StockMixer: A Simple yet Strong MLP-based Architecture for Stock Price Forecasting

Anonymous submission

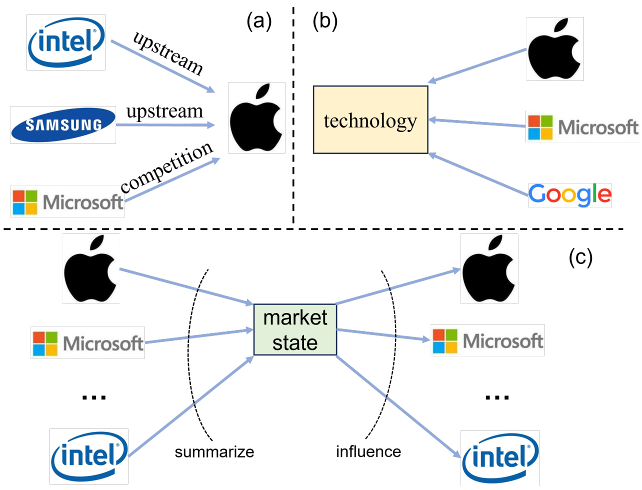

Figure 1: Some examples of three relationship utilization. (a) multi-relation graph; (b) hypergraph; (c) our proposed S-Mixer.

Difference among Relational Modules

GNN-based methods require extensive external concepts such as industrial category or inter-firm relations to construct effective predefined graphs, increasing the difficulty in data acquisition. Even when utilizing Graph Structure Learning techniques, the incorporation of prior graphs remains inevitable. Furthermore, previous researches focus on classic graphs and hypergraphs. The former in Figure1 (a) construct pairwise relations between individual stocks, hypothesizing that a company is affected by only several other related enterprises Hypergraph-based method refute the classic graphs, claiming that stocks are often related through higher-order relations as a collective group like Figure1 (b). However, stock entities always exhibit heterogeneity, whereas graphs are based on the assumption of homogeneity in the same industry and smoothness for node message transmission. As shown in Figure1 (c), we further recognize that the share trends are conducted by certain market states, which is summarized from all involved stocks.

Efficiency Analysis

The complexity of Graph Neural Network contains feature mapping of vertices and message passing. Support that a graph $\mathbf{G}$ has N nodes, E edges and representation dimension F, then the computation cost is $\mathbf{O}(NF^2) + \mathbf{O}(EF)$. And most of them encode the inputs with LSTM, which costs $\mathbf{O}(Td^2)$. T denotes the time steps and d denotes the hidden dimension proportional to T. Our StockMixer decouples the parameters into Indicator, Temporal and Stock respectively corresponding to $\mathbf{O}(F^2)$, $\mathbf{O}(T^2)$ and $\mathbf{O}(mN)$, excellent performance advantages in processing large-scale market data. To validate our efficiency, we conduct a series of models in 1 to show the parameter complexity and training time per epoch of different methods. LSTM is an efficient channel-independent model but suffers from poor prediction performance. Our StockMixer has high training efficiency and outperforms all the comparison methods.

Model	# of params	training time(s)
LSTM	14.28K	3.011
LSTM-RGCN	8464K	8.976
GAT	1039K	7.388
RSR-I	9493K	6.195
STHAN-SR	56.72K	7.085
ESTIMATE	185.3K	6.678
StockMixer	45.08K	4.242

Table 1: Training efficiency on NASDAQ.

References