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

Le Dai¹, Yu Yin¹, Chuan Qin¹,², Tong Xu∗¹, Xiangnan He², Enhong Chen¹,², Hui Xiong¹∗
¹School of Computer Science and Technology, University of Science and Technology of China, ²School of Data Science, University of Science and Technology of China, ³Baidu Talent Intelligence Center, Baidu Inc., ⁴Rutgers University

dl123.yxonic@mail.ustc.edu.cn, {tongxu.chenhe}@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 require 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 superior 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.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>The company set</td>
</tr>
<tr>
<td>$f_i$</td>
<td>Concatenated features of numeric features $\mu_i$ and textual features $t_i$ of company $c_i$</td>
</tr>
<tr>
<td>$G$</td>
<td>The company network with node set $V$, labeled edge set $E$ and relation set $R$</td>
</tr>
<tr>
<td>$N_i$</td>
<td>The neighboring nodes of node $i$</td>
</tr>
<tr>
<td>$emb_i$</td>
<td>The embedded vector of company $c_i$</td>
</tr>
<tr>
<td>$P_i^{coo}$, $P_i^{com}$</td>
<td>The sets of cooperation and competition preference vectors of company $c_i$</td>
</tr>
<tr>
<td>$N_{coo}, N_{com}$</td>
<td>Hyperparameters of the number of cooperation and competition preference vectors</td>
</tr>
</tbody>
</table>

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 throughout 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 establishment 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 $N_\mu$ is the dimension of numeric input, and represent its introduction and description text as a word sequence $w_i = \{w_{i1}, w_{i2}, \ldots, w_{iT}\}$ with each $w_{it} \in \mathcal{W}$, where $\mathcal{W}$ is the set of all words. Besides, the multiple inter-company relations are denoted as company relation graph $G = (V, E, R)$, where the set of nodes $V = \{1, 2, \ldots, |C|\}$ represents all companies, and the edge set $E$ contains all the relations, where $(u, r, v) \in E$ represents that there is a relation between company $u$ and $v$ of relation type $r \in R$.

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 $\mu_i$, a textual sequence $w_i = \{w_{i1}, w_{i2}, \ldots, w_{iT}\}$ with sequence length $T$ and a company relation graph $G = (V, E, R)$ containing relations of companies in company set $C$, our goal is to generate a set of $N_{coo}$ preference vectors $P_i^{coo}$ representing the cooperation strategy, and a set of $N_{com}$ preference vectors $P_i^{com}$ 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 $\mu_i$ and text input $w_i$ 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 $P_i^{coo}$ and $P_i^{com}$, 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 $L_{sign}$ 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 $L_{dec}$ 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 $N_\mu$ dimensions are normalized as numeric feature $\mu_i$, while its textual descriptions are in the form of a word sequence $w_i = \{w_{i1}, w_{i2}, \ldots, w_{iT}\}$ with $T$ words. Here, we first leverage word2vec [16] to transform each word $w_i$ in the sequence into a $d_\mu$-dimensional pre-trained word embedding vector. After the initialization, considering that the company descriptions have various lengths and can be long, we embed the
Numerical input $\mu_i$, Textual input $w_i$ → GRU → RGCN → Decoder

(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 $w_t = \{w_1, w_2, \ldots, w_T\}$ of company $c_i$ with sequence length $T$, we set the input of the first layer of Bi-GRU as $h^{(0)}_t = h^{(0)}_l = \{w_1, w_2, \ldots, w_T\}$. At time step $t$, forward hidden state $\overrightarrow{h}^{(l)}_t$ and backward hidden state $\overleftarrow{h}^{(l)}_t$ are updated at each layer $l$ based on the previous hidden states $\overrightarrow{h}^{(l)}_{t-1}$ and $\overleftarrow{h}^{(l)}_{t-1}$ for both directions as:

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

where $\theta_{\text{GRU}}$ and $\theta_{\text{GRU}}$ denote forward and backward GRU parameters to be learned respectively.

To extract deep context in the company descriptions, we introduce GRU architecture has $L_{\theta}$ 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 $\overrightarrow{h}^{(L)}_T$ and $\overleftarrow{h}^{(L)}_T$:

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

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

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

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 $G = (V, E, R)$. 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 $l$:

$emb_i^{(l+1)} = \sigma \left( \sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{|N_i^r|} W_r^{(l)} \cdot emb_j^{(l)} + W_0^{(l)} \cdot emb_i^{(l)} \right), \quad (5)$

where $\sigma$ is the activation function (e.g. sigmoid), $N_i^r$ 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 $L_i$ RGCN layers employed, more structural information can be preserved in the embedding vectors. Thus, the heterogeneous input is encoded as node embedding $emb_i = emb_i^{(L_i)}$ 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 $N_{\text{coo}}$ cooperation preference vectors $\mathcal{P}_{i}^{\text{coo}} = \{p_{i,k}^{\text{coo}} \mid k = 1, \ldots, N_{\text{coo}}\}$ and $N_{\text{com}}$ competition preference vectors $\mathcal{P}_{i}^{\text{com}} = \{p_{i,k}^{\text{com}} \mid k = 1, \ldots, N_{\text{com}}\}$ for each company $c_i$, with two hyper-parameters $N_{\text{coo}}$ and $N_{\text{com}}$. Specifically, the attention score $\alpha_{r,k}^{\text{coo}}$ of relation $r$ on the $k$-th cooperation preference vector is calculated over all company pairs with relation $r$ as:

$\alpha_{r,k}^{\text{coo}} = \frac{\exp(c_{r,k}^{\text{coo}})}{\sum_{r \in R} \exp(c_{r,k}^{\text{coo}})}, \quad (6)$
where \( w_{a,k} \), \( b_{a,k} \), \( b_{r,k} \) are all the parameters to be learned during the training process.

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

\[
\alpha_{r,k}^{\text{com}} = \frac{\exp(e_{r,k}^{\text{com}})}{\sum_{r \in \mathcal{R}} \exp(e_{r,k}^{\text{com}})}
\]

In equation (8), \( e_{r,k}^{\text{com}} = \tanh \left( \sum_{(i',j') \in \mathcal{E}} [\text{emb}_{i'}; \text{emb}_{j'}] + b_{r,k}^{\text{com}} \right) \).

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 \( p_{i,k}^{\text{coop}} \) and competition preference \( p_{i,k}^{\text{com}} \) of company \( c_i \) are represented as:

\[
p_{i,k}^{\text{coop}} = \sum_{j \in \mathcal{N}_i} \alpha_{r,j,k}^{\text{coop}} \text{emb}_{j},
\]

\[
p_{i,k}^{\text{com}} = \sum_{j \in \mathcal{N}_i} \alpha_{r,j,k}^{\text{com}} \text{emb}_{j},
\]

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.

\[
e_{r,k}^{\text{coop}} = v_{r,k}^{\text{coop}} \tanh \left( \sum_{(i',j') \in \mathcal{E}} [\text{emb}_{i'}; \text{emb}_{j'}] + b_{r,k}^{\text{coop}} \right),
\]

where \( v_{a,k}^{\text{coop}} \), \( W_{a,k}^{\text{coop}} \), \( b_{a,k}^{\text{coop}} \), \( b_{r,k}^{\text{coop}} \) are all the parameters to be learned during the training process.

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

\[
\alpha_{r,k}^{\text{com}} = \frac{\exp(e_{r,k}^{\text{com}})}{\sum_{r \in \mathcal{R}} \exp(e_{r,k}^{\text{com}})}.
\]

In equation (8), \( e_{r,k}^{\text{com}} = \tanh \left( \sum_{(i',j') \in \mathcal{E}} [\text{emb}_{i'}; \text{emb}_{j'}] + b_{r,k}^{\text{com}} \right) \).

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 \( p_{i,k}^{\text{coop}} \) and competition preference \( p_{i,k}^{\text{com}} \) of company \( c_i \) are represented as:

\[
p_{i,k}^{\text{coop}} = \sum_{j \in \mathcal{N}_i} \alpha_{r,j,k}^{\text{coop}} \text{emb}_{j},
\]

\[
p_{i,k}^{\text{com}} = \sum_{j \in \mathcal{N}_i} \alpha_{r,j,k}^{\text{com}} \text{emb}_{j},
\]

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.

\[
L_{\text{sign}} = \sum_{(i,j) \in \mathcal{E}_{\text{coop}}} L_{+,i,j} + \sum_{(i,j) \in \mathcal{E}_{\text{com}}} L_{-,i,j},
\]

(14)

where \( \mathcal{E}_{\text{coop}} \) and \( \mathcal{E}_{\text{com}} \) 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 \( \mathbf{P}^{\text{coo}}_i \) and competition preference vectors \( \mathbf{P}^{\text{com}}_i \) along with the node embeddings \( \mathbf{emb}_i \) 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:

\[
L_{\text{dec}} = \sum_{i \in \mathcal{V}} d(\text{dec}(\mathbf{emb}_i), f_i),
\]

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 \( \mathcal{P} \) are sampled from links in company relation graph \( G \), while company pairs \((i, j)\) in \( \mathcal{N} \) denote links that do not exists. The distance of pair \((i, j)\) in \( \mathcal{P} \) is expected to be closer than that in \( \mathcal{N} \). Formally, we can construct the objective function as:

\[
L_{\text{rel}} = \sum_{(i, j) \in \mathcal{P}, (u, v) \in \mathcal{N}} \max \{ 0, \|\mathbf{emb}_i - \mathbf{emb}_j\| - \|\mathbf{emb}_u - \mathbf{emb}_v\| \}. \tag{16}
\]

