Enterprise Cooperation and Competition Analysis with a Sign-Oriented Preference Network

Le Dai[1], Yu Yin[1], Chuan Qin[1,3], Tong Xu[1*], Xiangnan He[2], Enhong Chen[1,2], Hui Xiong[4*]

[1]School of Computer Science and Techonology, University of Science and Technology of China, [2]School of Data Science, University of Science and Technology of China, [3]Baidu Talent Intelligence Center, Baidu Inc., [4]Rutgers University

{dl123,yxonic}@mail.ustc.edu.cn, {tongxu,cheneh}@ustc.edu.cn,
{chuanqin0426,xiangnanhe}@gmail.com, hxiong@rutgers.edu

ABSTRACT

The development of effective cooperative and competitive strategies has been recognized as the key to the success of many companies in a globalized world. Therefore, many efforts have been made on the analysis of cooperation and competition among companies. However, existing studies either rely on labor intensive empirical analysis with specific cases or do not consider the heterogeneous company information when quantitatively measuring company relationships in a company network. More importantly, it is not clear how to generate a unified representation for cooperative and competitive strategies in a data driven way. To this end, in this paper, we provide a large-scale data driven analysis on the cooperative and competitive relationships among companies in a Sign-oriented Preference Network (SOPN). Specifically, we first exploit a Relational Graph Convolutional Network (RGCN) for generating a deep representation of the heterogeneous company features and a company relation network. Then, based on the representation, we generate two sets of preference vectors for each company by utilizing the attention mechanism to model the importance of different relations, representing their cooperative and competitive strategies respectively. Also, we design a sign constraint to model the dependency between cooperation and competition relations. Finally, we conduct extensive experiments on a real-world dataset, and verify the effectiveness of our approach. Moreover, we provide a case study to show some interesting patterns and their potential business value.

KEYWORDS

Enterprise Analysis, Graph Embedding, Signed Network

1 INTRODUCTION

With the accelerated process of globalization, the relations between enterprises have become closer than ever, brought with it opportunities as well as challenges. Faced with fierce competition, enterprises are devoting much attention to the selection of business partners, in order to achieve success in this globalized world [2]. Thus, the analysis of cooperation and competition has been a crucial task, beneficial to both enterprises and third-party investors with helpful development guidances and insightful investment cues [25].

In the literature, large efforts have been made on this issue with various techniques. Traditionally, experts of management science usually deal with this problem via empirical study on specific cases [5, 14], in which statistical methods are widely utilized. Specifically, some researches mainly focus on certain pairs of enterprises with cooperation or competition relationship [9, 19]. In these cases, the companies will be treated as separately or rarely connected, and then evaluated via theoretical analysis over quantitative experiments. However, they can hardly provide the overall analysis in consideration of global connections among enterprises. Meanwhile, some other researchers study this issue in the perspective of enterprise network with specific relations like shareholding [3], talent flow [32], etc. Along this line, they represent the company relation network with deep learning modules to support downstream tasks. However, current solutions may fail due to the following two reasons. First, existing studies may fail to automatically analyze strategies about cooperation and competition, integrating heterogeneous enterprise information and multiple relations. It remains unclear how to generate a unified representation for cooperation and competition strategies in a data driven way. Second, since existing studies mainly weigh the connections without distinguishing their attitude (cooperation or competition), the dependency between multiple relations are always ignored. For example, one common situation is that enterprises sometimes seek cooperation with the competitors of their competition candidates [23], as suggested by the idiom "the enemy of my enemy is my friend". Therefore, a more comprehensive solution is still urgently required.

To that end, in this paper, we propose a novel Sign-oriented Preference Network (SOPN) for more effective cooperation and competition analysis. In general, the following major challenges will be addressed. First, heterogeneous enterprise information for describing the profile of enterprises and multiple relations between them should be carefully integrated. Second, a unified representation about cooperation and competition strategies should be generated for a more comprehensive analysis. Third, the mutual dependency between cooperation and competition relations should also be considered. Last but not least, during the representation of enterprise

network, interpretability of cooperation and competition strategies must be ensured to support downstream applications. To deal with these challenges, SOPN generates two sets of preference vectors that represent enterprise strategies for cooperation and competition. Specifically, we first aggregate the company features and company relation graph, and employ a Relational Graph Convolutional Network (RGCN) [22] to generate deep representation of the heterogeneous data. Then, with the embedding vectors derived from RGCN, we leverage the attention mechanism [1] to distinguish the effects of multiple company relations on company strategies during the computation of cooperation preferences and competition preferences. Also, to model the dependency of the cooperation and competition relations, we design a sign constraint based on *signed graph theory* [30]. At last, the model is trained through a hybrid loss function, combining task specific loss with sign constraint and graph decoding loss, to ensure both effective modeling and better interpretability. To be specific, the contributions of this paper can be summarized as follows:

- We propose a novel embedding framework on Sign-oriented Preference Network (SOPN) for generating the unified representation of cooperation and competition strategies, in which heterogeneous information and multiple relations have been effectively integrated.
- The hybrid loss function has been carefully designed, which includes the sign-oriented constraint to describe the dependency among cooperation and competition relations, and the decoding loss to ensure the interpretability of strategies.
- Extensive experiments on real-world enterprise dataset have validated the effectiveness of our solution, and further revealed some interesting discoveries through case study.

2 RELATED WORK

Generally, the related work of our study can be grouped into the following: *Company Analysis* and *Graph Embedding*.

