TGSLN : Time-aware Graph Structure Learning Network for Multi-variates Stock Sector Ranking Recommendation

Quan Wan
Central South University
Changsha, China

Shuo Yin
Central South University
Changsha, China

Xiangyue Liu
Central South University
Changsha, China

Jianliang Gao*
Central South University
Changsha, China

Yuhui Zhong
Central South University
Changsha, China

ABSTRACT

In the field of financial prediction, most studies focus on individual stocks or stock indices. Stock sectors are collections of stocks with similar characteristics and the indices of sectors have more stable trends and predictability compared to individual stocks. Additionally, stock sectors are subsets of stock indices, which implies that investment portfolios based on stock sectors have a greater potential to achieve excess returns. In this paper, we propose a new method, **T**ime-aware **G**raph **S**tructure **L**earning **N**etwork (TGSLN), to address the problem of stock sector ranking recommendation. In this model, we use an indicator called Relative Price Strength (RPS) to describe the ranking change trend of the sectors. To construct the inherent connection between sectors, we construct a multi-variable time series that consists of multi-scale RPS sequences and effective indicators filtered through the factor selector. We also build a stock sector relation graph based on authoritative stock sector classifications. Specially, we design a time-aware graph structure learner, which can mine the sector relations from time series, and enhance the initial graph through graph fusion. Our model outperforms state-of-the-art baselines in both A-share and NASDAQ markets.

CCS CONCEPTS

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

KEYWORDS

Stock sector recommendation; graph neural network; time-aware; multi-variates

ACM Reference Format:
Quan Wan, Shuo Yin, Xiangyue Liu, Jianliang Gao*, and Yuhui Zhong. 2023. TGSLN : Time-aware Graph Structure Learning Network for Multi-variates Stock Sector Ranking Recommendation. In *35th International Conference on Scientific and Statistical Database Management (SSDBM 2023), July 10–12, 2023, Los Angeles, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3603719.3603741

SSDBM 2023, July 10–12, 2023, Los Angeles, CA, USA

ACM ISBN 979-8-4007-0746-9/23/07…$15.00
https://doi.org/10.1145/3603719.3603741

1 INTRODUCTION

In recent years, the stock market has developed rapidly and stock prediction has become an important tool for investment and portfolio construction. Most previous research[4] on stock prediction is based on individual stocks. However, individual stock prices have high volatility[8], making stock prediction tasks extremely difficult. Instead, stock sector indices have more stable trends, which greatly increase the predictability of the financial prediction. Additionally, since stock sectors have fixed concepts, such as the computer sector, the relationships between stock sectors are more long-term and accurate compared to individual stocks. These facts demonstrate the feasibility of stock sector prediction tasks. Meanwhile, many stock price predictions utilize graph neural networks to model stock nodes[2, 12]. However, their approaches are based only on predefined stock relation graphs obtained from prior knowledge and implicit graphs mined from time features. After the training process is finished, these methods do not have the ability to adapt and improve graph construction with the time series input. And the implicit graphs constructed by these methods are usually dense graphs[12], which may result in noise information contained in the redundant small values. To address these issues, we propose a novel method **T**ime-aware **G**raph **S**tructure **L**earning **N**etwork (TGSLN), which focuses on stock sector ranking recommendation. The temporal input of TGSLN is composed of multi-scale RPS time series and effective stock sector factors filtered by the factor selector, while the initial graph consists of a learnable embedding matrix and a predefined stock sector relation graph based on sector classifications. TGLSN consists of four modules:time-aware graph structure learner, time series constructor, recurrent time series encoder and ranking predictor. To ensure effective graph structure learning, we design a time-aware graph structure learner that incorporates a graph regularization layer based on Gumbel-Softmax and a graph structure enhancement fusion gate. Then, the time series encoder captures the time features from the time series and outputs the hidden states using the recurrent time series encoder. Finally, the ranking predictor outputs the RPS ranking score for stock sector recommendations. To validate the effectiveness of our model, we conduct experiments on real datasets from the A-share and NASDAQ markets, and our method achieves the best performance. The contributions of this paper are as follows:

- We find that the trend of the stock sector is significantly less volatile than that of individual stocks, and exhibits a stable and continuous characteristic. Furthermore, the stock sectors have long-term relationships based on stock sector classifications.

