Time-aware Graph Relational Attention Network for Stock Recommendation

Xiaoting Ying
Central South University

Cong Xu
Central South University

Jianliang Gao
Central South University

Jianxin Wang
Central South University

Zhao Li
Alibaba Group

ABSTRACT

Recommending stock with the highest return ratio is always a challenging problem in the field of financial technology. In this paper, we propose a time-aware graph relational attention network (TRAN) for stock recommendation based on return ratio ranking. In TRAN, time-aware relational attention mechanism is the key unit to capture time-varying correlation strength between stocks by the interaction of historical sequences and stock description documents. With the dynamic strength, the nodes of the stock relation graph aggregate the features of neighbor stock nodes by graph convolution operation. For a given group of stocks, our model can output the ranking results of stocks according to their return ratios. The experimental results on several real-world datasets demonstrate the effectiveness of our TRAN for stock recommendation.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Social and professional topics** → **Economic impact**.

KEYWORDS

Stock recommendation; graph neural networks; relational-attention; stock relation graph

ACM Reference Format:
Xiaoting Ying, Cong Xu, Jianliang Gao, Jianxin Wang, and Zhao Li. 2020. Time-aware Graph Relational Attention Network for Stock Recommendation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340531.3412160

1 INTRODUCTION

Stock market plays an important role in the economic operations of modern society, which motivates further exploration of stock prediction techniques to seek higher revenue [9]. But analysing stock market movements and price behaviours is extremely challenging because of the highly volatile and non-stationary nature of the market.

In order to predict stock price, some researchers utilize time series information such as historical stock price [11], other researchers dig into textual features such as financial news [6], social media [10], etc. Beside these basic series information and textual information, connection among corporations is utilized for stock prediction recently [2]. According to the efficient market hypothesis, the change of the stock price of a target corporation would be affected by corporations that are related [4]. Therefore, it is natural to utilize the stock relations as an injection of explicit knowledge when making stock prediction. However, there are two major challenges for utilizing the connection among stocks (corporations):

- How to quantify the strength of the connection among corporations. Corporations are connected with each other broadly via various relationships such as the classification of sector-industry. Even if a group of stocks falls into the same category, there is a difference in the strength of their correlation with each other.
- How to quantify the time-varying strength of the connection among corporations. Although correlation exists among corporations, the strength changes over time. The correlation strength between different corporations can remain essentially stable, decreased, and increased in the next time step. Therefore, it is necessary to design a method to describe the more detailed relation and capture the time-aware correlation between stocks.

In order to tackle these two challenges, we propose a **T**ime-aware Graph **R**elational **A**ttention **N**etwork (TRAN) for stock recommendation. Firstly, we construct the stock relation graph based on the classification of sector-industry, and extract the historical feature and stock attribute from historical sequence and stock description document, respectively. Then, a time-aware graph relational attention network is designed to capture time-aware stock relation strength by the dynamic interaction of historical feature and stock attribute. The relation strength is treated as attention weight on stock relation graph and graph convolution operation is utilized to aggregate historical features of related stock nodes. Finally, the output of graph convolution and the stock historical features are utilized to predict ranking of all stocks. We empirically show that the proposed model outperforms existing stock recommendation methods on two real market datasets.

2 PRELIMINARIES

For all S stocks, we get a set of historical sequence data and collect description documents of corporations that issue stocks. A set of stock historical sequence data at day t is represented by $\mathcal{X}_t = \{X_t^1, \cdots, X_t^S\} \in \mathbb{R}^{S \times T \times K}$, where T is the length of time series and K is the dimension of features such as *open*, *high*, *low*, *close*.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412160

