Exploring Stock Graphs for Stock Price Prediction

Qifan Zhang
Department of Statistics and Data Science
Yale University
New Haven, CT 06511
qifan.zhang@yale.edu

1 Introduction

In this project, we aim to explore whether Graph Learning methods can improve the accuracy in predicting stock price. The importance of this topic is that first, an accurate prediction of stock price can support individual or institutional investments and second, a graph-based model can also reveal the dynamical interaction effects among different stocks, which may further support some theory in asset pricing or behavioral finance. The problem considered in this project can be formally defined as follows.

- **Stock Graph** − The stock graph is a dynamic graph $\mathcal{G}_t = (V_t, \{E_t\})$, where V_t is a list of stock and $E_t \subset V \times V$ is the edge set (examples are given below) at time t. We use $\{E_t\}$ to indicate that the graph may be heterogeneous. In this project, we assume V_t is a constant function of t for simplicity and we denote it by V. That is, we do not change the list of stocks over time.
- **Node Features** − Given a stock $v_i \in V$, we can define its features at time t as $X_i^{(t)}$ such as price and volume. In this project, we only consider 10 features for each stock for simplicity.
- **Node Labels** − The label $y_i^{(t)}$ for each stock at time t is the τ-step stock price such as stock price after 5/10/20 days. We can also consider the stock return as label but the loss of momentum information may bring difficulty in predicting.
- **Edges** − The edge set E_t describes the relationships between different stocks such as supplier-customer, sector-industry and price correlation relationships. It can be static or dynamic depending on relationship type. In our project, we consider stock business, sector-industry and price correlation relationship as our edge sets.
- **Our Goal** − Fit a function f so that $f(v_i, t; \mathcal{G}_t)$ is close to $y_i^{(i)}$.

In this project, we construct the stock graphs using sector-industry, wikipedia relation and price and volume data. Model-wise, we select a LSTM + GAT architecture and use LSTM as our baseline due to the lack of off-the-shell and reproducible work in this topic. In addition, to overcome the issue brought by multiple edge types (over 150 in our datasets), we consider the basis learning method to balance space efficiency and prediction accuracy. In addition, we propose a stock graph augmentation method to enable one stock attend to the historical trend of another stock in the message passing process. Our results show that utilizing graph structures on stock price prediction can reduce the MSE loss by over 50%. Also, the use of basis learning can improve the performance of the model trained simplified stock graphs by around 10%. The introduction of the augmented stock graphs also bring about 25% improvement on one dataset. Then we conduct sensitivity analysis on the number of attention heads and the number of bases and show that a larger and more complicated model can bring significant improvement. We also explore the dynamic of attention matrix in our project to show the explanability brought GAT architecture. The code of this project is uploaded to https://github.com/zh-qifan/Stock_Graphs_Price_Prediction.

Preprint. Under review.

Table 1: Comparison of the three architectures

	Stock Graph	Sequential Model	Graph Model	Reproducible?
[Feng et al., 2019]	Supplier-Customer and Sector-industry	LSTM	GAT	No
[Tian et al., 2022]	Correlation	Attention	GAT	Yes
[Sawhney et al., 2021]	Supplier-Customer and Sector-industry	LSTM	HyperGAT	No

2 Related Work

Stock graphs, in which stocks are represented as nodes and edges depict connections between stocks, serve as a valuable tool for comprehending the characteristics of the stock market. For stock price predictions, researchers have created stock graphs by analyzing co-occurrence in social media [Si et al., 2014], shared sector affiliations and supplier-customer relationships [Feng et al., 2019] and co-movement dynamics of stock price [Tian et al., 2022]. To incorporate the information contained in these graphs, many models have been applied and compared such as Vector Autoregressive Model [Si et al., 2014] and Graph Attention [Feng et al., 2019, Tian et al., 2022]. Here, we analyze the three models referred in this project.

The Relational Stock Ranking (RSR) [Feng et al., 2019] is a model trained on supplier-customer and shared sector affiliations graphs for stock return prediction. The supplier-customer graph is constructed by the several pre-defined relation based on wikipedia data. The sector affiliations graph is constructed by connecting all stocks from the same sectors. Thus the graph consists of several fully connected components. The model first processes stock sequential data with LSTM and then use GCN for aggregating information from its graphs. In fact, the GCN considered in RSR is a GAT model. Then embeddings learned from LSTM and GAT are then concatenated for downstream prediction tasks. However, the paper is lack of reproducibility since it uses a pretrained LSTM without giving the code to generate it. Thus their results should not be used as benchmark.