2.1 Company Analysis

The problem of company analysis has been widely studied by experts and researchers, especially in the business and management domains. Existing researches can be divided into the following three categories. The first is based on case studies. For example, Quintana et al. [19] analyzed the effect of co-opetition (shortly for cooperation and competition) strategy on technological diversity and new product development through a panel data of European dedicated biotechnology firms. Also, Gnyawali et al. [9] investigated why and how co-opetition between large firms occurs, evolves, and impacts the participating firms and the industry through a case study of two giant companies in the electronics industry. Those studies requires experts to conduct careful empirical research over specific cases, which can be very labor intensive. The second is conducted through statistic methods. For example, Deng et al. [5] ranked the relative performance of competing companies with modified statistic method TOPSIS and conducted an empirical study of a real case to prove the effectiveness of their approach. And Kung et al. [14] utilized the Globalization Grey Relational Analysis (GRA), to find the significant financial ratio variables and other financial indicators affecting the financial performance of venture capital

enterprises in Taiwan. Although great success has been achieved, those approaches are limited to a few topics like financial performance evaluation [5] and supplier selection [17]. In addition, the third is based on machine learning techniques due to their superiority of capturing complex information. As an example, Zhang et al. [33] incorporated information from job transition records of digital resumes to help sharpen company talent strategy. Those works have achieved great results within specific tasks.

As we can see, despite the great success achieved in enterprise analysis, most of the existing researches are still in the stage of case analysis, lack of quantitative experiments combining various company information and multiple relations. Moreover, none of existing researches obtain a unified representation about cooperation and competition strategies.

2.2 Graph Embedding

To better utilize all the information of the company input, one of our main requirements is to properly generate representations for the company network, preserving heterogeneous company information and company relations.

In recent years, graph embedding methods have attracted increasing attention due to the ubiquity of networks in real world, and are proven to be efficient and effective in network representation. First, to model network relations, some proximity preserving methods have been introduced. Methods based on random walk assume that nodes with the similar network structure have similar vector representation. For example, Perozzi et al. [18] sampled local network structures by random walk, and then utilized skip-gram model to learn the vectorized representation. Some other methods learn node representations based on k-order distance between nodes in network. For example, Tang et al. [24] proposed LINE to explicitly define two functions for preserving first-order and second-order proximities, and minimized the combination of the two. Besides, some methods also incorporate deep learning to obtain high-order nonlinear representation of the local structural context [26–28]. Second, node representations are enhanced to combine extra attribute information. As an example, Kipf et al. [13] proposed Graph Convolutional Network (GCN) based on a first-order approximation of spectral convolutions on graphs to propagate attribute information. Hamilton et al. [11] sampled and aggregated attribute information from neighboring nodes to iteratively generate node embeddings in GraphSage. Third, interactions in multi-relational networks are taken into consideration. For example, Schlichtkrull et al. [22] adapted GCN for highly multi-relational data such as realistic knowledge bases. Zhang et al. [31] proposed HetGNN which aims at representation learning in heterogeneous graphs with multiple types of nodes as well as relations. Moreover, there are also some studies focusing on two specific relations, i.e. those of a signed graph [30], where relations are labeled as either positive or negative [6, 7, 29]. As an example, Derr et al. [6] proposed a node embedding network SGCN for signed graphs, considering different propagation rules and constraints for both signs under balance theory of sign networks.

All the previous studies focus on node embedding and applications based on it, but they do not generate representation for preferences, and cannot be directly applied to our cooperation and

Table 1: Mathematical notations.

Symbol	Description
C	The company set
f_i	Concatenated features of numeric features μ_i and textual features τ_i of company c_i
G	The company network with node set $\mathcal{V}$, labeled edge set $\mathcal{E}$ and relation set $\mathcal{R}$
$\mathcal{N}_i$	The neighboring nodes of node i
emb_i	The embedded vector of company c_i
$\mathcal{P}_i^{coo}, \mathcal{P}_i^{com}$	The sets of cooperation and competition preference vectors of company c_i
N_{coo}, N_{com}	Hyperparameters of the number of cooperation and competition preference vectors

competition analysis problem. Furthermore, most of the previous studies deal with single relation or single form of input, and are not capable for incorporating heterogeneous input and multiple relations in our problem.

3 METHODOLOGY

In this section, we present our method in detail. We first formulate the problem of enterprise cooperation and competition strategy analysis. Then, we give an overview of the model architecture. Afterwards, we describe all the details of the framework we propose.

3.1 Problem Definition

In this subsection, we formally introduce the cooperation and competition strategies analysis problem and clarify mathematical symbols in this paper. For facilitating illustration, Table 1 lists some important mathematical notations used through out this paper.

In our setup, the company information and company relations are given as input in the heterogeneous form, which contains numeric and textual company information and complex connections between companies. Numeric input includes some parameters and statistics of a running company, such as establish date, market capitalization and number of employees. They can reflect a company's age, scale, and many other beneficial information. Textual input consists of company description text, usually describing the domain, main business, and some other comments on a company, which gives the company a high-level summarization.

Formally speaking, for each company c_i in company set C, we denote its numeric information as its numeric feature $\mu_i \in \mathbb{R}^{N_\mu}$, where N_μ is the dimension of numeric input, and represent its introduction and description text as a word sequence $\mathbf{w_i} = \{w_1, w_2, \ldots, w_T\}$ with each $w_t \in \mathcal{W}$, where $\mathcal{W}$ is the set of all words. Besides, the multiple inter-company relations are denoted as company relation graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where the set of nodes $\mathcal{V} = \{1, 2, \ldots, |C|\}$ represents all companies, and the edge set $\mathcal{E}$ contains all the relations, where $(u, r, v) \in \mathcal{E}$ represents that there is a relation between company u and v of relation type $r \in \mathcal{R}$.

With the symbols stated above, we define the enterprise cooperation and competition strategy analysis problem as follows:

Definition 3.1. **Company cooperation and competition strategy analysis.** Given a company c_i with heterogeneous information, namely numeric information μ_i, a textual sequence $\mathbf{w_i} = \{w_1, w_2, \ldots, w_T\}$ with sequence length T and a company relation graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ containing relations of companies in company set C, our goal is to generate a set of N_{coo} preference vectors $\mathcal{P}_i^{coo}$ representing the cooperation strategy, and a set of N_{com} preference vectors $\mathcal{P}_i^{com}$ representing the competition strategy. Through the encoding of all companies, the cooperation and competition preference vectors should be as close to the representations of its cooperation or competition candidates as possible.

