Different Scenario , Different Meaning
-AI Stock Trade Method-

September 20, 2024

1 Abstract

Despite of stock market's trait that makes stock movement to be random, there's many similarity meaning similar shapes, distinct patterns , properties in each stock movement. Since participants in stock market have clear purpose and there always exist asymmetric information. From this, I suggest data preprocessing algorithm cutting data period in a certain criteria so whole input period is divided into two parts. Which I call it DSDM(different scenario , different meaning) model. By DSDM, I will clarify estimation based on data divided by certain criteria is better than just putting like 5-day data or 20-day data to deep learning model.

2 Introduction

Stock market has dynamic nature and therefore has extremely strong noise. so, there's a limitation in predicting stock movement. However, there are some clear logic in common sense. And I use two of them in this paper to build a model. First, some people have more information that are directly related to stock movement than majority people who bought/will buy those stocks, meaning insider information , rumors , etc. Second, people try to gain profit as safe as possible and maximize profit, meaning they want to buy stock before stock uprising. With these two common sense, this paper will follow a shape of convex function that only goes down first and then go es up. This has two benefits. Frist, pattern is easy to recognize. Second, it shows recent movement after price went down.

Trying rough estimations and changing model to reflect market change periodically were mainly used in the field. And in the research field, dividing long term trade, short term trade based on trading frequency [1] , using candle chart [2] were attempted. Especially, frequency tracking showed good performance since frequency can be an indicator of people's thinking whether this stock will increase or not in a long term. Also, not only analyzing individual stock's historical repetitive patterns but also trying to figure out correlations between stocks in the same sectors and global influence on each stock was tried.[3] Text data, emotion data were also used nowaday. But it is still hard to figure out market's intentions.

3 Approach

In this section, I will present simple one DNN model and some data preprocessing algorithm.

3.1 Data Preprocessing (1)

Step	Description
1	**Input:** Stock Data: D = [Date, Open, High, Low, Close, Volume]
2	**Output:** Preprocessed Data with Target Labels
3	**Parameters:** Minimum Threshold: $min\_threshold = 1.03$, Maximum Threshold: $max\_threshold = 1.06$
4	**For each day** t **in dataset:** Calculate $Target_t \leftarrow \left(\frac{High_{t+1} - Close_t}{Close_t}\right) + 1$
5	**If** $Target_t < min\_threshold$ Remove row t from dataset
6	**For each day** t **in filtered dataset:** **If** $Target_t \geq max\_threshold$, $Label_t \leftarrow 1$ **Else**, $Label_t \leftarrow 0$
7	**For each day** t **in dataset:** **If** $Close_t > Close_{t+1} \geq Close_{t+2}$ and $Close_t > Close_{t-1} \geq Close_{t-2}$, t is an input to day cutting algorithm
8	**For each row** t**:** $Normalized\_Close_t \leftarrow \frac{Close_t - \min(Close)}{\max(Close) - \min(Close)}$
9	**For each day** t**:** $Mx_t \leftarrow \max(Close_t, Open_t)$ $Mn_t \leftarrow \min(Close_t, Open_t)$
10	**Return:** Preprocessing dataset with normalized 'Close' values and target labels

3.2 Day Cutting Algorithm

(Day Cutting Algorithm)

(input) : data , day(resulted by data preprocessing (1) setp 7)

(return) : from i to len(day), data[i, day[i]+1:20] = 0

Features after day cutting algorithm become partially convex. Now model is able to figure out and focus on what decrease/increase in the past few days affect to future price change. This criterion identifies the first point in time, moving backward from the present, where a first decline in stock prices is observed in a macro perspective.

3.3 Additional algorithms

Step	Description
1	For Row t, If $5\_day\_MA[t,0] - 5\_day\_MA[t,1] >$ $(5\_day\_MA[t,1] - 5\_day\_MA[t,2]) \times$ constant then $Close[t,:] = 0$
2	For Row t, If $Mx[t,0] - Mn[t,0] <$ constant $\times$ max$(Mx[t,1:\text{day}[t]] - Mn[t,1:\text{day}[t]])$ then $Close[t,:] = 0$

Table 1: Second Stage For Preprocessing Data

3.4 DNN Model

DNN model is below. And activation function is relu() which is widely used nowaday to give non linearity to output.[4] Main model's output layer's out features=3

I used two simple layers instead of transformer architectures. Despite of a lot of cost to train, transformer sometimes is not robust enough to variations than simple models. [5]

(Main)

(swish) : SiLU()

(dropout) : Dropout(p=0.1, inplace=False)

(l1) : Linear(in features=20, out features=128, bias=True)

(l2) : Linear(in features=128, out features=3, bias=True)

In this paper, baseline is only Main model. Proposed model is additional data changes algorithms plus day cutting algorithm plus Main model.

4 Experiment - Train

4.1 Data Setting

Train set for models is a seperated dataset of Kospi stock dataset. 500 stocks in kospi(2022-04 : 2024-09) stock dataset are for train set and test set. Ratio is 70 percentage for train and 30 percentage for test. 496 stocks in Kosdaq(2022-04 : 2024-09) and Whole SP 500(2007 : 2016) stock data are for backtesting. four stocks, '042660','096300','099340','099350' in Kospi dataset are excluded for raising exception error.

Data	**input0**	**input1**	**...**	**input18**	**input19**
221	0.9	0.3	...	0	0.24
222	0.8	0.1	...	0.71	0.35

Table 2: Data Structure Example

4.2 Hyperparameter

Batch size of data is 256, learning rate is 0.004 for Main model. Hidden dim for main model is also 128.