- We propose a time-aware graph structure learning network, which extracts time features from multi-scale RPS time series and effective factors, and mines the stock sector relations with the time-aware graph structure learner.
- Combining the two metrics, we perform the stock sector ranking prediction task on two real datasets, and our method outperforms the baseline methods. The results confirm the effectiveness of our approach in stock sector ranking recommendation.

2 PRELIMINATION

To bridge the gap between stock sector prediction and investment returns, we build a portfolio based on the predicted ranking of stock sectors. Given a candidate set $S = \{s_1, ..., s_N\}$ of N stock sectors on trading day t, the input of stock sectors $X^t = \{x^{t-T}, ..., x^t\} \in \mathbb{R}^{\mathbb{T}\times\mathbb{N}\times\mathbb{F}}$ consists of the multi-scale RPS time series and the selected stock sector factors, $T < t$, where T is the historical time steps of input and F is the dimension of features. And we use two types of graphs to describe the relation between stock sectors. The explicit graph G comprises the multiple independent fully-connected stock sector relation graphs based on stock sector classifications. The implicit graph θ, can describe the correlation between stock sector embeddings. Our model TGSLN(G, θ, X^t) fuse these two graphs using graph structure learning modules and output the ranking scores of stock sectors Y^{t+1} on day $t+1$, $Y^{t+1} = \{y_1^{t+1}, ..., y_N^{t+1}\}$.

Definition 1. ***RPS_m^t.*** *RPS is defined as the ranking change of a single stock sector index among all stock sectors over a period of m days. Let $P_m^t \in \mathbb{R}$ be the rate of change of the stock sector index over m days on day t, $t > m$ and $t, m \in \mathbb{Z}^+$. Next, we sort the P_m^t of all n stock sectors in descending order and obtain the ranking $rank_m^t$ of this single sector, $n \in \mathbb{Z}^+$. And finally RPS_m^t can be defined as:*

$$RPS_m^t = 1 - \frac{1}{rank_m^t} \tag{1}$$

where $RPS_m^t \in \mathbb{Z}$ and $RPS_m^t \in [0, 100)$. Actually, RPS_m^t represents the percentage of stock sectors whose index growth rate over the recent m days is surpassed by a specific stock sector's index growth rate on day t. Stock sectors with higher yield rankings have higher RPS values. Therefore, RPS is very suitable as a proxy variable for ranking scores.

Definition 2. ***IC.*** *In this task, the factor selector is based on the IC test. The IC value of a tested stock sector factor is determined by calculating the absolute value of the correlation coefficient between the factor and the RPS_1 ranking score we aim to predict. Generally, factors with high IC values are more useful for prediction tasks.*

3 METHODOLOGY

3.1 Time-aware Graph Structure Learner

Graph Structure Initialization. Feng et al.[4] discover that modeling the correlation among stocks within the same sector can improve the model's ability to predict stock prices. This demonstrates the effectiveness of the hierarchical relationships. Inspired by this, we construct the predefined stock sector relation graph G based on authoritative sector classifications. Let $E \in \mathbb{R}^{N\times d}$ be the initialized node embedding matrix of the implicit graph, d be the embedding dimension, θ be the probability matrix of embeddings calculated by cosine similarity. The learner takes both the explicit graph G and the implicit graph θ as inputs.

Graph Regularization. Regularization methods can help prevent overfitting in models. Inspired by Jang et al.[6], we use the Gumbel-Softmax technique from reinforcement learning to address the regularization problem. Let δ be the Softmax function, and τ be the temperature variable of Gumbel-Softmax. The definition of the implicit stock sector relation matrix $M^{(t)}$ is as follows:

$$\begin{aligned} M_{ij}^t &= \sigma\left(\left(\log\left(\theta_{ij}/(1-\theta_{ij})\right) + \left(g_{ij}^1 - g_{ij}^2\right)\right)/\tau\right) \\ &\text{s.t. } g_{ij}^1, g_{ij}^2 \sim Gumbel(0,1) \end{aligned} \tag{2}$$

where θ_{ij} represents the probability of retaining the edge in the implicit graph. As a reparameterization method, Gumbel-Softmax can solve the problem of overly dense matrices and the representation ability of generated sparse graph is more effective. Unlike traditional regularization methods, Gumbel-Softmax can retain useful edges with maximum probability while discarding useless ones.