The second model is the Hybrid-attention dynamic graph neural network (HAD-GNN) [Tian et al., 2022]. The model is trained on the stock co-movement graphs constructed by stocks' correlation. The constructed correlation graph is a dynamic graph and depends on the hyperparameters such as window length and threshold. Model-wise, compared to RSR, it adds an additional temporal attention layer after the LSTM layer to focus more on recent data. The model is trained with a t-Batch method, which samples subgraphs on several dates for training at each step.

The third model is the Hyperbolic Stock Graph Attention Network (HyperStockGAT) [Sawhney et al., 2021], which leverage hyperbolic graph learning to model represent the complex, scale-free nature of inter-stock relations. The model also considers the supplier-customer and sector-industry graphs constructed in RSR. The Gromov's hyperbolicity calculated in the datasets considered in the model are all below 1.5, which supports the improvements brought by hyperbolic layers. The Sharpe Ratio, which is the ratio of portfolio return over standard deviation, of the stocks selected by HyperStockGAT outperforms those selected by RSR by around 30%, which also indicates the importance of optimizing the GNN layers for better prediction performance in stock prediction task.

The comparison of the three architectures is further summarized into the Table 1.

3 Methods

The model considered in this project is a LSTM + GAT architecture and the main contributions made by this project is that first, this project combines the supplier-customer, sector-industry and stock correlation graphs and shows the positive improvements provided by adding stock relationships; second, this project explores the use of basis learning to reduce the size of parameters when adding new types of stock relationship; third, this project explores the way of mixing temporal and cross-sectional data in the same graph to enable one stock to attend to the previous trend of another stock. The methods used in this project is further discussed below.

3.1 Stock Graphs

There are three types of stock graphs used in this project. **The supplier-customer graph** is about the first-order and second-order company relations from Wikidata. Company i has a first-order relation with j if there is a statement that has i and j as the subject and object, respectively. Companies i and j have a second-order relation if they have statements sharing the same object, such as *Boeing Inc.* and *United Airlines, Inc*. have different statements towards *Boeing 747* [Feng et al., 2019]. **The sector-industry graph** describes whether two companies belongs to the same sector. For example, *GOOGL* and *META* both belong to *Computer Software: Programming, Data Processing* industry which belongs to *Technology* sector. **The stock correlation graph** or the stock co-movement graph [Tian et al., 2022] is constructed by connecting stocks with high correlation in a historical time period. Given window length L and correlation threshold $\gamma > 0$, if the absolute value of correlation of stock A and stock B calculated in $[t - L + 1, t]$ is higher that γ, we add an edge between A and B at time t.

The three graphs serve different functions in our model. The sector-industry graph describes the inter-sector relationship while the supplier-customer and stock correlation graph can supplement cross-sector information.

3.2 LSTM + GAT

The LSTM + GAT architecture is straightforward. The input is a sequence data for each node and the LSTM module transforms it into an embedding vector. The embedding vectors are fed into the GAT module to add stock relationships information. Then the augmented embeddings are fed into a fully connected layer for stock price prediction. Although hyperbolic layers have been shown to have better expressive power in stock graphs [Sawhney et al., 2021], we select a GAT module because the training is faster and thus we are able to make more explorations.

3.3 Basis Learning

A stock graph may have multiple edge types and as we add more stock graphs, the number of edge types may increase rapidly. For example, two stocks in *Computer Software: Programming, Data Processing* industry usually have higher correlation than stocks in other industry. To reduce the size of parameters, we use the basis learning in heterogeneous graph processing. Since the principle of it was mentioned in our course before (See Lecture 7), we do not include it in our report.

3.4 Mixing Temporal and Spatial Graphs

In previous work in graph learning for stock price prediction, the processing of temporal data and relational data is usually separated. However, this is not the case in real world. When buying or selling in one stock at time t, traders may also consider the trend of another stocks at time t, $t - 1$, $t - 2$ or even earlier. To enable the model to have such ability, we construct an augmented graph by including nodes in $[t - L + 1, t]$ in the stock graphs at t. As shown in Figure 1, if v_i and v_j are connected at time t in the original stock graph, we add v_i^s and v_j^s for $s \in [t - L + 1, t]$ in the augmented graph and add $v_i^s \to v_j^t$ and $v_j^s \to v_i^t$ for $s \in [t - L + 1, t]$ in the augmented graph.