4.3 Train

Paper's model and baseline Dnn model with kospi train dataset,

Model	**Train_Loss**
Proposed Model	0.00012
Baseline	0.61828

Table 3: Training Test Result Of Two Models

Baseline model consist of exactly same main model and input data without using day cutting algorithm and other algorithms mentioned in this paper. Other things, hyperparameter and target are all the same. In additional algorithm, if condition satisfies then some inputs go to zero and targets become zero. There's a large target class imbalance resulted in a huge difference between train loss of proposed model and baseline.

5 Experiment-BackTesting

5.1 Data Setting

Kosdaq(2022 : 2024) for Backtesting. S&P(2007 : 2016) 50 stocks dataset is also for Backtesting. For Kospi and Kosdaq dataset, I collected these data from API I personally use.

5.2 BackTesting

Using pyalgotrade for individual stock, outputs from proposed model are 1 then broker buys stocks as possible. If current day is 30 days past the purchase date and current high price is higher than 1.5 * buying price, broker sells. If this condition does not satisfies until 400 days past the purchase date then broker just sells whatever the price is.

Proposed Model	**Mean Rate Of Return**	**Mean Number Of Trades**	**Number Of Shares Traded**
Kosdaq Dataset	13.134	1	39
S&P Dataset	12.193	1	9

Table 4: Results of Kosdaq and S&P Dataset for Proposed Model. S&P Data Source Link: `https://github.com/fulifeng/Adv-ALSTM/tree/master/data/kdd17/price_long_50`.

Baseline Model	**Mean Rate Of Return**	**Mean Number Of Trades**	**Number Of Shares Traded**
Kosdaq Dataset	3.684	1.176	266
S&P Dataset	16.842	1.586	29

Table 5: Results of Kosdaq and S&P Dataset for Baseline Model.

The mean rate of return can be calculated using the following formula:

$$\text{Mean Rate Of Return} = \frac{\sum_{i=1}^{n} \text{Rate Of Return}_i}{n}$$

Where:

- n is the number of stocks.
- Mean Rate Of Return$_i$ is the rate of return for each individual stock over the past ten years calculated without commission , trade fees.

The mean number of trades can be calculated using the following formula:

$$\text{Mean Number Of Trades} = \frac{\sum_{i=1}^{n} \text{Number Of Trades}_i}{n}$$

Where:

- n is the number of trades.
- $\text{Number Of Trades}_i$ is the number of trades for each individual stock over the past ten or seventeen years.

Result shows proposed model is better at understanding meaningful patterns in random noise contrast to baseline model. Since increasing of Stock price 100 days ago can affect to current price change sometimes. But sometimes it is totally meaningless. Now stock tries to go upward but 30 days ago there might be a price decline due to the bad event. Sometimes past day affect exactly the opposite to current movement. Only focusing on convex set makes model to see if after decline will it rise or not.

Baseline model predicted many stocks that rose by hundreds of percent whereas Proposed model did not. And baseline model also predicted stocks declined by tens to almost hundred percent.

Normally most of researches were conducted with the same market dataset for training and back testing. But this simulation check generalization ability by using different market data for training and back testing.

During the S&P500 dataset period, there was a financial crisis and index dropped for a while to 680 point but restored 2007's 1500 point and increased to 2200 point. But Baseline model's good performance for S&P 500 does not guarantee that model has a generalization ability and does not overfit since it showed a poor performance in Kosdaq dataset. Kosdaq's index was around early 700 in 2024-09 and was about 900 in 2022-04. When index goes down, compared to proposed model, baseline model does not hold lower bound. But Proposed model sustain a high profit and does not show any sudden drop.

6 Conclusion

Transformation of data type from time series to non time series will help solve data scarcity problem. Looking at specific data period, data have some geometric properties. It will give us novel insight that time series can be solved by regarding time series data structure as certain type of geometric shapes.

A lot of papers present test accuracy , sharpe ratio but it does not guarantee a minimum profit since often test set, validation set are seriously different from current real market. Patterns change because participants in market always try to find better solutions compared to current strategies.

If test , validation period is not long enough, model tends to overfit a lot. And this paper conducts huge period Backtesting to insist on common patterns existing in a certain period.

I suggested a result and numbers that prediction is different whether it is based on C period and A period. From this paper , I want to propose a message that I am not the same person when I'm in A group and when I'm in C group even if C contains A.

7 Future Experiment

Due to the lack of the number of data, had to do some trick to input data for making model performs good enough. By this, I need to set API to get data. I use some companies' api but didn't have time to connect it to produce csv type data files and use it. So I will try this.

I will use Deep reinforcement learning's stochastic policies to make the same effect as local model , preprocessed data do. Even if we heuristically expect similar results, sometimes stock skyrockets, fall down. So, stochastic policy such as PPO, SAC can be a key. Hence, this topic will be the next experiment object.

References

[1] Liheng Zhang, Charu Aggarwal, Gui-jun-Qi.*Stock Price Prediction via Discovering Multi-Frequency Trading Patterns*. 2017.

[2] Rosdyana Mangir , Irawan Kusuma1, Trang-Thi Ho, Wei-Chun Kao, Yu-Yen Ou1 , Kai-Lung Hua. *Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market*. 2019.

[3] J Yoo, Y Soun, Y Park, U Kang. *Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts*. 2021.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015.

[5] Akshita Jha, Adithya Samavedhi, Vineeth Rakesh, Jaideep Chandrashekar, Chandan K. Reddy. *Transformer-based Models for Long-Form Document Matching: Challenges and Empirical Analysis*. 2023.