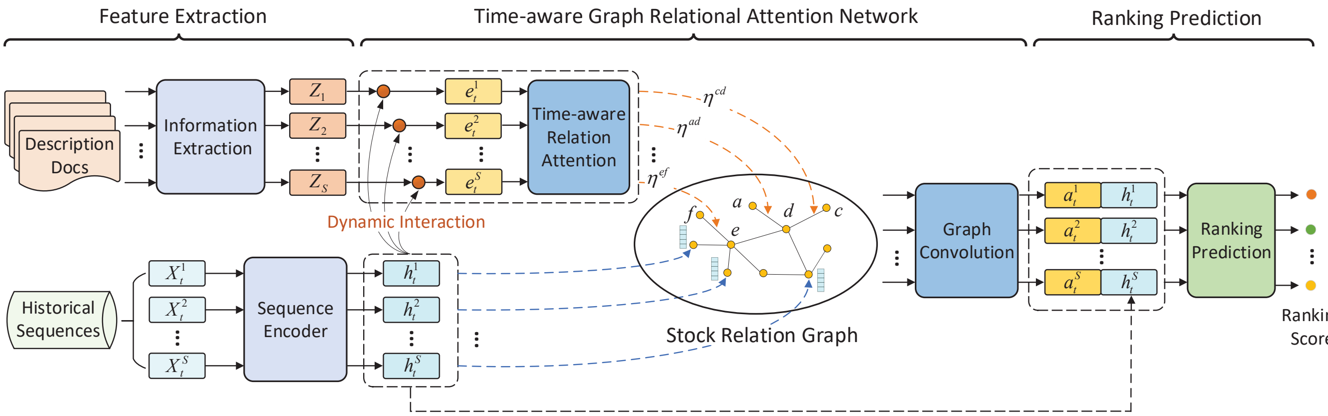

Figure 1: The overall framework of stock recommendation model.

Description documents of corporations that issue stocks are represented by $D = \{d_1, \cdots, d_S\}$. For each stock s, it has own description document d_s and historical sequence data $X_t^s = \{x_{t-T+1}^s, \cdots, x_t^s\}$ at day t. In particular, every stock has its attribute of sector-industry classification. For example, MSFT (Microsoft Inc.) and GOOGL (Alphabet Inc.) belong to the same sector-industry classification (Technology: Computer Software). Subsequently, we define a stock relation graph based on their sector-industry classifications as the explicit knowledge for stock recommendation.

Definition 1. **Stock Relation Graph.** *An undirected graph $G = \{V, E\}$, where V is a set of $|V| = S$ nodes representing stocks; E is a set of edges, indicating the relation between stocks. To determine the relation between stock i and j, their sector-industry classifications are converted into multi-hot binary vector C_i and C_j. If $C_i \cdot C_j \geq 1$, there is an edge connecting stock i and j. Each node s generates a feature vector at each time step t from the historical sequence X_t^s, and each edge generates the time-aware relation strength at each time step t.*

Definition 2. **Stock Recommendation.** *We define the return ratio as the ground-truth ranking score $y_{t+1}^s = (r_{t+1}^s - r_t^s)/r_t^s$ where r_t^s is the closing price at day t. Given a stock relation graph G, historical sequences $\mathcal{X}_t = \{X_t^1, \cdots, X_t^S\}$ and description documents $D = \{d_1, \cdots, d_S\}$, our problem is to learn a function f which is able to get the $t+1$ day ranking list $\hat{\mathbf{y}}_{t+1} = \{\hat{y}_{t+1}^1, \cdots, \hat{y}_{t+1}^S\}$ sorted by their ranking scores, and then top-1 stock is recommended to the investor. The mapping relation is represented as follows:*

$$[\mathcal{X}_t, D, G] \xrightarrow{f} \hat{\mathbf{y}}_{t+1}. \tag{1}$$

3 METHODOLOGY

In this section, we introduce the detailed design of our model and Figure 1 illustrates its overall framework, including *Feature Extraction, Time-aware Graph Relational Attention Network* and *Ranking Prediction.*

3.1 Feature Extraction

This module obtains the features from the historical sequences and the description documents of stocks (or the target corporations).

Sequence Encoder. The Long Short Term Memory (LSTM) networks have been widely applied to process sequential data [2]. Therefore, we also adopt the LSTM networks to encode the historical features of stocks over time. Given the historical sequence of stock s, $X_t^s = \{x_{t-T+1}^s, \cdots, x_t^s\}$, we input it into the LSTM networks. It is formulated as follows:

$$h_t^s = LSTM(h_{t-1}^s, x_t^s) \tag{2}$$

where $h_t^s \in \mathbb{R}^U$ is the last hidden state and U is the number of hidden units in LSTM.

Topic-based Information Extraction. The item description documents have been proved to valuable in the recommendation models [7]. For stock recommendation task, stock description documents contain many factors (e.g., business, capital) that are helpful in describing more detailed relations. Inspired by Latent Dirichlet Allocation model, we extract the topic distribution for the description document d_s of each stock s, $Z_s = \{z_1^s, \cdots, z_L^s\} \in \mathbb{R}^L$, where L is the topic numbers. Therefore, the description documents of all stocks D are encoded as the features of topic distributions $Z = \{Z_1, \cdots, Z_S\} \in \mathbb{R}^{L \times S}$ to describe the attributes of stocks.

