LNCS 10943

Ying Tan
Yuhui Shi
Qirong Tang (Eds.)

Data Mining and Big Data

Third International Conference, DMBD 2018
Shanghai, China, June 17–22, 2018
Proceedings

Springer

Check for updates

Deep Stock Ranker: A LSTM Neural Network Model for Stock Selection

Xiaolin Zhang and Ying Tan(✉)

Key Laboratory of Machine Perception (MOE), Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
{xlzhang92,ytan}@pku.edu.cn

Abstract. Stock prediction is a great challenge for the past decades because of the fact that it is a non-stationary, noisy, chaotic environment. Traditional stock prediction models including statistical and machine learning based methods almost use handcrafted features as input. With the development of deep learning, end-to-end models achieve state-of-the-art in many other tasks. However financial time series data is too noise to apply end-to-end models straightly, instead of predicting stocks' absolute future return, we propose a novel stock selection model *DeepStockRanker* to predict stocks' future return ranking. Experimental results show that our method is able to extract information from raw data to predict stocks' future return ranking and achieves much better performance compared with several advanced models.

Keywords: Stock selection · End-to-end · LSTM · Learning to rank

1 Introduction

The stock market is the real economy's "barometer" and "alarm", its role is not only valued by the government, but also paid attention by individual investors. For stock investors, the more accurate the stock price trend forecast in the future, the more certainty is the availability of profits and the avoidance of risks; For the government, economic development and financial construction of the country also need precise and robust stock prediction. Because of its importance, predictions on stock market has been widely studied by academia and industry for many decades. But financial time series is non-stationary, high-noise and chaotic [6], how to effectively model the time series so as to predict the market more effectively and more precise is a big challenge.

In recent years, there is huge amount of studies in this field. The main research focus on two aspects. The first is the use of public financial opinion or news on the internet to make predictions on the future stock prices.

Compared with the analysis of financial news, modeling historical stock price can be a more direct analysis method. The old-fashioned prediction method is linear based time series forecasting statistical methods, such as

Y. Tan et al. (Eds.): DMBD 2018, LNCS 10943, pp. 614–623, 2018.
https://doi.org/10.1007/978-3-319-93803-5_58

ARMA, ARIMA [1], GARCH [8], etc. The popular prediction method is to use machine learning based approaches, such as neural networks, support vector machines, to model historical price series [5]. However, these proposed machine learning methods is highly dependent on the data representation and feature extraction [4].

Different from aforementioned tasks, this paper considers how to end-to-end train a deep neural network for stock prediction without handcrafted feature extraction. The end-to-end manner of our method avoids separating feature design, such as technical analysis indicators, from stock prediction model as a two-stage procedure, which is the typical way of the methods using handcrafted features. End-to-end manner is applied successfully and achieves state-of-the-art in many tasks, e.g., machine translation, image processing and speech recognition. However, financial time series data is different from image data, audio data and etc. And its noise is too large to cause the information contained in the data to be relatively small. Handcrafted feature extraction is to add prior knowledge to data processing, extract effective information from high-noise data, and then modeling. Since neural networks is a very powerful and well-fitted model, training directly on high-noise data often leads to over-fitting results. Without handcrafted feature extraction, we propose a new mechanism taking advantage of learning to rank to solve the problem of too much noise during training. As for sequence data, we choice the LSTM (Long-Short term memory) network [15]. It is more capable to handle long sequences of input when compared to other recurrent neural networks that are only able to memorize short sequences. By combining LSTM network, a new ranking loss function and a new batch-training manner, we proposed *DeepStockRanker* to learning to rank the stock's future returns according to their historical price.

Experimental results demonstrate that this approach is able to rank the stock's future returns accurately. And by selecting the highly-ranked stocks, we can get much higher returns than the market average. Compared with the aforementioned approaches with handcrafted feature extraction, our approach also shows statistically significant improvement. The main contributions of this work are the followings: (1) a new stock selection model using deep learning based technique; (2) a newly designed loss function for noisy financial sequence data and stock rank task; (3) no handcrafted feature extraction.

The article is organized as follows. In the next section, the related works about stock selection tasks and LSTM are presented. Then, the framework and the details of our method is presented. In the Sect. 4, experimental results will be shown. In the last, we conclude with a discussion.