Time-aware Graph Feature Extractor. To extract temporal relations from the time series and use them to enhance the graph structure, we design two time-aware feature extractors to extract the features of two graphs. The process can be expressed as:

$$\begin{aligned} \tilde{G}^t &= \left(I + D^{-\frac{1}{2}} G D^{-\frac{1}{2}}\right) X^t W^{(g)} + {W_b}^{(g)} \\ \tilde{M}^t &= \left(I + D^{-\frac{1}{2}} M^t D^{-\frac{1}{2}}\right) X^t W^{(m)} + {W_b}^{(m)} \end{aligned} \tag{3}$$

where $W^{(g)}$, $W^{(m)}$, ${W_b}^{(g)}$, ${W_b}^{(m)}$ are parameters matrices, and X^t is the time series input.

Graph Fusion Gate. Graph fusion is commonly used to enhance graph structure. After performing feature extraction on both explicit and implicit graphs, we use custom self-attention layers to calculate weights. They assign higher weights to relation edges with high internal similarity, further enhancing the graph structure.

$$A^t = f_g\left(\tilde{G}^t; \theta_g\right) + f_m\left(\tilde{M}^t; \theta_m\right) \tag{4}$$

where θ_g and θ_m are the parameters of the self-attention layers.

3.2 Time Series Constructor & Encoder

Time Series Constructor. We construct a multi-scale RPS time series to help the model learn temporal features from both long-term and short-term perspectives. The RPS scales are 1 day, 3 days, 5 days, 10 days, 15 days, and 20 days. Meanwhile, other stock sector factors must undergo IC effectiveness test as explained in definition 2 in section 2. These two inputs make up the time series input.

Recurrent Time Series Encoder. To gain further insights into the stock sector relations and temporal interdependencies in the time series, we input the multi-variates time series into a recurrent time series encoder that includes a gate and gated recurrent unit (GRU) cells. The gate utilizes the structure of the mixed stock sector relation graph to selectively aggregate time series information and assists in updating the hidden state. The process of the network can be expressed as follows:

$$\begin{aligned} z_t &= \sigma\left(A^t\left(\left[X^t, h_{t-1}\right]\right)\right) \\ r_t &= \sigma\left(A^t\left(\left[X^t, h_{t-1}\right]\right)\right) \\ \tilde{h}_t &= \tanh\left(A^t\left(\left[r_t \odot h_{t-1}, X^t\right]\right)\right) \\ h_t &= z_t \odot \tilde{h}_t + (1 - z_t) \odot h_{t-1} \end{aligned} \tag{5}$$