3.2 Time-aware Graph Relational Attention Network

This module contains two core units: Time-aware Relational Attention (TRA) unit and Graph Convolution (GC) unit. TRA unit captures time-aware stock relation strength, and GC unit updates the stock representation based on stock relation graph.

TRA unit. Firstly, we interact dynamically the stock attribute Z_s and historical sequence encoding h_t^s to account for the temporal property and detailed relational feature of stock. For stock s at day t, we define the *Dynamic Interaction* function as:

$$e_t^s = \psi(W_{f1}^T [h_t^s; W_{f2} Z_s] + b_f), \tag{3}$$

where ψ is an activation function; W_{f1}^T denotes the transpose of W_{f1}; $W_{f1} \in \mathbb{R}^{2U \times U}$ and $W_{f2} \in \mathbb{R}^{U \times L}$ are the parameter matrixs of a fully connected layer; and $b_f \in \mathbb{R}^U$ is the learnable bias. We obtain all S stocks latent embedding $E_t = \{e_t^1, \cdots, e_t^S\} \in \mathbb{R}^{S \times U}$.

Then, we design a time-aware relational attention unit to get time-aware relation strength from the dynamic interaction, and treat the relation strength as edge weight η in the stock relation graph. Therefore, the time-aware relation strength between stock s and stock j at each day t is computed by:

$$\eta_t^{sj} = \frac{\exp(\alpha_t^s(j))}{\sum_{k \in \mathcal{N}_s} \exp(\alpha_t^s(k))}, \tag{4}$$

$$\alpha_t^s(j) = u_a^T \phi(W_a e_t^j + b_a), \tag{5}$$

where $\alpha_t^s(j)$ indicates the importance of stock node j's features to stock node s; $j \in \mathcal{N}_s$ and $\mathcal{N}_s$ is neighborhood of stock node s; ϕ is an activation function; $u_a \in \mathbb{R}^{M'}$, $W_a \in \mathbb{R}^{M' \times U}$, and b_a are parameters to be learned.

GC unit. General GCN model[5] cannot capture the time-aware relation strength, the adjacency matrix is fixed. We employ graph convolution unit to update the node representation in the stock relation graph with time-aware edge weight. For stock s, we will aggregate the features of its all neighbor stock nodes in the graph to get the relational feature representation $a_t^s \in \mathbb{R}^U$ at each day t. The graph convolution operation is formulated as follows:

$$a_t^s = \sum_{j \in \mathcal{N}_s} \frac{\eta_t^{sj}}{degree(j)} h_t^j, \tag{6}$$

where $degree(j)$ is the degree of stock node j.

3.3 Ranking Prediction

This module calculates the ranking score for each stock and output the ranking list of all stocks.

Score Calculation. The output a_t^s of TRAN are the updated representation of stock node. We concatenate a_t^s with the historical embedding h_t^s as the final representation m_t^s:

$$m_t^s = [a_t^s; h_t^s], \tag{7}$$

where $m_t^s \in \mathbb{R}^{2U}$. Then, we deploy a full connection layer as the predictive function to calculate the ranking score $\hat{y}_{t+1}^s$:

$$\hat{y}_{t+1}^s = LeakyReLU(W_p^T m_t^s + b_p), \tag{8}$$

where $W_p \in \mathbb{R}^{2U}$ and b_p are the learnable parameters. For all S stocks, we can get the ranking list $\hat{\mathbf{y}}_{t+1} = \{\hat{y}_{t+1}^1, \cdots, \hat{y}_{t+1}^S\} \in \mathbb{R}^S$.

Loss Function. In order to meet the needs of stock recommendation, we consider the relative comparison between stocks to select the stock with maximum return ratio. Therefore, we combine both pointwise regression loss and pairwise ranking-aware loss:

$$\begin{aligned} L(\hat{\mathbf{y}}_{t+1}, \mathbf{y}_{t+1}) &= \|\hat{\mathbf{y}}_{t+1} - \mathbf{y}_{t+1}\|^2 \\ &+ \alpha \sum_i \sum_j max\{0, -(\hat{y}_{t+1}^i - \hat{y}_{t+1}^j)(y_{t+1}^i - y_{t+1}^j)\}, \end{aligned} \tag{9}$$

where $\hat{\mathbf{y}}_{t+1}$ and $\mathbf{y}_{t+1}$ denote predicted and the ground-truth ranking list at day $t+1$, respectively; $\hat{y}_{t+1}^i$ and y_{t+1}^i denote the predicted and ground-truth ranking score of stock i at day $t+1$, respectively; and α is a hyperparameter to balance the two loss terms.

