LSTM For Stock Market Prediction

Group F: Dian Jiao, Yu Li, Xiaomeng Chen, Animesh Jain

Dec 2018

1 Introduction

Long short-term memory is a a specific type of recurrent neural network model with the ability to remove or add information to the cell states. It takes sequential time series data as its input, which is intuitively suitable for financial time series predictions.

We developed a two-class classification model using long short-term memory model with the information including daily price return, daily volume growth and daily sector performance of all S&P 500 index constituents from 11/2002 to 10/2018 to capture the stock market performance in a systematic approach. The accuracy is over 51 percent. Based on the prediction, we back-test three portfolio construction strategies including 130-30, market neutral and long-only. With the average daily return of 0.082% for 130-30 strategy prior to transaction cost and average daily return of 0.062% for 130-30 strategy after transaction cost, we find that long short-term memory is indeed suitable in the domain of stock market prediction. Our work is an extension of [1].

2 Data

We collect daily closing price, daily volume and GICS sector information of all S&P 500 index constituents, and the daily closing price of S&P 500 Sector GICS Level 1 Index from 11/2002 to 10/2018 using Bloomberg terminal. We write Python programs to merge all Excel files from Bloomberg terminal to create one single file which contains cross-sectional time series data. The code for this part is included in 1_Data_Colletion.ipynb.

2.1 Training and testing sets

We define a study period of 1000 consecutive business days. The training set includes the first 750 business days and the testing set includes the last 250 business days. The rolling box for each study period is 250 days which results in non-overlapping study periods. For each feature vector, we use past 240 days' feature space to predict one day ahead and keep rolling forward. We consider all stocks with the history they have available. If a stock does not exhibit the length of the full study period, it is considered for training or testing up until that day. If a stock does not exhibit the length of even one feature vector (241 days), it is not considered in our training sets or testing sets. The code for this part is included in the first half of 2_Data_Preparation.ipynb.

2.2 Feature space and target

We use the daily closing price return, daily volume growth, daily sector performance as three engineered features in our LSTM model. First, we calculate one-day arithmetic return of stock price, volume and sector index price using formula

$$R_t = \frac{P_t}{P_t - 1} - 1$$

Then, we normalize all three features by subtracting the sample mean and divided them by the sample standard deviation of the training set using the formula

$$\tilde{R}_t^{m,s} = \frac{R_t^{m,s} - \mu_{train}^m}{\sigma_{train}^m}$$

The target is defined as two equal-sized classes. Class 1 denotes the return of stock s in period $t+1$ is greater than or equal to the cross-sectional median return of all stocks in period $t+1$. Class 0 denotes if the return of stock s in period $t+1$ is smaller than the cross-sectional median return of all stocks in period $t+1$. The code for this part is included in the second half of 2_Data_Preparation.ipynb.

3 LSTM networks

The LSTM we used in our research work follows the introduction of [2]. The LSTM network consists of an input layer, one or more hidden layers, and an output layer, and the hidden layer contains memory cells, with a cell state, a forget gate, an input gate, and an output gate. The specified topology of the trained LSTM network is as follows:

- Input layer with 3 engineered features and 240 timesteps
- LSTM layer with 50 hidden neurons. This congfiguration yields 10,800 parameters
- A sigmoid activation function for the single neuron indicating the Bernoulli outputs whether the stock outperforms the market

For training the LSTM network, we apply the lower-level Python API of TensorFlow to engineer the model. We apply RMSprop, a mini-batch version of rprop, as the optimizer, which is believed to be a good choice for RNNs. Also, we set a dropout ratio of 0.1 to avoid overfitting and a maximum trainging duration of 1,000 epochs.

4 Model evaluation

4.1 Model Accuracy

First, we analyze the prediction accuracy by constructing a portfolio consisting of 2,000 stocks every day. The output of our LSTM model is the probability that stock s will be in Class 1 at date $t+1$. We rank the probability in an ascending order every day and select the top k stocks to short and bottom k stocks to long. We select k from $[10, 50, 100, 150, 200]$ and find that the highest accuracy is achieved at $k = 10$. Hence, we will focus our further analysis on the portfolio with 20 stocks.

In addition, we also separate our stocks to different sector to see if this model has a sector preference. As shown in Figure 1, the prediction accuracy for each sector is very similar to each other. Hence, we can conclude that our LSTM model does not have a clear sector preference.

