Deep Learning with Graphs | UE22AM342BA2

Axe Capital Investments

Assignment 2

Namita Achyuthan	PES1UG22AM100
Shusrith S	PES1UG22AM155
Siddhi Zanwar	PES1UG22AM161

Introduction

This assignment is all about learning how to leverage hypergraphs with temporal data using a bunch of different approaches- hypergraph neural networks, graph attention transformers, and even traditional machine learning methods. We've been given a stock hypergraph with data ranging from 2019 to 2022, and the task is to predict how 20 different stocks will behave in 2023.

The idea is to go beyond just individual stock trends and look at the bigger picture- how stocks are connected, how those relationships evolve over time, and how we can use those patterns to make better predictions.

Modelling Approaches

We used three different methods:

1. Graph Attention Transformers
2. Hypergraph Neural Networks
3. Classical ML Approaches

An in-depth explanation of the same is available on the ipynb we submitted as part of the assignment deliverables.

The best performing method of the three was used for evaluation with the bling_test_cases.json. A sample of the document is as follows:

predictedFromTheBlind.json


```json
[
    {
        "ticker": "GOOGL",
        "date": "2023-03-01",
        "predicted_open": 87.29524576020822,
        "predicted_high": 87.74281170114216,
        "predicted_low": 87.07152210193077,
        "predicted_close": 87.76804580688477,
        "predicted_volume": 20886870.08,
        "eval_metric": "MAPE"
    },
    {
        "ticker": "GOOGL",
        "date": "2023-03-08",
        "predicted_open": 87.29524576020822,
        "predicted_high": 87.74281170114216,
        "predicted_low": 87.07152210193077,
        "predicted_close": 87.76804580688477,
        "predicted_volume": 20886870.08,
        "eval_metric": "MAPE"
    },
```

Comparative Evaluation

Metrics used for evaluation

MAPE shows the average error as a percentage, making it especially useful for graphs where node features vary widely in scale. RMSE tells you how far off your predictions are on average, but it can be skewed by large outliers.
For hypergraph-based models, MAPE offers clearer, more interpretable feedback. We took MAPE as the main indicator of model performance.

Side-by-side performance comparison

Model	RMSE	MAPE	Validation Loss
GAT	5974327.3693	37.07%	0.1692
HGNN	6033094.6001	34.62%	0.1688
Classical ML	406783.2500	1.6407%	-

Strengths and weaknesses of each method

GAT

Strengths	Weaknesses
GAT allows each node to selectively focus on its most important neighbors, leading to smarter message passing.	Attention is restricted to 1-hop or 2-hop neighbors.
Combines local learning with global pooling to generate meaningful graph-level representations.	Global mean pooling may overlook hierarchical structures.
Compared to deeper or more complex GNNs, this model is efficient and converges fast.	-

HGNN

Strengths	Weaknesses
Captures higher-order relationships between stocks via shared sectors; more expressive than traditional GCNs.	Lacks adaptability to dynamic hyperedge structures

Snapshot-based approach is modular and easy to scale to longer time series.	No temporal data embedded within features- each day is treated independently.
MSE with Adam optimizer and learning rate scheduling ensure stable training.	Misses out on temporal dependencies that could improve predictive power in sequential patterns.

Traditional ML Models

Strengths	**Weaknesses**
Easy to implement and understand, especially with feature importance from RandomForest.	Moving averages offer limited temporal insight compared to sequence models like LSTMs or Transformers.
RandomForest handles non-linear relationships and works well with structured data.	Misses the benefits of attention in capturing dynamic feature relevance across time.
Hypergraph-based features like sector and moving averages still influence predictions.	Once features are engineered, model cannot adapt to new temporal patterns or context dynamically.

Conclusions

While all three modeling approaches offered unique advantages, the Graph Attention Transformer (GAT) and the Hypergraph Neural Network (HGNN) performed almost identically, with only minute differences in evaluation metrics. Both models were able to leverage the structural properties of the hypergraph effectively, capturing meaningful inter-stock relationships.

One notable challenge we faced across models was in handling the Volume attribute, which at times introduced inconsistencies or noise in the predictions. Despite this, the models remained robust, and the classical ML approach- though much simpler- also performed reasonably well given strong feature engineering.

Overall, our experiments highlight that structural relationships and careful temporal feature design can be just as important as the model architecture in financial forecasting tasks.