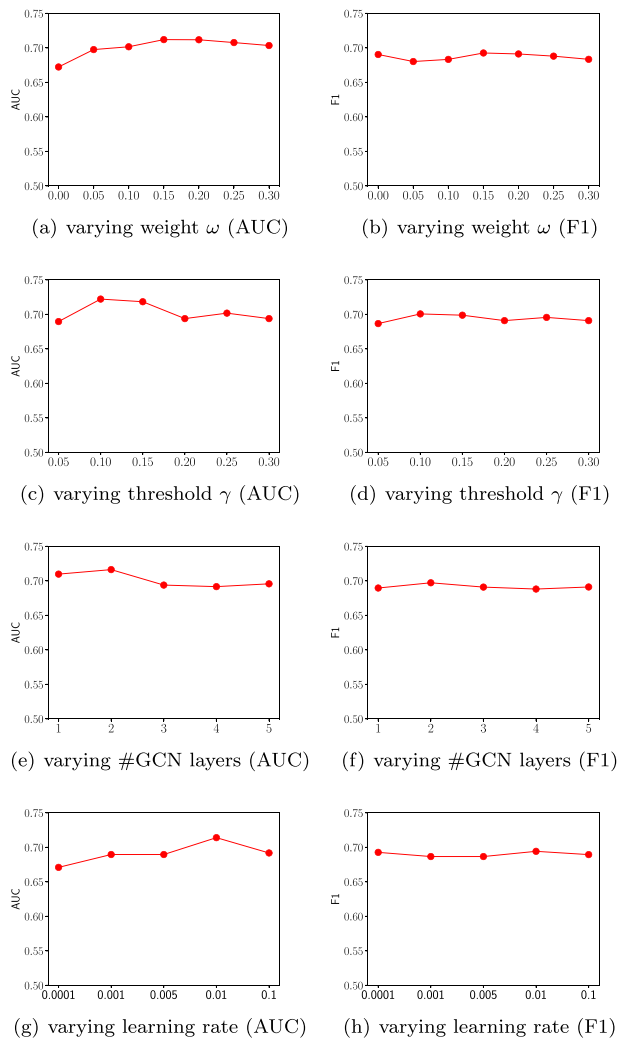
In order to test the impact of the varying number of layers L in the graph convolutional network, we evaluate our model with 1 to 5 GCN layers. The results are illustrated in Figure 4(e) and (f). Interestingly, unlike the fact observed before that deeper graph neural networks have worse performance, the overall impact of the number of GCN layers L is insignificant in our case. Consequently, we set the GCN model with a depth of 2 for higher efficiency while maintaining effectiveness.

Finally, we tested the learning rate lr with five values $\{1 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}, 0.01, 0.1\}$ with results in Figure 4(g) and (h). While proving the robustness of proposed PbGCN, the results also indicate that the salient point for learning rate is 0.01 on the Gene dataset. Taking the experiment results on other datasets into consideration, we finally set the default learning rate of our model to 0.01.

Table 4 Experiment results for ablation study

Dataset	Algorithm	AUC	F1	#Guess
Senate	NN-Baseline	80.103%	81.964%	0
	PbGCN	80.426%	82.282%	0
House	NN-Baseline	81.416%	82.950%	0
	PbGCN	81.579%	83.159%	0
Gene	NN-Baseline	70.166%	68.337%	0
	PbGCN	71.633%	69.725%	0
Synth1	NN-Baseline	52.346%	67.884%	0
	PbGCN	51.587%	69.939%	0
Synth2	NN-Baseline	50.970%	66.523%	0
	PbGCN	52.154%	67.163%	0

Fig. 4 Parameter Analysis Results in PbGCN on Gene Dataset



5 Conclusion

The signed bipartite graphs are becoming more and more ubiquitous in real-life applications, but few research works are conducted due to the complexities brought by the signed links and bipartite settings. In this paper, we propose the first graph neural network on signed bipartite graphs for sign prediction task. The proposed PbGCN first introduces the polarity attribute to the signed bipartite graphs, in order to describe the probability of a node for establishing positive/negative links with others. Based on these polarity values, we construct one-mode projection graphs that enable the developed graph convolutional network to perform neighbor aggregation more effectively on signed bipartite graphs. By integrating the learning-based and balance theory-based predictions, PbGCN boosts the sign prediction accuracy significantly. Comprehensive experiments on three real-life and two synthetic datasets prove the notable improvement of PbGCN compared with the state-of-the-art methods.

Acknowledgements This work is supported by NSFC 61802345, ZJNSF LQ20F020007 and ZJNSF LY21F020012.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Declarations

Conflicts of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Akoglu, L.: Quantifying political polarity based on bipartite opinion networks. In: Proceedings of the International AAAI Conference on Web and Social Media (2014)
2. Aksoy, S.G., Kolda, T.G., Pinar, A.: Measuring and modeling bipartite graphs with community structure. *Journal of Complex Networks* 5(4), 581–603 (2017)
3. Banerjee, S., Sarkar, K., Gokalp, S., Sen, A., Davulcu, H.: Partitioning signed bipartite graphs for classification of individuals and organizations. In: SBP, pp. 196–204. Springer (2012)
4. Chen, C., Wu, Y., Sun, R., Wang, X.: Maximum signed θ -clique identification in large signed graphs. *TKDE* (2021)
5. Chen, C., Zhu, Q., Wu, Y., Sun, R., Wang, X., Liu, X.: Efficient critical relationships identification in bipartite networks. *World Wide Web* (2021)
6. Derr, T., Johnson, C., Chang, Y., Tang, J.: Balance in signed bipartite networks. In: CIKM, pp. 1221–1230. ACM (2019)
7. Derr, T., Ma, Y., Tang, J.: Signed graph convolutional networks. In: ICDM, pp. 929–934. IEEE (2018)
8. Derr, T., Tang, J.: Congressional vote analysis using signed networks. In: ICDM workshop, pp. 1501–1502. IEEE (2018)
9. Gao, J., Lyu, T., Xiong, F., Wang, J., Ke, W., Li, Z.: Mgnn: A multimodal graph neural network for predicting the survival of cancer patients. In: SIGIR, pp. 1697–1700 (2020)
10. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: SIGKDD, pp. 855–864 (2016)

