Index Trend Predication

王嘉佳 **1801212935**

周雨田 **1801212996**

许煜川 **1801212958**

Mariia Semenenko 1902010565

2020/1/17

1 Introduction and Motivation

In this project, we try to predict the trend of indexes by machine learning. Firstly, we use MySQL to store and process data and use Lasso regression to choose features. Second,ly we use Random Forest and LSTM to forecast whether the price will go up or down or stable in next one minute. Also, we use K–means to choose indexes and do pair trading between indexes.

Three main motivations drive us to choose this topic. Firstly, we want to use neural network combined with big data analysis to obtain a model that can better predict the stock index trend and improve the return of trading strategies. Secondly, based on our literature review, we find that LSTM, random forest and some other models have impressive performance and are used by some famous global quantitative funds. Thirdly, we want to use the knowledge in this class to optimize the model and process the data

2 Literature Review

2.1 Random Forest

Random forest uses bootstrap resampling technology to generate a new training sample set, and generate classification trees to form a random forest. Randomness comes from features and samples choosing. Forest means the forest of decision trees.

It has ths following advantages: 1) Bootstrap and feature selection reduce the overfitting problem; 2) Good anti noise ability; 3) High accuracy; 4) Can deal with both discrete data and continuous data without normalization; 5) Easy to implement and fast to train; 6) Not sensitive to parameters.

It has ths following disadvantages: 1) Random forest likes a black box, and it is hard to control the internal operation of the model. 2) Imprecise when dealing with features with multiple value divisions.

Motivation behind Random Forest for Index Timing

- Random forest is mainly used in regression and classification. Random forest is very suitable for stock or stock timing as a classifier
- The noise of financial data is large, and the randomness of random forest can reduce over fitting and the impact of noise
- Random forest is an integrated learning. Decision trees are independent of each other and make the final classification decision by voting and the accuracy is improved

2.2 LSTM

Long short-term memory (LSTM) is a special RNN, which is mainly used to solve the problem of gradient disappearance and gradient explosion. Compared with ordinary RNN, LSTM can perform better in longer sequences. There are three main phases in LSTM: 1) Forget. Selectively forget the input from the previous node; 2) Select. Selectively "remembers" the input of this stage; 3) Output. Determine which outputs will be treated as the current state.

Motivation for using LSTM： Traditionally, time series forecasting has been dominated by linear methods because they are well understood and effective on many simpler forecasting problems. But Long Short-Term memory is one of the most successful RNNs architectures, that is why we decided to apply it to our big data problem as one of the models. LSTM in contrast to usual RNN introduces the memory cell, a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. With these memory cells, networks can effectively associate memories and input remote in time. Hence, LSTM suits to grasp the structure of data dynamically over time and gives high prediction capacity.

2.3 Lasso regression

Lasso regression is very similar to ridge regression, the difference is that they use different regularization terms. It prevents the effect of overfitting by constraining parameters. Lasso can shrink

the parameters of some features with relatively small effects to 0 to obtain sparse solutions. That is to say, with this method, the purpose of dimensionality reduction (feature screening) is achieved in the process of training the model.

The cost function of Lasso regression:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(y^{(i)} - (wx^{(i)} + b))^2 + \lambda||w||_1 = \frac{1}{2}MSE(\theta) + \lambda\sum_{i=1}^{n}|\theta_i|$$

2.4 K-means

The K-Means algorithm is an unsupervised classification algorithm, assuming unlabeled data sets:

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \end{bmatrix}$$

The task of this algorithm is to cluster the data set into C = {C1, C2, ..., Ck}. The minimization loss function is:

$$E = \sum_{i=1}^{k}\sum_{x\in C_i}||x - \mu_i||^2$$

μ is the central point of Ci:

$$\mu_i = \frac{1}{|C_i|}\sum_{x\in Ci} x$$

3 Model

3.1 Data Processing and Database

We use MySQL for these reasons: 1) Our data is structured so we can use a relational database; 2) The database is mainly used for storage, and the amount of data calculation is not large, so there is no need to choose a specific time series database; 3) Simple operation and convenient query; 4) Faster than excel. And we process by the following way: 1) connect data from different dates; 2) Remove data outside transaction time and deduplicate; 3) To ensure the consistency of different index data, we take the intersection of trading dates and times between different indices.

3.2 Model Assumptions

- Indexes can be traded;
- Trading can be executed immediately;
- Traders can make decision every three seconds;
- There is enough principal to start the transaction;
- If we consider cost, the cost is 1bp of trading value and no other cost;
- Whether indexes can be short or not can be determined by us.