2 Related Work

In recent years, researchers have conducted in-depth research and many experiments in this field. The main research focuses on two aspects: the first is to forecast the future stock price by using the public opinion or news analysis on the network; the second is to make the stock price prediction by modeling the stock price based on historical stock price series itself or its derived technical indicators.

As web information grows, recent work has applied Natural Language Processing (NLP) techniques to explore financial news for predicting market volatility. For example, Xie et al. [31], Tetlock et al. [28] and Ding et al. [9]. Park and Shin [23] analyzed the economic data and predicted the stock price movements.

As for modeling of historical stock price, a popular prediction method is to use neural networks or support vector machines to fit historical price series. Guresen et al. [13] summarized the experiments and results using BP neural network to predict the market index. Dong et al. [10] used neural networks to model the changes in stock prices after one and more steps respectively, and compared the effects of these models. Nelson et al. [22] studied the usage of LSTM networks to predict future trends of stock prices based on the price history, alongside with technical analysis indicators. And Azevedo et al. [2] made a phased summary of all the methods of modeling and forecasting using historical time series.

3 Deep Stock Ranker

The Deep Stock Ranker model we propose has four main components. First, because of the nonstationarity of price sequence data, some simple data preprocessing methods are adopted. After data preprocessing, the clean data is fed to a RNN model with LSTM cell to obtain deep representation of time series information. A one-layer and one-output-unit feedforward neural network is connected with the RNN and convert deep representation of time series information to stock ranking score. According to the stock ranking score, the stock selection strategy choose top M stocks to build up a stock portfolio.

3.1 Data Preprocessing

Historic price data for stocks are gathered in the format of time series of candles (the open price, close price, highest price, lowest price, and trading volume) in a granularity of one trading day. In [18], stock prices were proved to be logarithmic normal distribution, we convert the data to a normal distribution by *log* transformation. As for non-stationary, we employ the first-order difference method to deal with. These transformation can be expressed as:

$$log(p_t) - log(p_{t-1}) \tag{1}$$

One stock in one trading day constitutes a sample point. At date t, we look back at the historical data for N trading days, from $t - N + 1$ to t.

3.2 LSTM Network

Recurrent neural networks are inspired by the cyclical connectivity of neurons in brain, which introduce iterative function loops to store information [11]. One of the difference between a multilayered perceptron (MLP) and an RNN is that an MLP maps inputs to output vectors directly, where as an RNN can map whole previous inputs to each output. In other words, the RNNs allow a "memory" of previous inputs which stay in the networks and have effect on the outputs.

The main difference between LSTM cell and simple RNN cell lies in that it adds a "processor" to judge whether the information is useful or not. The structure of this processor is called "gate". A cell was placed three gates, respectively, called the input gate, forget gate and output gate. A message enters the LSTM's network and can be used to determine if it is useful. Only valid information will be left behind, and useless information will be forgotten through forget gate. The gates and cell are defined by the following equations:

$$i_t = \sigma(W_{wi}x_t + W_{hi}h_{t-1}), \tag{2}$$

$$f_t = \sigma(W_{wf}x_t + W_{hf}h_{t-1}), \tag{3}$$

$$o_t = \sigma(W_{wo}x_t + W_{ho}h_{t-1}), \tag{4}$$

$$\hat{c}_t = tanh(W_{wc}x_t + W_{hc}h_{t-1}), \tag{5}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \hat{c}_t, h_t = o_t \otimes tanh(c_t), \tag{6}$$

The preprocessed historical N trading day sequence data will be fed to the LSTM model and its timing information will be modeled and characterized to deep representation.

3.3 Rank Score Evaluation

We employ a simple one-layer and one-output-unit feedforward neural network to calculate the stock's ranking score according to the deep representation. *tanh* activation function is used and the ranking score is between -1 and 1. It is defined as follows:

$$Ranking_{t,i} = tanh(w_{l,1}d_{t,i} + b) \tag{7}$$

where $Ranking_{t,i}$ is the rank score of stock i at date t, $d_{t,i}$ is the deep representation generated by LSTM model, l is the dimension of $d_{t,i}$.

We expect that the stocks with higher rank scores will get higher returns in the trading days after. However, high rank score at time t can not guarantee that the future stock returns are absolutely positive, it can only be used to show that these high-scoring stocks perform better than the low-scoring stocks in the trading days after time t. Here, we hope this indicator can play a role in the next 10 trading days.