4.2 Trading strategies and back-testing

We construct our portfolio using the following strategies based on the output of the LSTM model.

- Long only strategy: long k stocks, equal-weighted, daily rebalancing
- Market neutral strategy: long the top k stocks and short the bottom k stocks, equal-weighted, daily rebalancing
- 130-30 strategy: short 30% of stocks, use the money from short selling to long stocks

First, we calculate the average daily return, standard deviation and accumulative return for three different strategies prior to transaction cost. The results are plotted in Figure 2, Figure 3, Figure 4 and Figure 5. The 130-30 strategy has the highest average daily return (0.082%) with the highest standard deviation (0.02109). The long only strategy has the medium average daily return (0.056%) with the medium standard deviation (0.01915). The market neutral strategy has the lowest average daily return (0.043%) with the lowest standard deviation (0.00763).

The accumulative returns are shown in Figure 6 and Figure 7. We can see that 130-30 strategy outperforms the other two strategies after 2008, owing to the bullish market in the latest 10 years; long only strategy provides the second largest average daily return, but it has a large drawdown period between 2007 to 2008 because of serious financial crisis; market neutral strategy provides relatively stable profit regardless of the market, but it has the smallest average daily return. The strategies all have two drawdown periods between 2007 to 2008 and 2015 to 2016.

Furthermore, we consider transaction cost in our three strategies. We deduct $2bps$ transaction cost everyday for market neutral and 130-30 strategies and deduct $1bps$ transaction cost every day in long only strategies since long only strategy only has k stocks in the portfolio. We realize that all three strategies are still able to produce positive return over long term period. The 130-30 strategy has the highest average daily return (0.062%) with the highest standard deviation (0.02109). The long only strategy has the medium average daily return (0.046%) with the medium standard deviation (0.01915). The market neutral strategy has the lowest average daily return (0.023%) with the lowest standard deviation (0.00763). The trend of accumulative return for long only strategy and 130-30 strategy are similar to the trend prior to transaction cost. The profit of market neutral strategy has been badly influenced after 2014 due to the bullish market and high transaction cost. The code for this part is included in 4_Accuracy_Backtesting.

5 Conclusions and outlook

In this project, we apply long short-term memory model to predict stock market from 11/2002 to 10/2018 using S&P 500 index constituents. Also, we use the predicted results to construct portfolios and back-test the performance.

5.1 Conclustions

We have made two conclusions in our project. First, we have proved that the LSTM model is able to capture the trend of stock market in a systematic approach by including different engineered features. Second, we have proved that the LSTM model is capable to accumulate long-term stable profits, by back-testing three common portfolio construction strategies including 130-30, market neutral and long only based on the output of the LSTM model.

5.2 Outlook

Our research has improved the work of [1] by introducing new features (daily volume and GICS sector information) to the LSTM model. The accuracy and back-testing performance has been enhanced by our approach. Besides, we collected the latest data (2010 to 2015) and tested the model on these new data.

Due to the constraints of the compuational power, we have not devoted a great deal to the selection of the setting of parameters. Looking forwards, we may test the model against more sets of parameters and training approaches. Also, collecting additional data and adding features might improve our work as well.