3.2 Method Overview

Figure 1 shows the overall architecture of Sign-Oriented Preference Network. As shown in Figure 1(a), we first generate node embedding emb_i for each company c_i based on the heterogeneous input introduced before. We aggregate numerical input μ_i and text input $\mathbf{w_i}$ processed by a Gated Recurrent Unit (GRU) [4] as the initial node feature f_i. Then we employ Relational Graph Convolutional Network (RGCN) [22] to encode the heterogeneous company features and company relations, and generate deep representations emb_i for each company. Second, as shown in Figure 1(b), to generate more representative cooperation and competition preference vectors $\mathcal{P}_i^{coo}$ and $\mathcal{P}_i^{com}$, we incorporate the attention mechanism to distinguish different influence of different relations for each company, preserving cooperation and competition strategies respectively. Third, we train the model through a sign-oriented hybrid loss function that we propose, combining task specific loss with other constraints. To model the interaction and dependency between cooperation and competition relations, we design sign constraint $\mathcal{L}_{sign}$ according to the *balance theory* in signed graph theory [30]. And to ensure that our approach is both effective and interpretable, we also include a decoding loss $\mathcal{L}_{dec}$ in the hybrid loss function. In the following subsections, we will explain how each part of our approach works in detail.

3.3 Company Embedding

The first step is to generate node embeddings for each company based on the heterogeneous input. The aim of doing so is to aggregate as much information about companies, and acquire unified representations of companies for further cooperation and competition strategy analysis. Thus, it is necessary to deal with the heterogeneous input properly.

As described above, the heterogeneous input is consisted of company features and a company relation graph G. Specifically, the company features contain mainly two types of input information, the numeric information (e.g. company registered capital and numbers of employees) and the textual descriptions (e.g. company introductions and detailed scope of business operations). For each company c_i, its numeric information of N_μ dimensions are normalized as numeric feature μ_i, while its textual descriptions are in the form of a word sequence $\mathbf{w_i} = \{w_1, w_2, \ldots, w_T\}$ with T words. Here, we first leverage word2vec [16] to transform each word w_t in the sequence into a d_0-dimensional pre-trained word embedding vector. After the initialization, considering that the company descriptions have various lengths and can be long, we embed the

(a) Company embedding   (b) Cooperation and competition preference aggregating
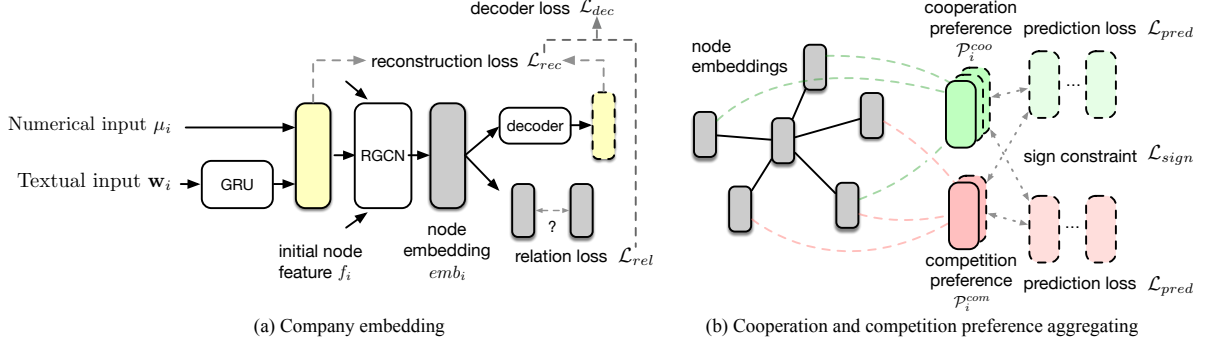
Figure 1: The framework of Sign-Oriented Preference Network (SOPN).

textual sequence by utilizing a multi-layer bi-directional variation of RNN, a bi-directional Gated Recurrent Unit (Bi-GRU), which preserves the most of contextual information of company description sentences from both forward and backward directions. Formally, given the description sequence $\mathbf{w_i} = \{w_1, w_2, \ldots, w_T\}$ of company c_i with sequence length T, we set the input of the first layer of Bi-GRU as $\overrightarrow{h}^{(0)} = \overleftarrow{h}^{(0)} = \{w_1, w_2, \ldots, w_T\}$. At time step t, forward hidden state $\overrightarrow{h}_t^{(l)}$ and backward hidden state $\overleftarrow{h}_t^{(l)}$ are updated at each layer l based on the previous hidden states $\overrightarrow{h}_{t-1}^{(l)}$ and $\overleftarrow{h}_{t-1}^{(l)}$ for both directions as:

$$\overrightarrow{h}_t^{(l)} = \text{GRU}(\overrightarrow{h}_{t-1}^{(l)}; \overrightarrow{\theta}_{GRU}), \quad (1)$$

$$\overleftarrow{h}_t^{(l)} = \text{GRU}(\overleftarrow{h}_{t-1}^{(l)}; \overleftarrow{\theta}_{GRU}), \quad (2)$$

where $\overrightarrow{\theta}_{GRU}$ and $\overleftarrow{\theta}_{GRU}$ denote forward and backward GRU parameters to be learned respectively.