3.4 Stock Selection Strategy

With stock rank score in hand, we need to construct a stock portfolio. It means that which stocks will be invested and how much percent of total money will be paid on the chosen stocks should be decided. Portfolio management is another hot task which has been discussed and studied by many researchers, especially financial economists [12]. In this paper, we employ two simple stock selection strategy, listed as follows:

Equally Weighted. First, at time t, we sort the stocks according to the rank score, and select the top M stocks, total money is divided equally into M parts and is invested in each chosen stock separately. After holding T trading days, we use new generated historical price data to calculate new stock rank scores at time $t + T$, select the new top M stocks, sell the old holding stocks and invest the newly chosen stocks.

Score Weighted. We invest all stocks at time t with the weights calculated by the rank score. It is defined as follows:

$$\overline{Ranking}_{t,i} = Ranking_{t,i} - \min_i Ranking_{t,i} \tag{8}$$

$$weights_{t,i} = \frac{\overline{Ranking}_{t,i}}{\sum_i \overline{Ranking}_{t,i}} \tag{9}$$

Same as equally weighted method, the weights will be recalculated and the money will be reinvested every T days.

4 Experiments

The experiment is designed to answer the following questions. (1) Whether the proposed method is able to extract related information from the raw data without handcrafted features? (2) How effective is the *batch_t* training method? (3) What is the difference between the performance of two kinds of loss functions? (4) Can *DeepStockRanker* compete with other advanced stock prediction model for stock ranking?

4.1 Datasets and Evaluation Metrics

In this paper, we use the data of all the listed stocks in China's A-share market from January 1, 2006 to December 31, 2017, including Shanghai A-shares and Shenzhen A-shares. However the price movements of newly listed stocks are more unusual, significantly different from other stocks, we remove the newly listed 3-month stock from the total dataset. The total dataset is split into three parts by time, training set (2006-01-01 to 2013-12-31), verification set (2014-01-01 to 2014-12-31) and test set (2015-01-01 to 2017-12-31). The summary statistics in different sets are detailedly presented in Table 1, where *Dates* means number of the trading date among the period, $Total$ means the total number of samples in the set, *Daily* means the average daily number of samples.

To validate and evaluate the effectiveness of the proposed *DeepStockRanker*, some evaluation metrics which are widely used in quantitative investment are employed in this paper.

Information Coeffiencient *(IC)*. It is used as a performance metric for the predictive skill of a ranking prediction model. It is similar to correlation in that it

Table 1. Dataset statistics

Sets	Periods	Dates	Total	Daily
Training	20060101-20131231	1940	3585361	1848.1
Verification	20140101-20141231	245	615464	2512.1
Test	20150101-20171231	732	2131568	2912.0

can be seen to measure the linear relationship between two random variables, e.g. predicted stock returns and the actualized returns, represents model's predictive ability. It is defined as follows:

$$IC = corr(Ranking_{t,i}, RET_{t,T,i}) \tag{10}$$

Active Return *(AR)*. It refers to that segment of the returns in an investment portfolio that is due to active management decisions made by the portfolio manager. In this paper, the portfolio manager is *DeepStockRanker*. The active return is calculated as the return of the portfolio minus some benchmark return:

$$AR = R_p - R_b \tag{11}$$

where R_p is the portfolio return constructed by the ranking model, and R_b is the benchmark return which in this paper is the market average return calculated by averaging all the stocks' return.

Information Ratio *(IR)*. It is a measure of the risk-adjusted return of a portfolio. It is defined as expected active return divided by tracking error, where tracking error is the standard deviation of the active return:

$$IR = \frac{E[R_p - R_b]}{\sigma} = \frac{E[R_p - R_b]}{\sqrt{var[R_p - R_b]}} \tag{12}$$

The information ratio is often used to gauge the skill of managers of funds. In this case, it measures the active return of the manager's portfolio divided by the amount of risk that the manager takes relative to the benchmark. The higher the information ratio, the higher the active return of the portfolio, given the amount of risk taken, and the better the manager.

In this paper, AR and IR are both annualized.

4.2 Experimental Setting

For the first half of the model except stock selection strategy and stock portfolio, the number of units in the LSTM cell is 256 and the state at final step is fed to a fully connected layer also with 256 units for the stock rank score prediction. All these are optimized end-to-end using the ADAM [17] optimizer, and gradient clip is set to be 5. As for equally weighted stock selection strategy, M is set to 100, which means the portfolio holds 100 stocks simultaneously. The hyperparameters of *DeepStockRanker* are chosen without careful design.