Appendix

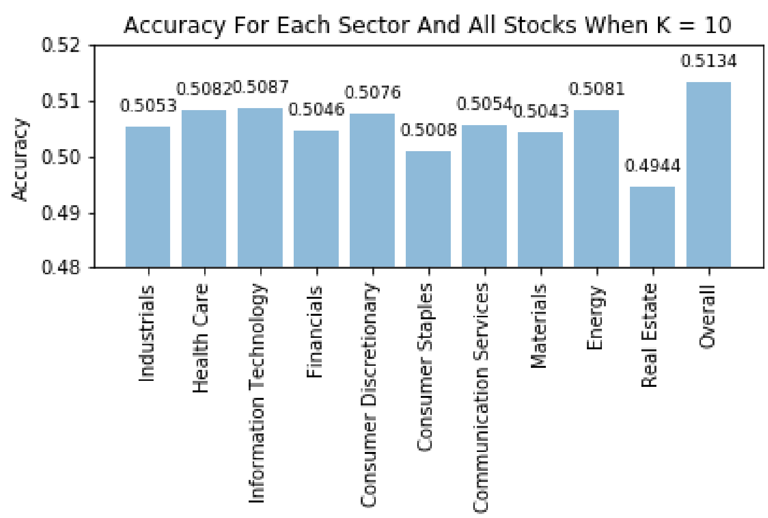

Figure 1: Prediction Accuaracy for $k = 10$

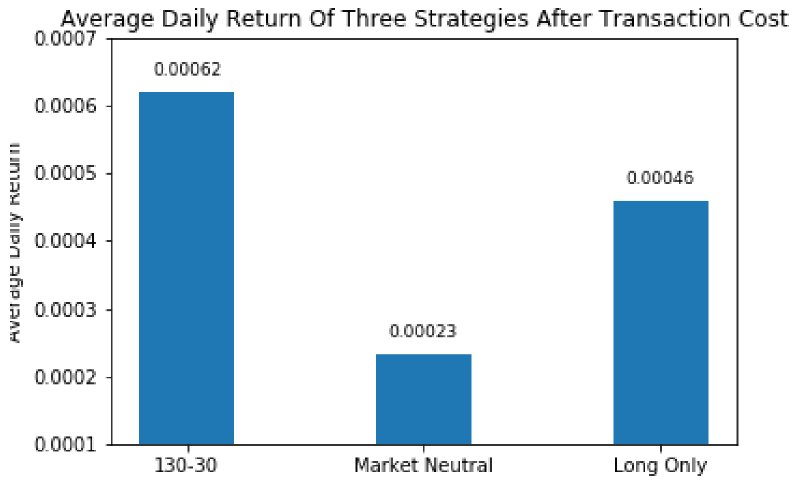

Figure 2: Average Return after Transaction cost

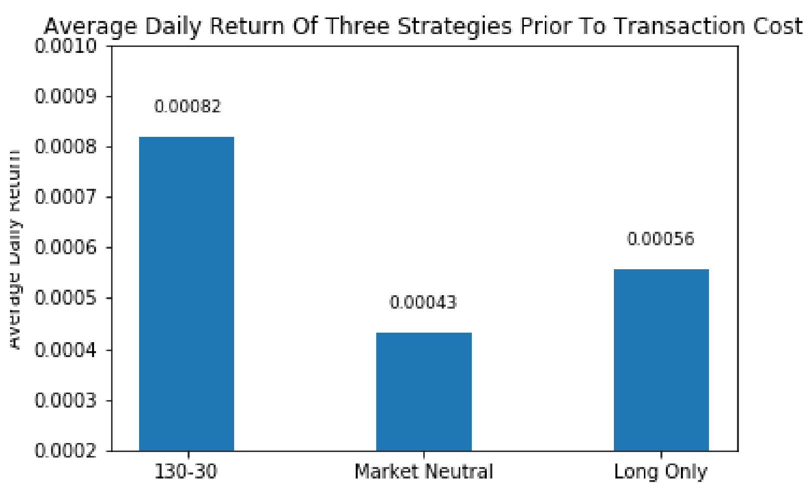

Figure 3: Average Return before Transaction cost

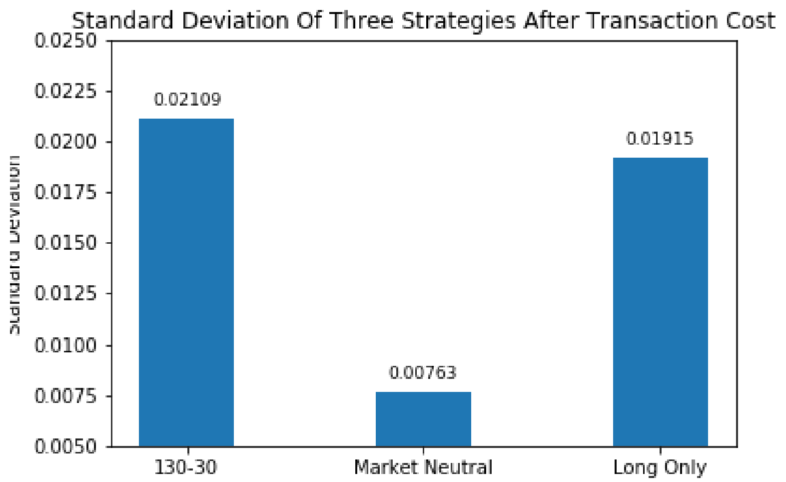

Figure 4: Standard Deviation aftre Transaction cost