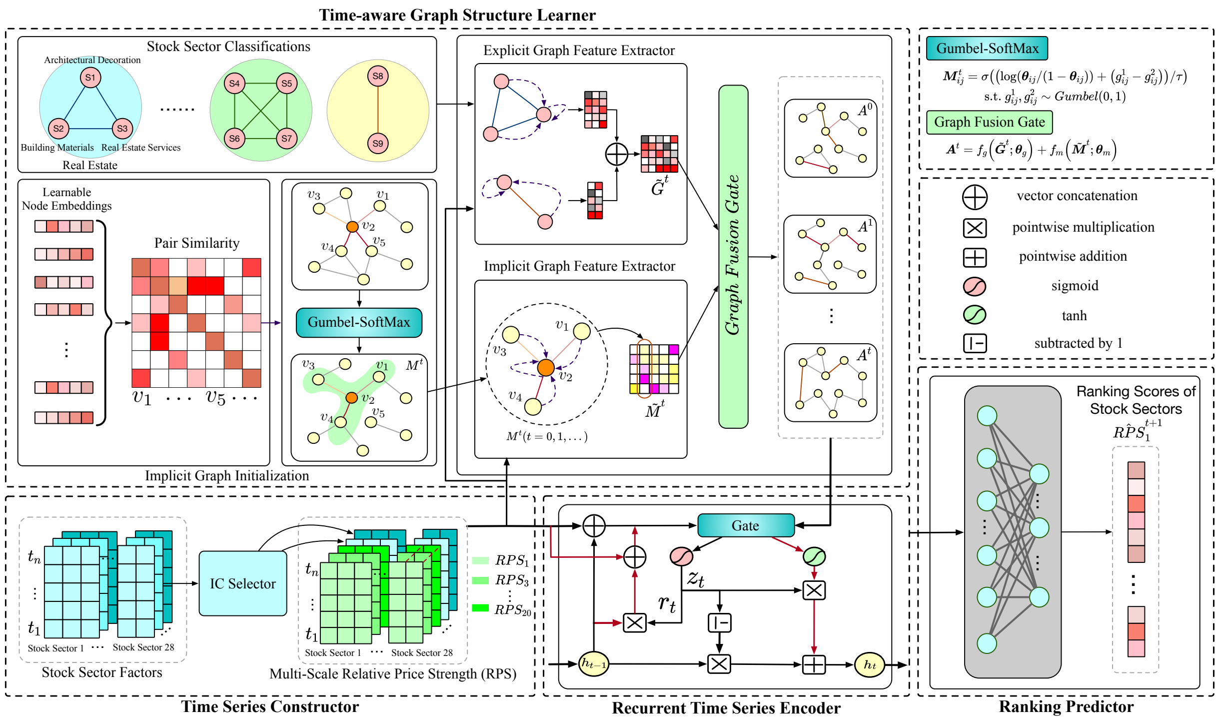

Figure 1: The overall framework of the proposed TGSLN.

In this process, z_t is the update gate and r_t is the reset gate, and we input multi-variates time series X^t to update the hidden state h_t.

3.3 Ranking Predictor

Ranking Recommendation. To predict the ranking score $\hat{RPS}_1^{t+1}$, we input the hidden state h_{t+1} into the MLP which outputs the ranking scores for day $t+1$. Finally, stock sectors with higher ranking scores are given priority in the recommendations.

Rank-loss. For the ranking task, we adopt rank-loss[10] as the loss function, which analyzes pairwise ranking errors and jointly calculates pointwise regression and pairwise ranking losses with weighted coefficients, minimizing the difference between predicted y_i^t and actual values $\hat{y}_i^t$ while maintaining the relative order of stock sectors, and embeddings are updated correspondingly.

$$\mathcal{L} = \sum_{i=1}^{N} \left\|\hat{y}_i^t - y_i^t\right\|^2 + \alpha \sum_{i=1}^{N} \sum_{j=1}^{N} ReLU\left(-\left(\hat{y}_i^t - \hat{y}_j^t\right)\left(y_i^t - y_j^t\right)\right) \quad (6)$$

4 EXPERIMENTS

4.1 Experimental Setup

Datasets. We select stock sector data from the A-share market from 02/01/2005 to 12/10/2021 and stock sector data from the NASDAQ market from 03/02/2016 to 03/01/2023 as our datasets, which are obtained from JoinQuant[1], akshare[2], and Morningstar[3]. Table 1 shows the detailed statistical data.

Table 1: Statistics of datasets.

Datasets		A-share	NASDAQ
Multi-variates Time Series	#Training Days	2868	1220
	#Validation Days	819	174
	#Testing Days	410	348
Stock Sector Classifications	#Nodes	28	123
	#Classifications	8	11
	#Edges	46	807

Market Simulation. We assume that the market can meet all the trading needs. Investors buy the funds corresponding to the predicted $Top5$ stock sectors, with an equal amount of money invested in each fund. The funds will be sold on the next trading day.

Evaluation Metrics. We use two evaluation metrics including Mean Reciprocal Rank (MRR), and Investment Return Rate (IRR) to evaluate the performance. **MRR** equals the average reciprocal rank of the $Top5$ stock sector based on daily ranking scores during the testing period and **IRR** is an evaluation metric that represents the cumulative return rate of testing days based on the market simulation strategy mentioned in market simulation.

Implementation Details. TGSLN is implemented by PyTorch and optimized by Adam, with a learning rate of 0.001 and a batch size of 16. We use the average results of 10 repeating experiments, and each time has 100 epochs. We utilize an early stop strategy and 7-day raw inputs to predict the next stock sector ranking scores.

[1]https://www.joinquant.com/
[2]https://github.com/akfamily/akshare
[3]https://www.morningstar.com/

Table 2: Performance comparison with baseline methods.

Datasets	Metrics	TGSLN	RNN	LSTM	CNN-GRU	Transformer	RANK_LSTM	MTGNN	AGCRN	HIST
A-share	MRR	**0.144**	0.110	0.111	0.119	0.118	0.121	0.127	0.126	0.129
	IRR	**0.670**	0.346	0.366	0.365	0.401	0.430	0.548	0.473	0.558
NASDAQ	MRR	**0.074**	0.036	0.038	0.045	0.045	0.050	0.054	0.055	0.058
	IRR	**0.180**	-0.111	-0.089	-0.043	-0.024	0.063	0.102	0.075	0.100

Table 3: Ablation study.