4.3 Model Performance Comparison

In order to show the quality of no handcrafted feature extraction method, we compare our model with other several advanced methods using handcrafted feature extraction. We employ 11 technical indicators as handcrafted features which are described in [20] and widely used in quantitive investment.

SVR. A version of SVM for regression, proposed in 1996. SVR+RAW denotes that we input raw trading data into SVR to predict future stock ranking. SVR+TI denotes that we input 11 technical indicators into SVR to predict future stock ranking.

RBM. Refering to [20], RBM based stock prediction model is implemented by ourself. Before the 11 technical indicators are fed to RBM, they will be transformed to discrete ones by TDDPL according to [24].

MLP. Different from *DeepStockRanker*, We replace LSTM with sample FNN and other parts of the model remain the same.

LSTM+TI. Different from *DeepStockRanker*, We use 11 technical indicators as input and other parts of the model remain the same.

Table 2. Performance of each method.

Model	IC	AR	IR
SVR+RAW	0.0312	0.036	0.5262
SVR+TI	0.0357	0.0357	0.5543
RBM	0.0874	0.1021	1.5460
MLP	0.1025	0.1226	1.8488
LSTM+TI	0.0892	0.1132	1.7012
DeepStockRanker+sw	**0.1259**	0.1752	**3.0521**
DeepStockRanker+ew	**0.1259**	**0.2015**	2.8698

Table 2 shows the experimental results among different models. DeepStockRanker+sw denotes score weighted stock selection strategy and DeepStockRanker+ew denotes equally weighted stock selection strategy. Experimental results indicate that *DeepStockRanker* has much better performance than other methods. The net value of portfolio constructed by *DeepStockRanker* in test set are shown in Fig. 1. DeepStockRanker+sw has higher IR but lower AR than DeepStockRanker+eq. This is because DeepStockRanker+sw holds all stocks and DeepStockRanker+ew only holds top 100 stocks, DeepStockRanker+ew

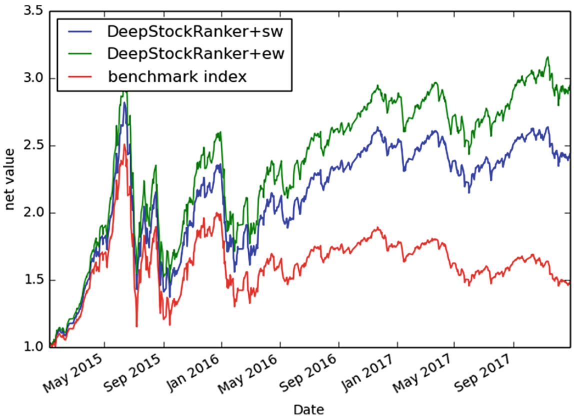

Fig. 1. Net value of portfolios in test set.

is more aggressive with higher return and high tracking error. Interestingly, LSTM+TI perform such poor than *DeepStockRanker* only because the input change to handcrafted features.

The intuitive explanation can be given in the following. Handcrafted feature extraction is limited to people's prior knowledge, it is hard to be searched or optimized. In most case, the use of handcrafted feature extraction is not optimal for the problem to be solved because of the limitations of prior knowledge. Training a model from raw data to loss function (end-to-end) can make the model learn to collect relevant and valid information on specific issues. Since *DeepStockRanker* is proposed, the noisy financial trading data can be learned by neural network using end-to-end training, naturally, this approach can achieve the best results.

5 Conclusion

A simple but efficient method is proposed to predict future stock return ranking without handcrafted features. In contrast with other several advanced methods, We demonstrate that our approach significantly outperforms state-of-the-art techniques. A new loss function is designed for noisy financial sequence data and stock rank task to train the neural networks by end-to-end.

As to the future research, we intend to employ more methods of deep learning, especially in NLP, such as attention mechanism to model the investors' attention of the stock market.

Acknowledgment. This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61673025 and 61375119 and Supported by Beijing Natural Science Foundation (4162029), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China under grant no. 2015CB352302.