3.3 Modelling details

- Choosing 3s as the interval of input features;
- Predicting 1min trend (price up or down or stable) of price every 3s;
- Train set: before 26/03/2019, 55 days; Test set: 27,28,29/03/2019, 3 days;
- Input features: low, open, close, low/open, high/open, high, lastprice, volume, log3s, log6s, log9s, log15s, log30s, log60s, log90s, log120s, (high-last price)/open, (last price-low)/open, (open-close)/close;

- Input y:

$$\text{Price Up(1): } \frac{\text{Return}}{\text{max\_return in train or test set}} > 0.005$$

$$\text{Price Stable(0): } -0.005 < \frac{\text{Return}}{\text{max\_return in train or test set}} < 0.005$$

$$\text{Price Down(-1): } \frac{\text{Return}}{\text{max\_return in train or test set}} < -0.005$$

3.4 LSTM

Neural network: one LSTM layer and three dense layers.

LSTM layer: Output dimension: 128 dimensions, Drop out weight = 0.1 Activation function = atan, Kernel regularizer = l2(0.1), Activity regularizer = l1(0.000001)

Dense layer: 1) Output dimension: 64 dimensions, Activation function = linear;

2) Output dimension: 16 dimensions, Activation function = linear;

3) Last dense layer: With continuous output: 1 dimension, Activation function = softmax. Loss = mse, Optimizer = adam. With discrete output: 3 dimensions, Activation function. = softmax. Loss = categorical crossentropy, Optimizer = sgd. With continuous output LSTM model, the input y is one dimension. And with discrete. output, the input y is three dimension which is 1 is [1 0 0], 0 is [0 1 0], -1 is [0 0 1].

Result: Taking index 399001 as an example: with continuous output, assume that if. output >. 0.005, y = 1, buy the index, if output < -0.005 y = -1, then the accuracy on test set is 0.35. With discrete output, the accuracy on train set is about 0.4, and the accuracy on test set is about 0.5.

The prediction accuracy of LSTM model is not as good as of Random Forest model, so we focus on the Random Forest model.

3.5 Random Forest

Features: we use Lasso to choose features, delete features low, open, close, low/open, high/open and the remain features are high, last price, volume, log3s, log6s, log9s, log15s, log30s, log60s, log90s, log120s, (high-last price)/open, (last price-low)/open, (open-close)/close

Indexes:

Table 1 Indexes List

Number	Name	Number	Name
399001	深证成份指数	399235	建筑业指数
399004	深证 100 指数	399236	批发零售指数
399005	中小板指数 P	399237	运输仓储指数
399006	创业板指数 P	399239	信息技术指数
399007	深证 300 价格	399240	金融业指数
399008	中小板 300P	399241	房地产业指数
399231	农林牧渔指数	399242	商务服务指数
399232	采矿业指数	399243	科研服务指数
399233	制造业指数	399244	公共环保指数
399234	水电煤气指数	399248	文化传播指数

Investor risk preference [a, b] and max long c: When prediction is stable (0), we generate a random number between 0 and 1. If the random number is less than a, long the index. If the random number is larger than b, short the index. [a, b] is given by investors. The total long size should be less than c.

Strategy: Our strategy is combined of 20 single strategies. Single strategy makes decision every 60s.

Every minute should close the position one minute ago and make a new decision of weather trading at this moment. Multi- strategy combined of 20 single strategies which start from 3s, 6s …… 57s.

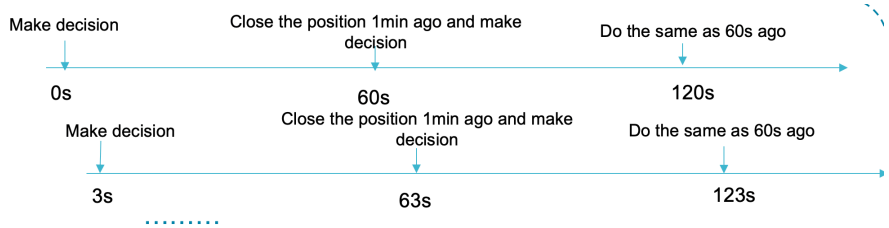

Figure 1. Strategy Explanation

Single strategies are combined together only by having one total long size limit and short size limit. The multi-strategies results are combined by time series.