Methods	Metrics	A-share	NASDAQ
w/o Explicit-G	MRR	0.132	0.060
	IRR	0.491	0.116
w/o Implicit-G	MRR	0.137	0.064
	IRR	0.510	0.128
w/o G-Regularization	MRR	0.142	0.071
	IRR	0.624	0.167
TGSLN	MRR	**0.144**	**0.074**
	IRR	**0.670**	**0.180**

Baselines. To demonstrate the effectiveness of our method, we conduct experiments in comparison with other models including RNN[7], LSTM[5], CNN-GRU[3], Transformer[9], RANK_LSTM[4], MTGNN[11], AGCRN[1] and HIST[12].

4.2 Experimental Results & Analysis

Performance Comparison with Baseline Models. The experimental results are presented in Table 2, and the best results are highlighted in bold. TGSLN outperforms other state-of-the-art baseline models on the A-share and NASDAQ markets. We can summarize the results from the following perspectives: 1) The graph-based methods outperform other methods indicating that the stock sector relation graph can provide more effective information for prediction tasks. 2) Compared to univariate regression methods such as LSTM, the models based on the multiple variates achieve better results on the stock sector ranking recommendation. This demonstrates the effectiveness of multi-scale RPS sequences and selected stock sector factors. 3) Compared to the result of MTGNN and HIST, TGSLN performs best, proving that time-aware implicit graphs and graph fusion enhancement can optimize the structure of the stock sector relation graph. 4) Our method TGSLN outperforms the method AGCRN, which highlights the importance of information extracted from the predefined stock sector classification graph.

Ablation study. To validate the interpretability of our proposed model, we design ablation experiments on A-share and NASDAQ datasets. As shown in Table 3, TGSLN without the explicit graph (w/o Explicit-G) performs the worst, highlighting the importance of long-term hierarchical classifications on individual stock sector nodes. TGSLN without the implicit graph (w/o Implicit-G) is also outperformed by the complete TGSLN model, emphasizing the significance of time-aware implicit graph modeling. Additionally, compared to TGSLN without graph regularization (w/o G-Regularization), TGSLN performs better, demonstrating the effectiveness of the Gumbel-Softmax method in optimizing the graph structure by discarding unnecessary relations.

IRR Comparison with market indices. During the testing period, the A-share Market indices show a cumulative percent change of 28.8% (000300.SH) and 37.8% (399001.SZ), respectively, and the cumulative percent change of NASDAQ is -24.7%, all of which are lower than the IRR achieved by TGSLN. This indicates the ability of TGSLN to obtain excess returns and outperform the market indices.

5 CONCLUSION

In this paper, we propose TGSLN for stock sector ranking recommendation. Compared to previous work, we design a time-aware graph structure learning network to mine the implicit relationships within stock sectors. Meanwhile, we fuse the implicit relation graph with a predefined explicit relation graph based on stock sector classifications to enhance the relation graph structure. Experimental results demonstrate the effectiveness of our method.

ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China (No. 62272487).

REFERENCES

[1] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems* 33 (2020), 17804–17815.

[2] 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.

[3] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014.* 103–111.

[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 (TOIS)* 37, 2 (2019), 1–30.

[5] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[6] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.*

[7] Michael I Jordan. 1997. Serial order: A parallel distributed processing approach. In *Advances in psychology.* Vol. 121. 471–495.

[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] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[10] Heyuan Wang, Tengjiao Wang, Shun Li, Jiayi Zheng, Shijie Guan, and Wei Chen. 2022. Adaptive Long-Short Pattern Transformer for Stock Investment Selection. In *Proc. 31st Int. Joint Conf. Artif. Intell.* 3970–3977.

[11] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining.* 753–763.

[12] Wentao Xu, Weiqing Liu, Lewen Wang, Yingce Xia, Jiang Bian, Jian Yin, and Tie-Yan Liu. 2021. HIST: A Graph-based Framework for Stock Trend Forecasting via Mining Concept-Oriented Shared Information. *CoRR* abs/2110.13716 (2021).