4 Experiments

4.1 Datasets

The two datasets used in this project are NASDAQ and NYSE stock datasets from 01/02/2013 to 12/08/2017. The two datasets has similar structures and contain three kinds of data: 1) historical price data, 2) sector-industry relations, and 3) supplier-consumer relation. The datasets can be downloaded from https://github.com/fulifeng/Temporal_Relational_Stock_Ranking [Feng et al., 2019]. Following the existing work on the datasets, we chronologically divide the full dataset into training (2013 - 2015), validation (2016) and testing set (2017). The number of stocks and dates considered in each dataset is summarized into Table 2. The statistics of the sector-industry, the supplier-customer and the stock correlation relations are summarized in Table 3.

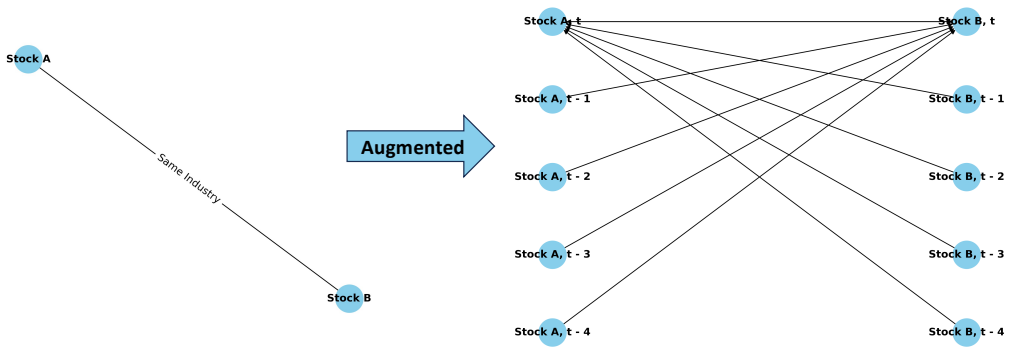

Figure 1: Mix Temporal and Spatial Information with $L = 5$

Table 2: Statistics of the datasets

Market	Stock#	Training Days# 01/02/2013 12/31/2015	Validation Days# 01/04/2016 12/30/2016	Testing Days# 01/03/2017 12/08/2017
NASDAQ	1,026	756	252	237
NYSE	1,737	756	252	237

4.2 Training

First, we construct the stock graphs in our project. The supplier-customer graph is constructed by connecting stocks sharing the same wiki relations in our dataset and the graph is static. The sector-industry graph is constructed by connecting all stocks within the same industry and the graph is also static. The correlation graph is constructed by comparing the correlation of two stock returns during a previous time period $[t - L + 1, t]$ with a given threhold γ. In our project, $L = 30$ and $\gamma = 0.7$. The correlation graph is dynamic.

Second, we construct the node features i.e. features for each stock at each time step. For simplicity, we only consider moving average features for price and volume with window lengths $[5, 10, 20, 30]$. The features are further standardized using mean and variance in training set for all stocks respectively.

Third, we construct the node labels. Although stock returns are more important in real settings, it is much more difficult to predict it than stock price due to the lack of momentum information. Thus in our project, we consider node labels as the stock price 5 days ahead.

In our experiment, we set the number of epochs to be large enough so that we always early stop. We set the patience for early stopping to be 30 and the minimal loss improvement to be 0.01. The loss function is set as mean squared error. At each epoch, we permute the time steps and train the model on the whole graph at each time step. We use the Adam optimizer with learning rate 0.01 and cosine annealing scheduler. The random seed is set as 1234. Other model hyperparameters used in specific models will be mentioned in Section 4.3.

Table 3: Statistics of different relations

	Sector-Industry		Supplier-Customer		Correlation (Average)	
Market	Types#	Ratio	Types#	Ratio	Types#	Ratio
NASDAQ	112	5.00%	42	0.21%	1	2.71%
NYSE	130	9.37%	32	0.30%	1	1.67%

Table 4: Experiment Results [MSE Mean (Std)]

Model	NASDAQ	NYSE
LSTM	8.45 (0.05)	2.93 (0.07)
LSTM-GAT-Agg	3.95 (0.09)	1.34 (0.17)
LSTM-GAT-Full	**3.48 (0.08)**	**0.83 (0.13)**
LSTM-GAT-Basis-5	3.84 (0.10)	1.51 (0.10)
LSTM-GAT-Basis-30	3.58 (0.13)	1.27 (0.11)
LSTM-GAT-Agg-Aug	4.27 (0.13)	1.01 (0.21)

Table 5: Sensitivity Analysis on the Number of Heads [MSE Mean (Std)]