To extract deep context in the company descriptions, the introduced GRU architecture has L_0 Bi-GRU layers. Thus, the deep linguistic information are able to be captured in the hidden states. As hidden state at each direction only contains one-side context, it is beneficial to combine them into one vector. Therefore, we obtain the company description sequence representation of company c_i at the last time step T by concatenating $\overrightarrow{h}_t^{(l)}$ and $\overleftarrow{h}_t^{(l)}$:

$$\tau_i = [\overrightarrow{h}_T^{(L_0)}; \overleftarrow{h}_T^{(L_0)}]. \quad (3)$$

With normalized numeric input μ_i and textual input τ_i derived from Bi-GRU, now we can get the input feature f_i of company c_i by concatenating μ_i and τ_i:

$$f_i = [\mu_i; \tau_i]. \quad (4)$$

So far, we have obtained the deep representation of company features, combining both numeric and textual information. However, each feature vector f_i only describes information about company c_i, but its relations with other companies are not considered yet. As discussed in Section 3.1, the various relations between all companies are represented as company relation graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{R})$. The unified representations for all companies should contain not only features of nodes, but also structural information in G. Thus, we employ a Relational Graph Convolutional Network (RGCN) to generate embedding for each node i in G due to its multiple relations. Formally, for each company c_i or node i, with its input features $\{f_i\}$

as the initial node features, the embedding of node i is calculated through messages passed from other nodes at layer l:

$$emb_i^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} emb_j^{(l)} + W_0^{(l)} emb_i^{(l)} \right), \quad (5)$$

where σ is the activation function (e.g. sigmoid), $\mathcal{N}_i^r$ denotes neighbors of company c_i under relation r, $c_{i,r}$ is normalization constant of node i and relation r that can either be learned or chosen in advance, and $W_r^{(l)}$ and $W_0^{(l)}$ are two weight matrices to be learned.

Since a RGCN layer only aggregates information from direct neighbors, we need more layers to propagate information of the whole network and model deep structural information. With L_1 RGCN layers employed, more structural information can be preserved in the embedding vectors. Thus, the heterogeneous input is encoded as node embedding $emb_i = emb_i^{(L_1)}$ at each node i.

3.4 Cooperation and Competition Preferences Aggregating

In this subsection, we aim at generating two sets of preference vectors for each company c_i based on node embeddings $\{emb_i\}$ acquired, representing its cooperation and competition strategies respectively. As a company's cooperation and competition strategies have great influence on how it connects with other companies, it is intuitive to aggregate information from its neighbors. Besides, various relations between companies are of different importance to the cooperation and competition strategies. Therefore, we leverage the attention mechanism to learn the contributions of different relations when generating the set of N_{coo} cooperation preference vectors $\mathcal{P}_i^{coo} = \{p_{i,k}^{coo}, k = 1, ..., N_{coo}\}$ and N_{com} competition preference vectors $\mathcal{P}_i^{com} = \{p_{i,k}^{coo}, k = 1, ..., N_{com}\}$ for each company c_i, with two hyper-parameters N_{coo} and N_{com}. Specifically, the attention score $\alpha_{r,k}^{coo}$ of relation r on the k-th cooperation preference vector is calculated over all company pairs with relation r as:

$$\alpha_{r,k}^{coo} = \frac{\exp(e_{r,k}^{coo})}{\sum\limits_{r \in \mathcal{R}} \exp(e_{r,k}^{coo})}, \quad (6)$$

$$e_{r,k}^{coo} = v_{\alpha,k}^{coo} \tanh\left(W_{\alpha,k}^{coo} \sum_{\substack{(i,r',j)\in\mathcal{E} \\ \text{if } r'=r}} [emb_i; emb_j] + b_{\alpha,k}^{coo} \right), \quad (7)$$

where $v_{\alpha,k}^{coo}, W_{\alpha,k}^{coo}, b_{\alpha,k}^{coo}$ are all the parameters to be learned during the training process.

Similarly, the attention score $\alpha_{r,k}^{com}$ of relation r on the k-th competition preference vector is computed as:

$$\alpha_{r,k}^{com} = \frac{\exp(e_{r,k}^{com})}{\sum_{r\in\mathcal{R}} \exp(e_{r,k}^{com})}, \quad (8)$$

$$e_{r,k}^{com} = v_{\alpha,k}^{com} \tanh\left(W_{\alpha,k}^{com} \sum_{\substack{(i,r',j)\in\mathcal{E} \\ \text{if } r'=r}} [emb_i; emb_j] + b_{\alpha,k}^{com} \right). \quad (9)$$

After the attention scores are defined, we can aggregate information for each company from its neighboring node embeddings, with some relations emphasized and others weakened. The k-th cooperation preference $p_{i,k}^{coo}$ and competition preference $p_{i,k}^{com}$ of company c_i are represented as:

$$p_{i,k}^{coo} = \sum_{j\in\mathcal{N}_i} \alpha_{r_{i,j},k}^{coo} emb_j, \quad (10)$$

$$p_{i,k}^{com} = \sum_{j\in\mathcal{N}_i} \alpha_{r_{i,j},k}^{com} emb_j, \quad (11)$$

where $\mathcal{N}_i$ denotes the set of neighboring companies that are connected with company c_i.

Till now, we have defined the vectorized cooperation preference set and competition preference set for each company. To summarize, we first encode the heterogeneous company features and relations into embeddings by utilizing a RGCN model. Specifically, the textual features of company features are extracted with a Bi-GRU network from word sequence input. Then, we leverage the attention mechanism to model the importances of different relations, as we aggregate feature information from neighboring nodes and generate the cooperation and competition preferences. Next, we will describe in detail how to train the model with the designing of a sign-oriented hybrid loss function.

3.5 Sign-oriented Hybrid Loss

Due to the complexity of the cooperation and competition analysis problem, we still need a well designed objective function to train the SOPN model. In this subsection, we focus on the construction of a novel sign-oriented hybrid loss function during the training process. First, a sign constraint based on the balance theory is introduced for modeling the dependency between cooperation and competition relations. Second, a decoding loss is constructed to extract meaningful information from the cooperation and competition preferences and ensure the interpretability of the strategy analysis. At last, the sign constraint and the decoding loss are combined with the main prediction task objective to train the model and learn all the parameters introduced in the previous subsection.
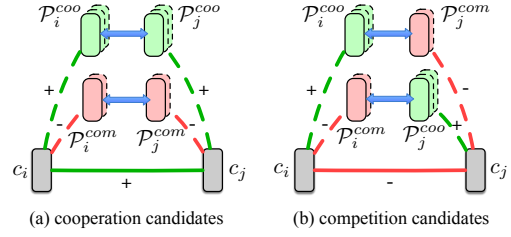

Figure 2: Possible balanced triangles between cooperation and competition candidates.