Robust: The random forest model has robust itself because of the random choosing of samples and we also used the first three days or the last three days as the test set, which shows the similar accuracy results. The accuracy of the model is stable

Other situation detail: We also tested situations with cost with short, without cost with short, with cost without short, without cost without short. We calculate the cost as 1bp of price.

Results: All the accuracy on train set is higher than 95% and almost all the accuracy on test set is higher than 60%. The following figures shows the return of our strategies for index 399001 and 399233.“w”means with and “with” means without in the figures. Time means the number of time we make the decision. By the results, we can see we make profit in all situation for these two indexes. And for most indexes in the twenty indexes we can make profit.

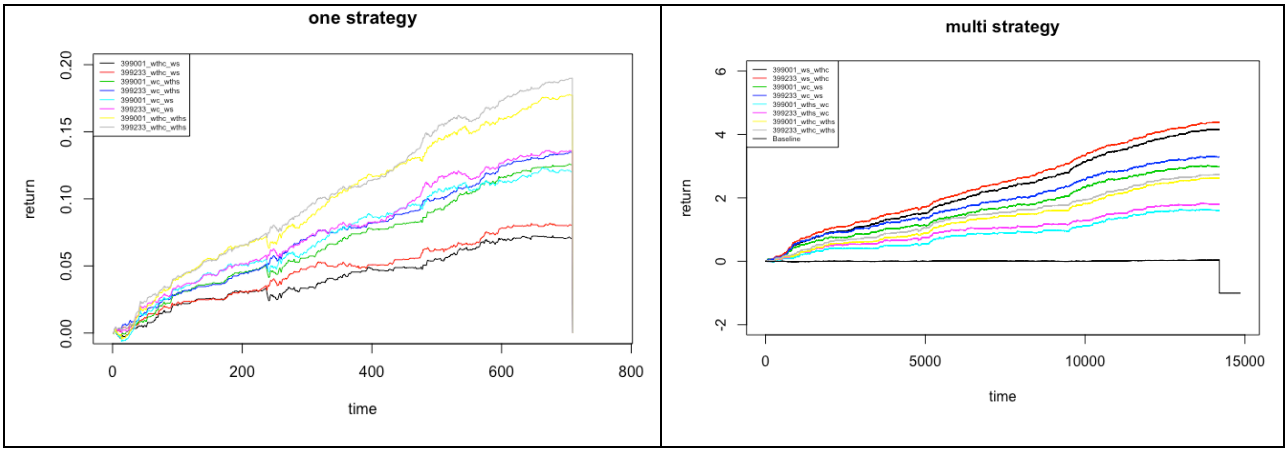

Figure 2. Return of Random Forest Model

4 Pair Trading

4.1 Trading Strategy

After forecasting the price of a single stock index separately, we want to make a pairing transaction between the two indices.

First we use K-Means to cluster the original price of each index, and set the number of classes to 2. According to the results of clustering, the first three indices are in the first cluster and the other indices are in the second cluster. Select two stock indexes with the highest prediction accuracy from two clusters with the codes 399001 and 399233, respectively.

If we can find the time when the price changes of the two stock indexes are reversed and make corresponding transactions, then we will seize this arbitrage opportunity. For example, if the price of

399001 will increase in the next minute and the price of 399233 will decrease in the next minute, then we will make a trading strategy of buying 399001 and selling 399233, so that we can arbitrage.

We have set up four different pair trading trading strategies. According to the strictness of the trading restrictions, they are as follows:

For three consecutive seconds, the predicted price changes in the opposite direction, that is, the price of one index rises for three consecutive seconds and the price of another index falls for three consecutive seconds;

Predicted prices do not change synchronously for three consecutive seconds, that is, one index price rises or remains unchanged for three consecutive seconds while another index price falls or remains unchanged for three consecutive seconds;

There is a situation where the predicted price changes in the opposite direction, that is, the price of one index increases while the price of another index decreases;

There is a situation where the predicted price changes in the opposite direction, that is, the price of one index rises or remains unchanged while the price of another index falls or remains unchanged.