Model	NASDAQ	NYSE
LSTM-GAT-Agg-head-3	3.95 (0.09)	1.34 (0.17)
LSTM-GAT-Agg-head-5	3.75 (0.12)	1.35 (0.14)
LSTM-GAT-Agg-head-7	**3.61 (0.12)**	**1.22 (0.11)**

4.3 Results

The results are listed in Table 4. The number of layers, sequence length and hidden size of LSTM are set as 2, 16 and 64 respectively. The number of layers, number of heads, ouput channels of GAT are set as 1, 3 and 64 respectively. Since the sector-industry graph is blockwise fully connected, one layer of GAT is enough to capture the stock relationships. In addition, we add 0.5 dropout on the attention matrix of GAT. The experiments are run three times to calculate the mean and standard deviation of loss. Since our goal in this project is to understand whether stock graphs can improve the stock price prediction, we simply pick the LSTM model as our baseline. The LSTM-GAT-Agg is the model trained on the aggregated stock graphs (only 3 edge types); the LSTM-GAT-Full is trained on all edge types (aroun 150 types in total); the LSTM-GAT-Basis-k means to train on all edge types with k number of bases; the LSTM-GAT-Agg-Aug means to train on aggregated stock graphs with augmented graph structures. From our experiment results, we can see that all models that utilize the stock graphs outperform the performance of LSTM. The largest model LSTM-GAT-Full achieve the best result on the two datasets and we attribute the model accuracy to the large amount of trainable parameters for the model to capture complex temporal and spatial relationships in stock price prediction. The LSTM-GAT-Basis-5 and LSTM-GAT-Basis-30 achieve outperform the LSTM-GAT-Agg by around 2.5% and 10% respectively. Here, the basis learning method is proved to be a good strategy to balance space efficiency and prediction accuracy. The LSTM-GAT-Agg-Aug outperform the LSTM-GAT-Agg on NYSE but not on NASDAQ.

4.4 Analysis

Given the long training time of our models, for the analysis of model hyperparameters, we only consider the number of heads of GAT and the number of bases in basis learning. In our results, LSTM-GAT-Agg-head-k represents LSTM + GAT model trained on the aggregated stock graphs with k number of attention heads. From the results, we can infer that more attention heads can improve our pediction accuracy. From the sensitivity analysis on the number of bases, we can see that more number of bases enables the model to approximate the performance of LSTM-GAT-Full but there may be an upper limit of the approximation since we observe that the LSTM-GAT-Basis-30 performs better than the LSTM-GAT-Basis-50 on NASDAQ. Nevertheless, this statement should be further tested with more numerical experiments.

In addition to sensitivity analysis, we also explore what the attention mechanism of GAT informs us about the dynamic of stock relationships. We consider the change of attention values of stock *AAPL* in NASDAQ and stock *BAM* in NYSE on their sector-industry and supplier-customer neighbors. We use LSTM-GAT-Agg on test set to do this experiment and pick the first head to visualize. As shown in Figure 2 and 3, we can see that on this attention head, *AAPL* attends more to CRAY and *SMCI* on its sector-industry neighbor and non bluechip stocks on its supplier-customer neighbor. For *BAM*, its

Table 6: Sensitivity Analysis on the Number of Bases [MSE Mean (Std)]

Model	NASDAQ	NYSE
LSTM-GAT-Basis-5	3.84 (0.10)	1.51 (0.10)
LSTM-GAT-Basis-10	3.80 (0.12)	1.52 (0.10)
LSTM-GAT-Basis-20	3.66 (0.07)	1.37 (0.12)
LSTM-GAT-Basis-30	**3.58 (0.13)**	1.27 (0.11)
LSTM-GAT-Basis-50	3.73 (0.11)	**1.12 (0.14)**