3.5.1 Sign Constraint on Cooperation and Competition Preferences.
As described before, the cooperation and competition relations are not independent. One common situation is that a company can somehow seek cooperation with the competitors of its competition candidates, as the idiom suggested, "the enemy of my enemy is my friend". This gives us the insight of modeling the dependency between cooperation and competition relations into a sign constraint, with cooperation and competition considered as positive and negative links respectively. According to the balance theory in signed network, each triangle of three companies contains either two negative links or zero negative links. That means, for each company pair c_i and c_j with cooperative (positive) relation, the links between company c_i and c_j and any other company are much likely to have the same signs, positive or negative. Intuitively, as illustrated in Figure 2(a), we expect the cooperation preference vectors $\mathcal{P}_i^{coo}$ of company c_i to be closer to its cooperation candidate company c_j's cooperation preference vectors $\mathcal{P}_j^{coo}$, than company c_j's competition preference vectors $\mathcal{P}_j^{com}$, and vice versa. Formally, the sign constraint between two companies with cooperative relation can be formulated as:

$$l_+(i,j) = \sum_k \max\{0, \min_{k_1} \|p_{i,k}^{coo} - p_{j,k_1}^{coo}\| - \min_{k_2} \|p_{i,k}^{coo} - p_{j,k_2}^{com}\|\}$$
$$+ \sum_k \max\{0, \min_{k_1} \|p_{i,k}^{com} - p_{j,k_1}^{com}\| - \min_{k_2} \|p_{i,k}^{com} - p_{j,k_2}^{coo}\|\}. \quad (12)$$

On the contrary, as illustrated in Figure 2(b), for two companies c_i and c_j with competitive relation, the cooperation preference vectors $\mathcal{P}_i^{coo}$ of company c_i should be closer to the competition preference vectors $\mathcal{P}_j^{com}$ of company c_j, than its cooperation preference vectors $\mathcal{P}_j^{coo}$. Formally, the sign constraint between two companies with competitive relation can be formulated as:

$$l_-(i,j) = \sum_k \max\{0, \min_{k_1} \|p_{i,k}^{coo} - p_{j,k_1}^{com}\| - \min_{k_2} \|p_{i,k}^{coo} - p_{j,k_2}^{coo}\|\}$$
$$+ \sum_k \max\{0, \min_{k_1} \|p_{i,k}^{com} - p_{j,k_1}^{coo}\| - \min_{k_2} \|p_{i,k}^{com} - p_{j,k_2}^{com}\|\}. \quad (13)$$

Thus, the total sign constraint between cooperation and competition relations is as follows:

$$\mathcal{L}_{sign} = \sum_{(i,j)\in coo} l_+(i,j) + \sum_{(i,j)\in com} l_-(i,j), \quad (14)$$

where *coo*, *com* denote the set of company pairs labeled with cooperative and competition relations, respectively.

3.5.2 Node Embeddings Decoding. For further analysis on enterprise cooperation and competition strategies, we expect to extract

meaningful information from the cooperation and competition preferences. That requires the cooperation preference vectors $\mathcal{P}_i^{coo}$ and competition preference vectors $\mathcal{P}_i^{com}$ along with the node embeddings emb_i to have the capability of reconstructing input features and structural information of each node i. In doing so, a decoder is employed to transfer the node embeddings into input features. The objective function of the decoder is formulated as:

$$\mathcal{L}_{rec} = \sum_{i \in \mathcal{V}} d(\text{dec}(emb_i), f_i), \quad (15)$$

where d can be chosen as L2 distance.

Besides, to prove the node embeddings' capability of capturing structural information, we construct two sets $\mathcal{P}$ and $\mathcal{N}$, both of which are consisted of sampled company pairs (i, j). The company pairs (i, j) in set P are sampled from links in company relation graph G, while company pairs (i, j) in N denote links that do not exists. The distance of pair (i, j) in P is expected to be closer than that in N. Formally, we can construct the objective function as:

$$\mathcal{L}_{rel} = \sum_{\substack{(i,j) \in P, \\ (u,v) \in N}} \max\left\{0, \|emb_i - emb_j\| - \|emb_u - emb_v\|\right\}. \quad (16)$$

Thus, the loss function of decoding node embeddings into input features and structural information is as follows:

$$\mathcal{L}_{dec} = \beta \mathcal{L}_{rec} + (1 - \beta) \mathcal{L}_{rel}, \quad (17)$$

where β controls the proportion of the two terms.

3.5.3 Prediction Task Objective. With the sign constraint and the decoding loss taken into consideration, now we can focus on the main prediction objective for training the model. The parameters defined in the SOPN model can be divided into several parts: $\overrightarrow{\theta}_{GRU}$ and $\overleftarrow{\theta}_{GRU}$ from the Bi-GRU, $W_{r,k}^{(l)}$ and $W_{r,k}^{(l)}$, $l = 0, \ldots, L_1$ from the RGCN network and $v_{\alpha,k}^{coo}, W_{\alpha,k}^{coo}, b_{\alpha,k}^{coo}, v_{\alpha,k}^{com}, W_{\alpha,k}^{com}, b_{\alpha,k}^{com}$ during the calculation of attention scores. In order to learn all these parameters, we design a cooperation and competition prediction task with some manually labeled cooperation and competition relations. For each company c_i, as its cooperation and competition strategies are represented as cooperation and competition preference vectors, the cooperation preference set $\mathcal{P}_i^{coo}$ and competition preference set $\mathcal{P}_i^{com}$ should preserve as much information as possible about its cooperation and competition companies, respectively. Therefore, we formulate the objective function of this task as:

$$\mathcal{L}_{pred} = \sum_{(i,j) \in coo} \min_k d(emb_j, p_{i,k}^{coo}) + \sum_{(i,j) \in com} \min_k d(emb_j, p_{i,k}^{com}), \quad (18)$$

where *coo* and *com* denote the labeled cooperation and competition company sets and $d(x, y)$ is a loss function that measures distance between vectors x and y, such as L2 loss.