4.2 Results and Conclusion

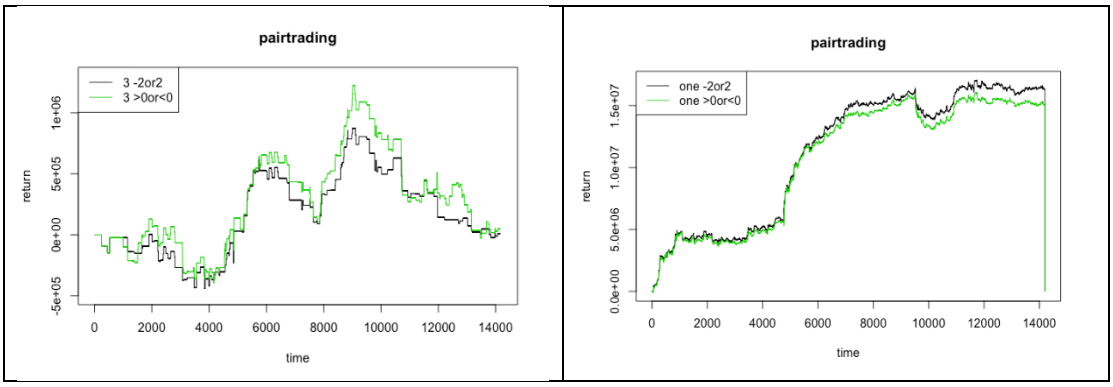

Figure 3. Return of Pair Trading

The income from the above figure shows that under the assumption of no transaction fees, the returns of the latter two trading strategies increase more steadily, and the returns of the former two strategies are more volatile.

The result of the third trading strategy is better than the fourth one. This is because when we relax the trading restrictions until the price changes, we trade. The larger the arbitrage space, the greater the profit.

The result of the second trading strategy is better than the first. This may be because the conditions for the price to reverse in three consecutive seconds are harsh enough, so when there is an arbitrage opportunity, you can make a profit.

We also investigated the reasons why the time-consuming strategy of trading is not profitable. Because in the previous single prediction, our prediction accuracy was about 60%. When we coarsely calculated and trusted this result to arbitrage the two indices, the prediction accuracy would be reduced to 36%, so the effect was not good.

Reference

[1] LEO BREIMAN, Random Forests, Statistics Department, University of California, Berkeley, CA 94720, Machine Learning, 45, 5–32, 2001

[2] Amit, Y., Blanchard, G., & Wilder, K. (1999). Multiple randomized classifiers: MRCL Technical Report, Department of Statistics, University of Chicago.

[3] Breiman, L. (1996a). Bagging predictors. Machine Learning 26(2), 123–140.

[4] Breiman, L. (1996b). Out-of-bag estimation, ftp.stat.berkeley.edu/pub/users/breiman/OOBestimation.ps

[5] Breiman, L. (1998a). Arcing classifiers (discussion paper). Annals of Statistics, 26, 801–824.

[6] Breiman. L. (1998b). Randomizing outputs to increase prediction accuracy. Technical Report 518, May 1, 1998,

[7] Statistics Department, UCB (in press, Machine Learning).

[8] Breiman, L. 1999. Using adaptive bagging to debias regressions. Technical Report 547, Statistics Dept. UCB.

[9] Breiman, L. 2000. Some infinity theory for predictor ensembles. Technical Report 579, Statistics Dept. UCB.

[10] Amit, Y. & Geman, D. (1997). Shape quantization and recognition with randomized trees. Neural Computation, 9, 1545–1588.

[11] Ho, Tin Kam (1995). Random Decision Forests

[12] Ho TK (1998). "The Random Subspace Method for Constructing Decision Forests" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832–844.

[13] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2008). The Elements of Statistical Learning (2nd ed.). Springer. ISBN 0-387-95284-5.

[14] 李欣海. 随机森林模型在分类与回归分析中的应用[J]. 应用昆虫学报,2013,04:1190-1197.

[15] 董师师,黄哲学. 随机森林理论浅析[J]. 集成技术,2013,01:1-7.

[16] 王淑燕, 曹正凤, 陈铭芷. 随机森林在量化选股中的应用研究[J]. 运筹与管理, 2016, 25(3): 163–168, 177.

[17] 刘微, 罗林开, 王华珍. 基于随机森林的基金重仓股预测[J]. 福州大学学报(自然科学版), 2008, 36(S1): 134–139.

[18] 曹正凤, 纪宏, 谢邦昌. 使用随机森林算法实现优质股票的选择[J]. 首都经济贸易大学学报, 2014, 16(2): 21–27.

[19] 于晓雯, 张延良. 金砖国家股票市场成长性指标体系构建[J]. 中国证券期货, 2013(1): 63–64.

[20] 刘敏, 郎荣玲, 曹永斌. 随机森林中树的数量[J]. 计算机工程与应用, 2015, 51(5): 126–131.