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

\[
L_{\text{dec}} = \beta L_{\text{rec}} + (1 - \beta) L_{\text{rel}}, \tag{17}
\]

where \( \beta \) 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: \( \bar{\theta}_\text{GRU} \) and \( \bar{\theta}_\text{GRU} \) from the Bi-GRU, \( W^{(l)}_{r, a, k} \) and \( W^{(l)}_{r, a, k} \), \( l = 0, \ldots, L_1 \), from the RGCN network and \( \mathbf{W}^{\text{coo}}_{a, k}, \mathbf{W}^{\text{coo}}_{a, k}, \mathbf{W}^{\text{coo}}_{a, k}, \mathbf{W}^{\text{coo}}_{a, k}, \mathbf{W}^{\text{com}}_{a, k}, \mathbf{W}^{\text{com}}_{a, k} \) 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 \( \mathbf{P}^{\text{coo}}_i \) and competition preference set \( \mathbf{P}^{\text{com}}_i \) should preserve as much information as possible about its cooperation and competition companies, respectively. Therefore, we formulate the objective function of this task as:

\[
L_{\text{pred}} = \sum_{(i, j) \in \text{coo}} \min d(\mathbf{emb}_j, \mathbf{P}^{\text{coo}}_{i, k}) + \sum_{(i, j) \in \text{com}} \min d(\mathbf{emb}_j, \mathbf{P}^{\text{com}}_{i, k}), \tag{18}
\]

where \( \text{coo and com} \) 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 \( L_{\text{sign}} \), the decoding loss \( L_{\text{dec}} \) from the reconstruction of input features and structural information, and the prediction loss \( L_{\text{pred}} \) from the cooperation and competition prediction task:

\[
L = \lambda_1 L_{\text{sign}} + \lambda_2 L_{\text{dec}} + \lambda_3 L_{\text{pred}}, \tag{19}
\]

where \( \lambda_1, \lambda_2 \) and \( \lambda_3 \) control the proportions of different objectives.

### 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\(^\text{3}\), 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 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**\(^\text{[21]}\): 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**\(^\text{[20]}\): 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**\(^\text{[4]}\): 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.

---

\(^{\text{3}}\)https://www.tianyancha.com
Table 3: Performance comparisons on cooperation and competition candidate prediction.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cooperation</th>
<th>Competition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>BPR</td>
<td>0.378</td>
<td>0.445</td>
</tr>
<tr>
<td>LibFM</td>
<td>0.396</td>
<td>0.454</td>
</tr>
<tr>
<td>GRU</td>
<td>0.501</td>
<td><strong>0.534</strong></td>
</tr>
<tr>
<td>Node2Vec</td>
<td>0.247</td>
<td>0.250</td>
</tr>
<tr>
<td>LINE</td>
<td>0.395</td>
<td>0.409</td>
</tr>
<tr>
<td>GCN</td>
<td>0.557</td>
<td>0.443</td>
</tr>
<tr>
<td>GraphSage</td>
<td>0.644</td>
<td>0.431</td>
</tr>
<tr>
<td>RGCN</td>
<td>0.622</td>
<td>0.431</td>
</tr>
<tr>
<td>RGCN-T</td>
<td>0.578</td>
<td>0.500</td>
</tr>
<tr>
<td>SOPN</td>
<td><strong>0.765</strong></td>
<td>0.453</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Cooperation</th>
<th>Competition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>SOPN-R</td>
<td>0.555</td>
<td>0.340</td>
</tr>
<tr>
<td>SOPN-N</td>
<td>0.636</td>
<td>0.477</td>
</tr>
<tr>
<td>SOPN-T</td>
<td>0.661</td>
<td>0.488</td>
</tr>
<tr>
<td>only $L_{pred}$ &amp; $L_{sign}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>with $L_{pred}$ &amp; $L_{sign}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOPN (full)</td>
<td><strong>0.765</strong></td>
<td>0.453</td>
</tr>
</tbody>
</table>

- **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 a 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 PyTorch\(^2\) 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 for ranking candidate companies.

[^2]: https://pytorch.org
[^3]: https://github.com/dmlc/dgl
for a more thorough mining of potential candidates, and thus more suitable for real-world scenarios. Third, from comparison between GRU, GCN and RGSCN, 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 $L_{dec}$ 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 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 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 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 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 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.

ACKNOWLEDGMENTS

This research was partially supported by grants from the National Key Research and Development Program of China (Grant No. 2018YF B1402600), and the National Natural Science Foundation of China (Grant No. 9174301, 61703386, U1605251).

REFERENCES