Thus, the total loss of the training process contains three parts, namely the sign constraint loss $\mathcal{L}_{sign}$, the decoding loss $\mathcal{L}_{dec}$ from the reconstruction of input features and structural information, and the prediction loss $\mathcal{L}_{pred}$ from the cooperation and competition prediction task:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{sign} + \lambda_2 \mathcal{L}_{dec} + \lambda_3 \mathcal{L}_{pred}, \quad (19)$$

where λ_1, λ_2 and λ_3 control the proportions of different objectives.

Table 2: Statistics of the enterprise dataset.

Name	Value
#Companies	3,595
#Labeled companies	300
#Numeric features	6
#Relation types	5
#Edges	3,188
#Labeled cooperation pairs	714
#Labeled competition pairs	1,552
Avg. words in description	106.36
Avg. edges per company	10.63
Avg. cooperation relations	2.38
Avg. competition relations	5.17

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Dataset Description. We conduct experiments on the enterprise dataset collected and hand labeled by experts. For the input information of our framework, we collect the enterprise dataset from a public data service website *Tianyancha* in China[1], which contains a vast repository of Chinese enterprise information, including various company information and multiple relations about companies. Company information collected includes numerical features of each company (e.g. company registered capital and numbers of employees) and textual descriptions (company introductions and detailed scope of business operations). Besides, we also collect $\mathcal{R}$ different relations among these companies, including shareholding relation, litigation relation and so on. For cooperation and competition relations used in the training step, we manually extract those relations for a selection of companies from each company's public prospectus carefully labeled by analysts and experts. Some of the statistics are listed in Table 2 to demonstrate the dataset.

4.1.2 Baselines. In order to demonstrate the accuracy of learned preference vectors, we compare with some state-of-the-art algorithms on cooperation and competition relation prediction. The baseline methods can be divided into two categories. The first is to propose cooperation and competition candidates based on item recommendation. The second is to directly predict the possible relations of a pair of companies based on their company embeddings. The details of these baselines are illustrated as follows:

- **BPR**[21]: It is an implicit feedback based recommendation method. In our setup, we can see observed cooperation and competition samples as positive feedback separately, and recommend companies for each relation independently.
- **LibFM**[20]: It is a recommendation method best at context-aware prediction. The setup under our problem is similar to BPR, but with company information added as context.
- **GRU**[4]: Different from the above methods, the following methods are embedding based. This method only leverages company information and conducts company description embedding using GRU, a common variation of RNN.

---

[1]https://www.tianyancha.com

Table 3: Performance comparisons on cooperation and competition candidate prediction.

Method	Cooperation						Competition					
	Precision	Recall	F1	AUC	NDCG	MAP	Precision	Recall	F1	AUC	NDCG	MAP
BPR	0.378	0.443	0.408	0.688	0.788	0.709	0.508	0.446	0.475	0.523	0.777	0.653
LibFM	0.396	0.454	0.423	0.696	0.786	0.707	0.514	0.460	0.485	0.528	0.776	0.653
GRU	0.501	**0.534**	0.517	0.701	0.792	0.752	0.583	0.362	0.447	0.561	0.779	0.670
Node2Vec	0.247	0.250	0.248	0.590	0.681	0.700	0.508	0.446	0.475	0.523	0.777	0.653
LINE	0.395	0.409	0.402	0.626	0.703	0.630	0.521	0.489	0.505	0.537	0.771	0.645
GCN	0.557	0.443	0.493	0.695	0.736	0.721	0.545	**0.495**	0.518	0.556	0.781	0.664
GraphSage	0.644	0.431	0.517	0.695	0.783	0.733	0.574	0.430	0.492	0.567	0.780	0.674
RGCN	0.622	0.431	0.510	0.703	0.806	0.742	0.596	0.446	0.510	0.583	0.778	0.666
RGCN-T	0.578	0.500	0.536	0.724	0.815	0.762	0.619	0.481	0.541	0.603	0.796	**0.692**
SOPN	**0.765**	0.453	**0.569**	**0.745**	**0.857**	**0.767**	**0.630**	0.493	**0.553**	**0.613**	**0.815**	0.682

Table 4: Ablation experiments demonstrating model performance with different input types and losses.

Method	Cooperation						Competition					
	Precision	Recall	F1	AUC	NDCG	MAP	Precision	Recall	F1	AUC	NDCG	MAP
SOPN-R	0.555	0.340	0.422	0.658	0.809	0.728	0.572	0.415	0.481	0.564	0.788	0.650
SOPN-N	0.636	0.477	0.545	0.726	0.850	0.755	0.629	0.483	0.546	0.609	0.811	0.677
SOPN-T	0.661	0.488	0.562	0.733	0.857	0.767	0.625	0.479	0.543	0.607	0.808	0.675
only $\mathcal{L}_{pred}$	0.651	0.488	0.558	0.732	0.853	0.747	0.611	0.465	0.528	0.595	0.807	0.675
with $\mathcal{L}_{sign}$	0.676	**0.500**	**0.575**	0.739	0.842	**0.774**	**0.630**	0.491	0.552	**0.614**	0.810	0.678
with $\mathcal{L}_{dec}$	0.653	0.386	0.485	0.683	0.849	0.755	0.597	0.423	0.495	0.579	0.796	0.657
SOPN (full)	**0.765**	0.453	0.569	**0.745**	**0.857**	0.767	**0.630**	**0.493**	**0.553**	0.613	**0.815**	**0.682**

- **Node2Vec**[10]: This method incorporates unsupervised network embedding based on biased truncated random walk.
- **LINE**[24]: It learns the network embedding by preserving the first-order proximity or second-order proximity of the network structure separately.
- **GCN**[13]: This method is based on an efficient variant of convolution neural network operating directly on graphs. We also optimize node embedding in semi-supervised learning as proposed in their paper.
- **GraphSage**[11]: It is an up-to-date network embedding method for large scale graph embedding.
- **RGCN**[22]: This represents the state-of-the-art method for relational network embedding.
- **RGCN-T**: This is an variation of RGCN, where we also incorporate text input in the network. The input features have the same preprocessing as in our method.