4 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of TRAN on two real market datasets. The experimental results are compared to the state-of-the-art models.

Table 1: Statistics of the datasets.

Datasets		NASDAQ	NYSE
Stock Historical Sequence	#Training Days	756	756
	#Validation Days	252	252
	#Testing Days	237	237
Stock Description Document	#Descriptions	5130	5955
	#Words	319297	427932
Stock Relation Graph	#Nodes	1026	1737
	#Edges	52586	98065

4.1 Datasets

Stock Historical Sequence. We use the stocks from NASDAQ and NYSE markets that have historical sequences between 01/02/2013 and 12/08/2017, including 1,026 and 1,737 stocks, respectively [4]. In detailed, the entire dataset is divided into the training set, validation set and testing set, and their lengths are shown in Table 1.

Stock Description Document. We collect the description documents of the stocks from Yahoo Finance[1]. As shown in Table 1, 5,130 and 5,955 description documents are collected in NASDAQ and NYSE, respectively.

Stock Relation Graph. We get the 112 and 130 kinds of sector-industry classifications in the NASDAQ and NYSE markets [4], which are used to construct stock relation graph. Table 1 shows the summary statistics for stock relation graph.

4.2 Experimental Setup

Market Simulation. Similar to [3], we adopt a market simulation strategy to evaluate the performance through calculating the cumulative return ratio on testing period. At first, we assume the investor spends the same amount of money on every trading day and the stock market is liquid enough to satisfy investor's trading requests. Then, the investor buys the top-1 stock at the closing price at day t based on the predicted ranking list and sells it at trading day $t+1$.

Evaluation Metrics. We evaluate the prediction performance with three metrics: Mean Square Error (MSE), Mean Reciprocal Rank (MRR) [1], and Investment Return Ratio (IRR). MSE evaluates the volatility between the ground-truth and predicted ranking scores over all stocks on every testing day. MRR evaluates the predicted rank of the top-1 return ratio stock in the ground-truth over the testing days. IRR is our main metric, which is the sum of return ratio of every testing day based on market simulation strategy. Note that better performance is smaller value of MSE (≥ 0) and larger value of MRR ([0,1]) and IRR.

Parameter Settings. Our proposed model is implemented with Tensorflow and optimized by Adam with a learning rate of 0.001. Depend on perplexity results for Latent Dirichlet Allocation (LDA), the topic numbers of NASDAQ and NYSE are 50 and 60, respectively. For the length of sequential input T and the number of hidden units U in LSTM, we select them via grid-search within the ranges of [2, 4, 8, 16] and [16, 32, 64, 128]. And we tune α in loss function within [0.1, 1, 10].

[1] https://finance.yahoo.com

Table 2: Experimental results of different models.

Methods	NASDAQ			NYSE		
	MSE	MRR	IRR	MSE	MRR	IRR
Rank_LSTM	3.79e-4±1.11e-6	**4.17e-2±7.50e-3**	0.68±0.60	2.28e-4±1.16e-6	3.79e-2±8.82e-3	0.56±0.68
GCN	3.80e-4±2.66e-6	3.66e-2±6.69e-3	0.39±0.49	2.27e-4±1.12e-6	4.07e-2±7.54e-3	0.82±0.71
RSR_E	3.82e-4±2.96e-6	3.16e-2±3.45e-3	0.20±0.22	2.29e-4±2.77e-6	4.28e-2±6.18e-3	1.00±0.58
RSR_I	3.80e-4±7.90e-7	3.17e-2±5.09e-3	0.23±0.27	2.26e-4±5.30e-7	4.51e-2±2.41e-3	1.06±0.27
TRAN	**3.79e-4±3.90e-7**	3.81e-2±4.37e-3	**0.92±0.25**	**2.26e-4±2.30e-7**	**4.91e-2±4.82e-3**	**1.38±0.35**

4.3 Baselines

Our method is compared with a number of prediction baselines:

- **Rank_LSTM**: Rank_LSTM treats stock prediction as a return ratio ranking task based on the LSTM network [4].
- **GCN**: We replace TRAN of our proposed model with a common GCN layer [5].
- **RSR_E**: RSR_E is a stock ranking prediction model based on stock relations, which adds relational embedding layer of explicit modeling to Rank_LSTM [4].
- **RSR_I**: RSR_I is a variant of RSR_E through replacing explicit modeling with implicit modeling [4].