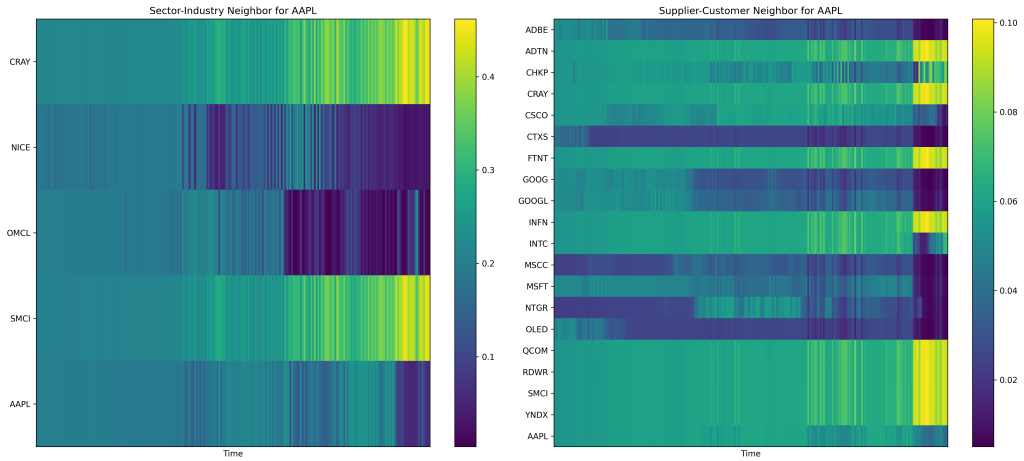

Figure 2: Attention Analysis for *AAPL*

attention on itself and *PDM* changes swiftly through time and it pays great attention to those large investment banks on its supplier-customer neighbor.

5 Conclusion

In this project, we develop a LSTM + GAT model to capture the stock relationships in three types of stock graphs: the supplier-customer, the sector-industry graph and the stock correlation graph. With the graph structures, our LSTM + GAT model can reduce the MSE of LSTM model by over 50%. To reduce the size of parameters brought by multiple edge types, we propose to use the ba-

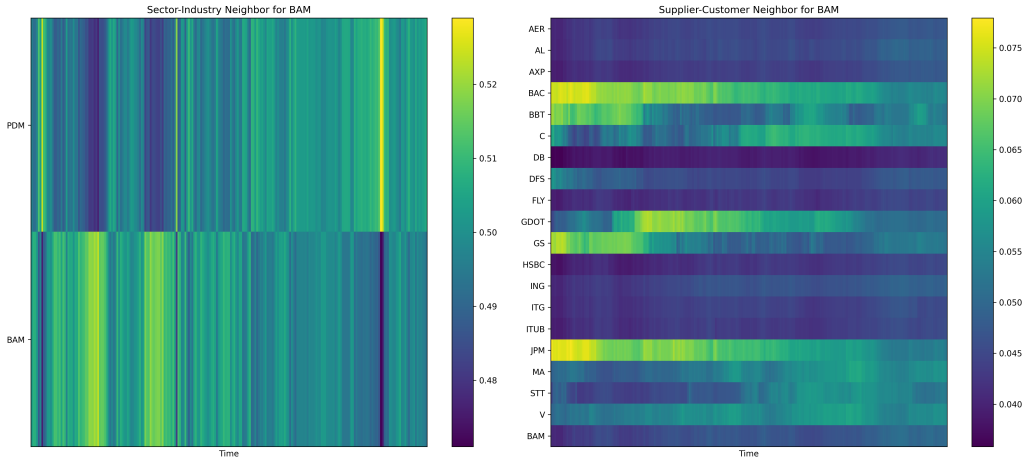

Figure 3: Attention Analysis for *BAM*

sis learning method and show that this method can be a good approximation of the model directly trained on the heterogeneous stock graphs. In addition, to capture the stock relationship with time delay consideration, we introduce a graph augmentation method by mixing the temporal and spatial relations. This approach is shown to have great potential in improving our graph learning method through our experiments. Finally, we explore the sensitivity of our models on the number of attention heads and the number of bases and conclude that stock price prediction requires a large and complicated model to capture the underlying relationship among different stocks. We also conduct an analysis on the dynamic of the attention matrix in our model and analyze how the attention values of *AAPL* and *BAM* change over time. The methods used in this project in reducing parameter size, introducing mixed relations in stock graph and attention analysis can also be generalized to other temporal graphs problem such as highway traffic forecasting and time-evolving social networks.

6 Reproducibility

Please see https://github.com/zh-qifan/Stock_Graphs_Price_Prediction.

References

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

Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah. Exploring the scale-free nature of stock markets: Hyperbolic graph learning for algorithmic trading. In *Proceedings of the Web Conference 2021*, pages 11–22, 2021.

Jianfeng Si, Arjun Mukherjee, Bing Liu, Sinno Jialin Pan, Qing Li, and Huayi Li. Exploiting social relations and sentiment for stock prediction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1139–1145, 2014.

Hu Tian, Xiaolong Zheng, Kang Zhao, Maggie Wenjing Liu, and Daniel Dajun Zeng. Inductive representation learning on dynamic stock co-movement graphs for stock predictions. *INFORMS journal on computing*, 34 (4):1940–1957, 2022.