4.1.3 Evaluation Protocols. In order to evaluate the accuracy of the generated cooperation and competition preference vectors, we split the hand-labeled relation into 60% training, 10% validation, and 30% testing data sets. It is also worth mentioning that, to mimic the situation where cooperation and competition companies are all unknown in reality, we filtered all the edges connect across training and testing set to avoid information leakage in the testing stage.

We compare the results with two categories of evaluation metrics. The first is classification correctness for labeled pairs of companies. The metrics include Precision, Recall, F1-score and AUC. The second

is ranking metrics, where for each company, we compare output scores of the candidates with the real labels on each relation. The metrics include NDCG and MAP.

4.1.4 Experimental Setup. We implement our model using Py-Torch[2] and DGL[3]. The parameters are all initialized using Xavier [8] initialization. We set the number of GRU layers as 1 and number of RGCN layers as 2, and the embedding dimensions are set as 50. In the process of model training, we use the Adam optimizer [12] for parameter optimization. We set learning rate as 0.01 and mini-batch size as 32. The parameters of baselines are set up similarly as our method and are all tuned to be optimal to ensure fair comparisons.

4.2 Experimental Results

4.2.1 Overall Performance. To demonstrate the effectiveness of our generated strategy vectors, we first compare our SOPN model with all the baseline methods on cooperation and competition candidate prediction problem. All the results are shown in Table 3.

From the results, we can get several observations. First, the performance of SOPN surpasses the baseline methods on most of the evaluation metrics. This clearly proves that our generated strategies have the ability to accurately represent all the cooperation and competition candidates. Second, our SOPN obtains much higher precision than baselines, which shows that our approach is useful

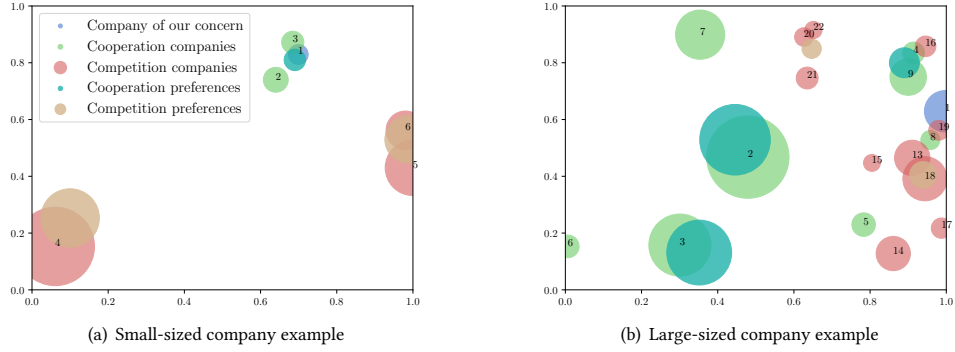(a) Small-sized company example   (b) Large-sized company example

Figure 3: Visualization of two companies' cooperation and competition preferences and candidates. Marker scales are in proportion to real market capitalization of each company.

for a more thorough mining of potential candidates, and thus more suitable for real-world scenarios. Third, from comparison between GRU, GCN and RGCN, we can easily see that applying different forms of input has a different impact on the final performance, which proves the necessity of heterogeneous input embedding. Last but not least, we can see that cooperation relies more on text input, and relation information plays more important role in competition, as GRU works better on cooperation but graph embedding methods have better performance on competition prediction.

4.2.2 Ablation Study. Moreover, we conduct some ablation experiments to further show how each part of our method affects final results. We experiment with different types of inputs and different loss configuration, and show the results in Table 4. The method SOPN-R is SOPN with randomly initialized feature as node feature input, instead of numeric and textual inputs. SOPN-N is SOPN with numerical inputs but without text inputs, and SOPN-T is the opposite. We also omit different parts of loss for comparison.

From the results, we can draw the following conclusions: First, it is clear that the more information the model is fed with, the better performance it has. Second, sign loss can boost the performance alongside the prediction loss, which proves that there exists certain correlation between cooperation and competition, and our model is capable of capturing such correlation. Third, decoding loss $\mathcal{L}_{dec}$ slightly reduces the performance in many cases. This is probably because the model strives to achieve two different goals at the same time. But with the slight decrease of performance, we are able to obtain more interpretability, which is a huge need for further cooperation and competition strategy analysis.

4.2.3 Parameter Sensitivity. We investigate the sensitivity of our model parameter in this section. Specifically, we mainly evaluate how the numbers of preference vectors N_{coo} and N_{com} affect the performance. The results are shown in Figure 4.

In the first few steps, the precision keeps going up. This is because more preference vectors mean more capability of capturing different aspects of cooperation or competition strategies. But this would do harm to the recall, as shown in Figure 4, since more preference vectors can cause more companies to be mistaken as candidates. In fact, it may also lead to serious over-fitting problem
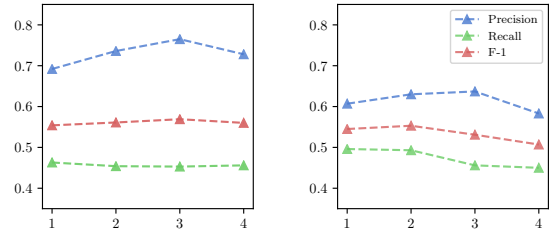

Figure 4: Comparison on different numbers of cooperation and competition preference vectors generated.

once the numbers is larger than 3, and even precision decreasing. Next thing we can notice is that, as the numbers of preference vectors increase, the F-1 scores gradually increase, and then decrease. To ensure best performance, we choose the optimal value $N_{coo} = 3$, $N_{com} = 2$ as the number of preference vectors.