References

1. Ariyo, A.A., Adewumi, A.O., Ayo, C.K.: Stock price prediction using the arima model. In: 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation (UKSim), pp. 106–112. IEEE (2014)
2. Azevedo, J.M., Almeida, R., Almeida, P.: Using data mining with time series data in short-term stocks prediction. Int. J. Intell. Sci. **2**(4A), 177–181 (2012)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
4. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
5. Bisoi, R., Dash, P.K.: A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented kalman filter. Appl. Soft Comput. **19**, 41–56 (2014)
6. Boyacioglu, M.A., Avci, D.: An adaptive network-based fuzzy inference system (anfis) for the prediction of stock market return: the case of the istanbul stock exchange. Expert Syst. Appl. **37**(12), 7908–7912 (2010)
7. Chen, Z., Du, X.: Study of stock prediction based on social network. In: 2013 International Conference on Social Computing (SocialCom), pp. 913–916. IEEE (2013)
8. Chou, R.Y.: Volatility persistence and stock valuations: some empirical evidence using garch. J. Appl. Econ. **3**(4), 279–294 (1988)
9. Ding, X., Zhang, Y., Liu, T., Duan, J.: Using structured events to predict stock price movement: an empirical investigation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1415–1425 (2014)
10. Dong, G., Fataliyev, K., Wang, L.: One-step and multi-step ahead stock prediction using backpropagation neural networks. In: 2013 9th International Conference on Information, Communications and Signal Processing (ICICS), pp. 1–5. IEEE (2013)
11. Graves, A., et al.: Supervised Sequence Labelling with Recurrent Neural Networks, vol. 385. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24797-2
12. Grinold, R.C., Kahn, R.N.: Active portfolio management (2000)
13. Guresen, E., Kayakutlu, G., Daim, T.U.: Using artificial neural network models in stock market index prediction. Expert Syst. Appl. **38**(8), 10389–10397 (2011)
14. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Sig. Process. Mag. **29**(6), 82–97 (2012)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
16. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. arXiv preprint arXiv:1412.2306 (2014)
17. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: The International Conference on Learning Representations (ICLR), San Diego, USA (2015)
18. Kon, S.J.: Models of stock returns a comparison. J. Financ. **39**(1), 147–165 (1984)
19. Li, X., Xie, H., Song, Y., Zhu, S., Li, Q., Wang, F.L.: Does summarization help stock prediction? a news impact analysis. IEEE Intell. Syst. **30**(3), 26–34 (2015)
20. Liang, Q., Rong, W., Zhang, J., Liu, J., Xiong, Z.: Restricted boltzmann machine based stock market trend prediction. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 1380–1387. IEEE (2017)

21. Makrehchi, M., Shah, S., Liao, W.: Stock prediction using event-based sentiment analysis. In: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 1, pp. 337–342. IEEE (2013)
22. Nelson, D.M., Pereira, A.C., de Oliveira, R.A.: Stock market's price movement prediction with LSTM neural networks. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 1419–1426. IEEE (2017)
23. Park, K., Shin, H.: Stock price prediction based on a complex interrelation network of economic factors. Eng. Appl. Artif. Intell. **26**(5–6), 1550–1561 (2013)
24. Patel, J., Shah, S., Thakkar, P., Kotecha, K.: Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Syst. Appl. **42**(1), 259–268 (2015)
25. Quan, Z.Y.: Stock prediction by searching similar candlestick charts. In: 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW), pp. 322–325. IEEE (2013)
26. Schumaker, R.P., Chen, H.: A quantitative stock prediction system based on financial news. Inf. Process. Manag. **45**(5), 571–583 (2009)
27. Skuza, M., Romanowski, A.: Sentiment analysis of twitter data within big data distributed environment for stock prediction. In: 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1349–1354. IEEE (2015)
28. Tetlock, P.C., Saar-Tsechansky, M., Macskassy, S.: More than words: quantifying language to measure firms' fundamentals. J. Financ. **63**(3), 1437–1467 (2008)
29. Tsai, C.F., Quan, Z.Y.: Stock prediction by searching for similarities in candlestick charts. ACM Trans. Manag. Inf. Syst. (TMIS) **5**(2), 9 (2014)
30. Wu, D., Fung, G.P.C., Yu, J.X., Pan, Q.: Stock prediction: an event-driven approach based on bursty keywords. Front. Comput. Sci. China **3**(2), 145–157 (2009)
31. Xie, B., Passonneau, R.J., Wu, L., Creamer, G.G.: Semantic frames to predict stock price movement (2013)