11. Gu, K., Fan, Y., Zeng, A., Zhou, J., Di, Z.: Analysis on large-scale rating systems based on the signed network. *Physica A: Statistical Mechanics and its Applications* **507**, 99–109 (2018)
12. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS, pp. 1024–1034 (2017)
13. Hu, J., Zheng, W.X.: Bipartite consensus for multi-agent systems on directed signed networks. In: 52nd IEEE Conference on Decision and Control, pp. 3451–3456. IEEE (2013)
14. Javari, A., Derr, T., Esmailian, P., Tang, J., Chang, K.C.C.: Rose: Role-based signed network embedding. In: WWW, pp. 2782–2788 (2020)
15. Kim, J., Park, H., Lee, J.E., Kang, U.: Side: representation learning in signed directed networks. In: WWW, pp. 509–518 (2018)
16. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR 2017 (2017)
17. Lee, Y.C., Son, J., Kim, T., Park, D., Kim, S.W.: Exploiting uninteresting items for effective graph-based one-class collaborative filtering. *The Journal of Supercomputing*, 1–20 (2021)
18. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: WWW, pp. 641–650 (2010)
19. Li, C., Jia, K., Shen, D., Shi, C.J.R., Yang, H.: Hierarchical representation learning for bipartite graphs. In: IJCAI, pp. 2873–2879 (2019)
20. Li, Y., Tian, Y., Zhang, J., Chang, Y.: Learning signed network embedding via graph attention. In: AAAI, pp. 4772–4779 (2020)
21. Li, Z., Shen, X., Jiao, Y., Pan, X., Zou, P., Meng, X., Yao, C., Bu, J.: Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications. In: ICDE, pp. 1677–1688. IEEE (2020)
22. Papaikonomou, A., Kardara, M., Tserpes, K., Varvarigou, D.: Edge sign prediction in social networks via frequent subgraph discovery. *IEEE Internet Computing* **18**(5), 36–43 (2014)
23. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: SIGKDD, pp. 701–710 (2014)
24. Shams, B., Haratizadeh, S.: Sibrank: Signed bipartite network analysis for neighbor-based collaborative ranking. *Physica A: Statistical Mechanics and its Applications* **458**, 364–377 (2016)
25. Sun, R., Chen, C., Wang, X., Zhang, Y., Wang, X.: Stable community detection in signed social networks. *TKDE* (2020)
26. Sun, R., Zhu, Q., Chen, C., Wang, X., Zhang, Y., Wang, X.: Discovering cliques in signed networks based on balance theory. In: DASFAA, pp. 666–674 (2020)
27. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. [arXiv:1710.10903](https://arxiv.org/abs/1710.10903) (2017)
28. Wang, H., Lian, D., Liu, W., Wen, D., Chen, C., Wang, X.: Powerful graph of graphs neural network for structured entity analysis. *World Wide Web*, 1–21 (2021)
29. Wang, H., Lian, D., Zhang, Y., Qin, L., He, X., Lin, Y., Lin, X.: Binarized graph neural network. *World Wide Web* **24**(3), 825–848 (2021)
30. Wang, H., Lian, D., Zhang, Y., Qin, L., Lin, X.: Gognn: Graph of graphs neural network for predicting structured entity interactions. [arXiv:2005.05537](https://arxiv.org/abs/2005.05537) (2020)
31. Wang, K., Lin, X., Qin, L., Zhang, W., Zhang, Y.: Vertex priority based butterfly counting for large-scale bipartite networks. *PVLDB* (2019)
32. Wang, X., Zhang, Y., Zhang, W., Lin, X.: Efficient distance-aware influence maximization in geo-social networks. *TKDE* **29**(3), 599–612 (2017)
33. Wang, X., Zhang, Y., Zhang, W., Lin, X., Chen, C.: Bring order into the samples: A novel scalable method for influence maximization. *TKDE* **29**(2), 243–256 (2017)
34. Wang, Z., Zhou, M., Arnold, C.: Toward heterogeneous information fusion: bipartite graph convolutional networks for in silico drug repurposing. *Bioinformatics* **36**(Supplement_1), i525–i533 (2020)
35. Yang, F., Fan, K., Song, D., Lin, H.: Graph-based prediction of protein-protein interactions with attributed signed graph embedding. *BMC bioinformatics* **21**(1), 1–16 (2020)
36. Yang, P., Wang, H., Zhang, Y., Qin, L., Zhang, W., Lin, X.: T3s: Effective representation learning for trajectory similarity computation. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 2183–2188. IEEE (2021)
37. Zhang, T., Jiang, H., Bao, Z., Zhang, Y.: Characterization and edge sign prediction in signed networks. *Journal of Industrial and Intelligent Information* **1**(1), 19–24 (2013)
38. Zhou, T., Ren, J., Medo, M., Zhang, Y.C.: Bipartite network projection and personal recommendation. *Physical Review E* **76**(4), 046115 (2007)