4.3 Cooperation and Competition Analysis

To further demonstrate how to utilize the generated preference vectors and analyze enterprise cooperation and competition, we will continue our discussion with several case studies.

As shown in Figure 3, we first visualize the company of our concern (blue), along with its cooperation (green) and competition (red) preferences and all the related companies in two-dimensional space. All the company embeddings and preference vectors learned by SOPN are projected into two dimensional with the widely used visualization tool t-SNE [15]. Then we set the pointer size in proportion to the real market capitalization of each company. As each preference vector can be seen as a cluster center of preferred companies' embeddings, we use the decoder defined in Section 3.5.3 to also restore preferred company scale for each preference vector. With size and distance representing company's actual scale and its relevance with other company in the company network respectively, we can intuitively discover how each company and preference relate to each other.

From Figure 3, we can observe two different cooperation and competition strategies by two companies. The left company (labeled

as 1 in the left) in Figure 3(a) has a relatively small scale. From the visualization, we can infer that in order to fight against several large companies (red markers, representing companies in competition), this particular company unites with several small companies that are already closely related (green markers, representing companies in cooperation). In contrast, the right company (labeled as 1 in the right) in Figure 3(b) has conducted a different strategy. With more fierce competition in the industry (implied by the red markers nearby), it has to cooperate with larger companies that are less related (as the green markers imply). Above these observations, it is also beneficial for discovering potential partners and/or competitors for an enterprise, which shows a great possibility of our method to help realize business value.

5 CONCLUSION

In this paper, we proposed a Sign-oriented Preference Network (SOPN) for the analysis of enterprise cooperation and competition strategies. Specifically, we first exploited a Relational Graph Convolutional Network (RGCN) to generate a deep representation for companies based on heterogeneous company information. Also, with the embedding vectors derived from RGCN, we designed the attention mechanism to distinguish the effects of multiple company relations on company strategies by quantitatively measuring their corresponding cooperation and competition preferences. Then, to model the dependency of the cooperation and competition relations, we designed a sign-oriented constraint based on the signed graph theory. Moreover, we trained the model through a hybrid loss function, combining task specific loss with the sign constraint and graph decoding loss, to ensure both effective modeling and better interpretability. Finally, extensive experiments on real-world enterprise dataset showed the effectiveness of our approach. Meanwhile, we provided a case study to reveal some interesting patterns as well as their business implications.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

[2] Ignacio Bretos and Carmen Marcuello. 2017. Revisiting globalization challenges and opportunities in the development of cooperatives. *Annals of Public and Cooperative Economics* 88, 1 (2017), 47–73.

[3] Yingmei Chen, Zhongyu Wei, and Xuanjing Huang. 2018. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1655–1658.

[4] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*.

[5] Hepu Deng, Chung-Hsing Yeh, and Robert J Willis. 2000. Inter-company comparison using modified TOPSIS with objective weights. *Computers & Operations Research* 27, 10 (2000), 963–973.

[6] Tyler Derr, Yao Ma, and Jiliang Tang. 2018. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 929–934.

[7] Dongfang Du, Hao Wang, Tong Xu, Yanan Lu, Qi Liu, and Enhong Chen. 2017. Solving link-oriented tasks in signed network via an embedding approach. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 75–80.

[8] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 249–256.

[9] Devi R Gnyawali and Byung-Jin Robert Park. 2011. Co-opetition between giants: Collaboration with competitors for technological innovation. *Research policy* 40, 5 (2011), 650–663.

[10] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[11] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.

[12] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[13] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR '17)*.

[14] Chaang-Yung Kung and Kun-Li Wen. 2007. Applying grey relational analysis and grey decision-making to evaluate the relationship between company attributes and its financial performance-a case study of venture capital enterprises in Taiwan. *Decision support systems* 43, 3 (2007), 842–852.

[15] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13)*. Red Hook, NY, USA, 3111–3119.

[17] Semih Önüt, Selin Soner Kara, and Elif Işik. 2009. Long term supplier selection using a combined fuzzy MCDM approach: A case study for a telecommunication company. *Expert systems with applications* 36, 2 (2009), 3887–3895.

[18] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

[19] Cristina Quintana-Garcia and Carlos A Benavides-Velasco. 2004. Cooperation, competition, and innovative capability: a panel data of European dedicated biotechnology firms. *Technovation* 24, 12 (2004), 927–938.

[20] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 1–22.

[21] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[22] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 593–607.

[23] Hubert Schmitz. 1999. Global competition and local cooperation: success and failure in the Sinos Valley, Brazil. *World development* 27, 9 (1999), 1627–1650.

[24] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.

[25] David J Teece. 2010. Business models, business strategy and innovation. *Long range planning* 43, 2-3 (2010), 172–194.

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations* (2018).

[27] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 1225–1234.

[28] Hao Wang, Tong Xu, Qi Liu, Defu Lian, Enhong Chen, Dongfang Du, Han Wu, and Wen Su. 2019. MCNE: An End-to-End Framework for Learning Multiple Conditional Network Representations of Social Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1064–1072.

[29] Shuhan Yuan, Xintao Wu, and Yang Xiang. 2017. SNE: signed network embedding. In *PAKDD*. Springer, 183–195.

[30] Thomas Zaslavsky. 1982. Signed graphs. *Discrete Applied Mathematics* (1982), 47–74.

[31] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 793–803.

[32] Le Zhang, Tong Xu, Hengshu Zhu, Chuan Qin, Qingxin Meng, Hui Xiong, and Enhong Chen. 2020. Large-Scale Talent Flow Embedding for Company Competitive Analysis. In *Proceedings of The Web Conference 2020*. 2354–2364.

[33] Le Zhang, Hengshu Zhu, Tong Xu, Chen Zhu, Chuan Qin, Hui Xiong, and Enhong Chen. 2019. Large-Scale Talent Flow Forecast with Dynamic Latent Factor Model. In *The World Wide Web Conference*. 2312–2322.