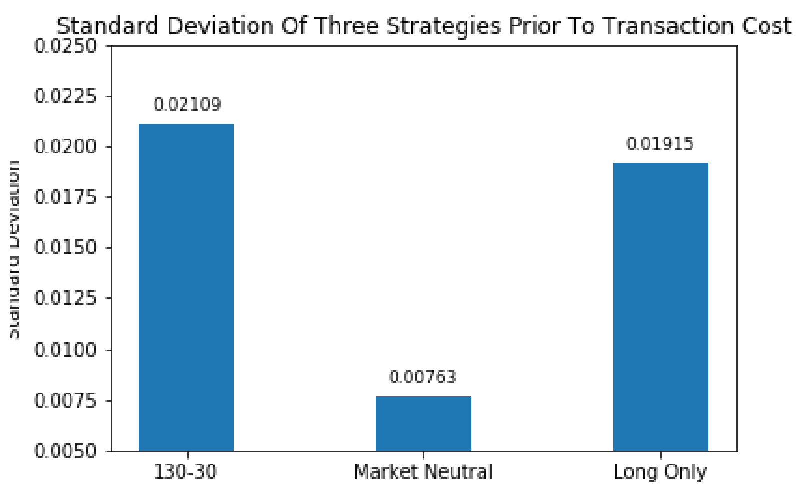

Figure 5: Standard Deviation before Transaction cost

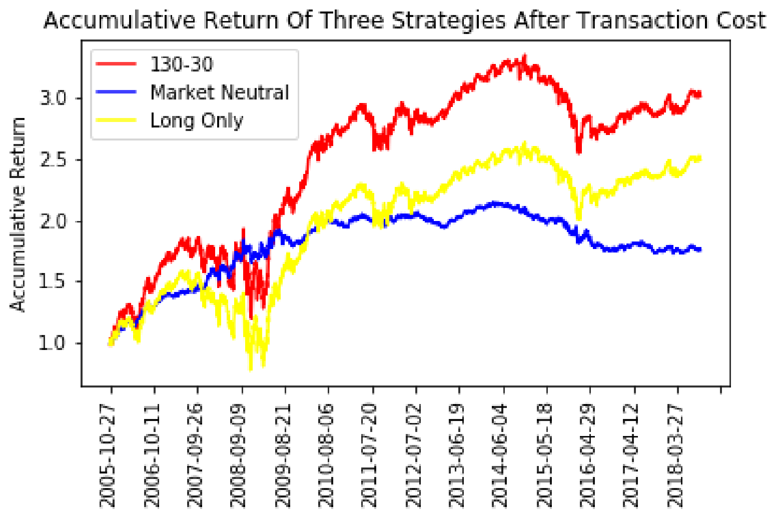

Figure 6: Accumulated Return after Transaction cost

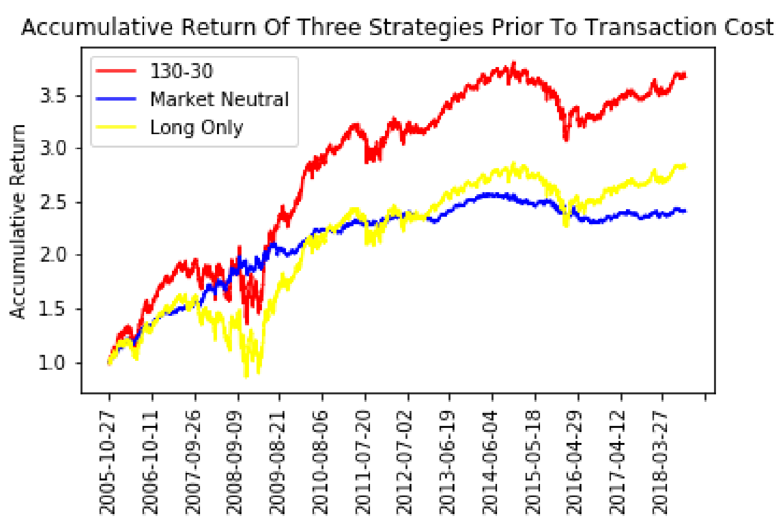

Figure 7: Accumulated Return before Transaction cost

References

[1] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.