4.4 Experimental Results

According to the experimental results of the proposed TRAN and the other baselines, we analyze the overall performance and the impacts from different aspects.

Overall Performance. As the most important metric in stock recommendation, the IRR of our model achieves the highest performance in both NASDAQ and NYSE markets, comparing to other state-of-the-art models. For example, the IRR values are 0.92 and 1.38 for our method, respectively. This result verifies the advantage and practicability of the proposed stock recommendation model. Additional, the stability is also verified by the MSE metric. In both NASDAQ and NYSE markets, the MSE values of TRAN are the smallest compared to other baselines.

Impact of Stock Relation Graph. In the proposed TRAN, stock relation graph is used to formulate the relations among multiple stocks. Rank_LSTM method make prediction without considering the relations. Therefore, comparison between Rank_LSTM and TRAN shows that the effectiveness of considering stock relations according to the results of MRR and IRR. Although our method fails to beat Rank_LSTM regarding MRR in the NASDAQ market, the reason could be attributed to that the relations between stable stocks can be encoded more efficiently, and NASDAQ is considered as a much more volatile market than NYSE [8].

Impact of Time-Aware Graph Relational Attention Network. GCN method takes stock relations into consideration, but ignoring the temporal properties of stock relations. Although RSR_I and RSR_E simply utilize the relation obtained from the historical sequence, they still perform better than GCN. It indicates that the more accurate the description of the dynamic time-aware relations between stocks, the better the performance of stock recommendation task. And our model outperforms GCN, RSR_I and RSR_E in terms of all metrics on two datasets. These results prove the effectiveness of our proposed TRAN, that is, TRAN can obtain the property of detailed time-aware relation from the stock attribute and the dynamic trend of historical sequence and capture time-aware relation strength to aggregate sequence feature.

5 CONCLUSIONS

In this paper, we have proposed TRAN for stock recommendation based on return ratio ranking. Our model utilizes both series information and textual information to characterize the stock relation. In the proposed TRAN, the time-aware relational attention mechanism is able to capture time-aware relation strength among stocks. Meanwhile, graph convolution is applied on the stock relation graph to aggregate the features of the related stocks with time-aware relation strength. Our TRAN model can rank the stocks according to their expected return ratios. Experimental results show that our model can achieve better performance for stock recommendation.

ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China under Grant No.: 61873288.

REFERENCES

[1] Chi Chen, Li Zhao, Jiang Bian, Chunxiao Xing, and Tie-Yan Liu. 2019. Investment Behaviors Can Tell What Inside: Exploring Stock Intrinsic Properties for Stock Trend Prediction. In *KDD*. 2376–2384.

[2] Yingmei Chen, Zhongyu Wei, and Xuanjing Huang. 2018. Incorporating Corporation Relationship via Graph Convolutional Neural Networks for Stock Price Prediction. In *CIKM*. 1655–1658.

[3] Matthew Dixon, Diego Klabjan, and Jin Hoon Bang. 2017. Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance* 6, 3-4 (2017), 67–77.

[4] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems* 37, 2 (2019), 27:1–27:30.

[5] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*. 1–14.

[6] Qikai Liu, Xiang Cheng, Sen Su, and Shuguang Zhu. 2018. Hierarchical complementary attention network for predicting stock price movements with news. In *CIKM*. 1603–1606.

[7] Julian J. McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring Networks of Substitutable and Complementary Products. In *KDD*. 785–794.

[8] G William Schwert. 2002. Stock volatility in the new millennium: how wacky is Nasdaq? *Journal of Monetary Economics* 49, 1 (2002), 3–26.

[9] Huizhe Wu, Wei Zhang, Weiwei Shen, and Jun Wang. 2018. Hybrid deep sequential modeling for social text-driven stock prediction. In *CIKM*. 1627–1630.

[10] Yumo Xu and Shay B Cohen. 2018. Stock movement prediction from tweets and historical prices. In *ACL*. 1970–1979.

[11] Liheng Zhang, Charu C. Aggarwal, and Guo-Jun Qi. 2017. Stock Price Prediction via Discovering Multi-Frequency Trading Patterns. In *KDD*. 2141–2149.