Two-stage Stock Price Prediction and Arbitrage Trading Strategy

Zifeng Wang, Guoqing Zhang and Zhiyuan Chen
Tsinghua-Berkeley Shenzhen Institute (TBSI),
Tsinghua University, Shenzhen, China

Abstract

In the Market Analysis challenge, we are required to achieve two tasks: stock price prediction (task 1) and trading strategy building (task 2). In task 1, we propose a two-stage model, which consists of a day model and an intra-day model, to doing minute level prediction on stock prices. The day model, built via Seasonal ARIMA algorithm, being trained on each day opening price, gets 0.22 of MSE in test; the intra-day model, on the basis of LSTM, works on each minute price day by day and gets 0.2 in MSE. In task 2, a statistical arbitrage trading method is introduced. Here, we make use of a statistics term named cointegration. In practice, we develop a specific script for automatic strategy mining and with these mined rules obtain 10% return during the mock trading in one month.

I. Background

The whole project is a challenge of stock market analysis, offered by course named *Fundamentals of Applied Information Theory* taught by Prof. Lin Zhang in Tsinghua-Berkeley Shenzhen Institute (TBSI). In this project, students are required to achieve two tasks: the first is to doing stock price prediction and the second is to building a trading strategy whether based on the former result or not then performing it in mock trading via the given API.

In task one, the training data is a database consists of more than 500 stocks and their minute prices from Jan. 2008 to Nov. 2009. In fact, the data sets remain further processing because of existing special scenarios like suspension that the stock stops transaction thus there would be no change in price in several days even months. Besides, the testing data set has 500 stocks waited for forecast, which requires highly on the model's generality and robustness.

In task two, it supposes that each group has 100,000 RMB initially. We need to design an automatic portfolio strategy in order to maximize our profit with the setup that no transaction fee is considered here. This strategy, which has to takes stock selection, transaction volume and stop loss into account, will be evaluated in mock trading during Dec. 2009 by the amount of profit acquired at last.

Overall, this challenge is a highly simplified circumstance compared with the real market, for example, it assumes that only stock price is considered, regardless of the market factor, trading volume, transaction fee, price slippage, etc. This setup makes market analysis feasible but also limits the source of information, setting some other barriers in our task, which will be presented in the following chapters.

II. Task One : Stock Price Prediction

Before building models, let us take a glance over what data we have for modeling (Table I). In fact, the only information can be drawn is one price each minute, which leads to a natural idea of utilizing time-series based approaches for analysis. For time-series analysis, there are two main approaches: the traditional Econometric regression methods [1] e.g. AR, MR, ARMA; and the deep learning models like Recurrent neural networks (RNN), Gated Recurrent Unit (GRU), Long-Short Term Memory (LSTM) trained via back-propagation algorithm [2]. Here, we make use of both two to building a mixed two-stage model.

TABLE I
A Sample of Formatted Training Dataset

	ID	Stock Code	Opening Price	Last Closing Price	Highest Price	Lowest Price	Closing Price
2008-01-02 09:30:00	1	600000	53	52.8	53	52.99	52.99
2008-01-02 09:31:00	2	600000	53	52.99	53.1	52.9	52.9
2008-01-02 09:32:00	3	600000	52.99	52.9	53	52.87	52.95
2008-01-02 09:33:00	4	600000	52.87	52.95	52.87	52.8	52.8
2008-01-02 09:34:00	5	600000	52.7	52.8	52.75	52	52
2008-01-02 09:35:00	6	600000	52	52	52.01	52	52
2008-01-02 09:36:00	7	600000	52	52	52.01	52	52.01
2008-01-02 09:37:00	8	600000	52	52.01	52	51.9	51.9

This two-stage model consists of two components: **day model** and **intra-day model**. This structure derives from the fact that each day we have 241 minute samples thus training and predicting directly with minute frequency is relatively tough and is of deviation. The Day Model is used to learning and predicting on one sample each day while the Intra-day Model is designed for using one sample each minute. In test, it is a little bit like *interpolation* process shown in Figure 1, that is, at first we have day model predict on each day opening price, obtaining the general trend; then we fill the remaining 240 points day by day with intra-day model. There is a point here that either day model or intra-day model does **NOT** fit on price series Y_t instead fit on price differences $D_t = Y_{t+1} - Y_t$, which is able to improve model robustness and diminish error significantly in practice. For example, if $Y_0 = 1$ and predictions are $D_0 = 0.1$, $D_1 = -0.1$, $D_2 = 0.2$, the final result is cumulative sum of D plus Y_0: $P_1 = 1.1$, $P_2 = 1.0$,$P_3 = 1.2$.

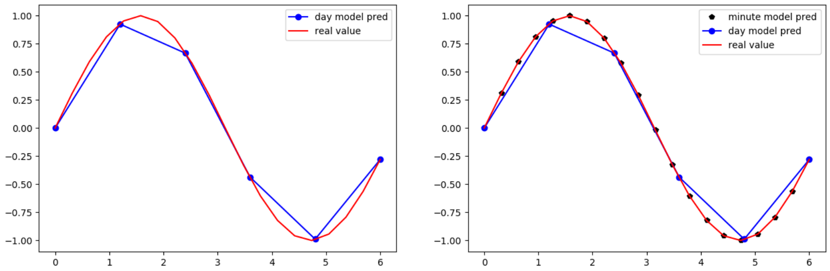

Fig. 1. Scheme of how two-stage model works

A. Day Model

The day model is built with the *Seasonal Autoregressive Integrated Moving Average* (SARIMA) [1]. An $ARMA(p, q)$ is given by:

$$X_t - \alpha_1 X_{t-1} - \cdots - \alpha_p X_{t-p} = \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} \tag{1}$$

or equivalently

$$1 - \sum_{i=1}^{p} \alpha_i L^i = (1 - \sum_{i=1}^{p-d})(1 - L)^d \tag{2}$$

where the L is named lag operator. Furthermore, an $ARIMA(p, d, q)$ with the d degree of difference is:

$$(1 - \sum_{i=1}^{p-d} \phi_i L^i)(1 - L)^d X_t = (1 + \sum_{i=1}^{q} \theta_i L^i)\epsilon_t \tag{3}$$

Hence, the SARIMA is ARIMA with seasonality. Here the seasonality of week, month, two month and quarter are introduced to forming a more robust model than the original ARIMA model. In experiment, the opening prices of stock 600000 from 2009/1/1 to 2009/7/1 are selected as training set then test on 2009/7. The results, with total 0.22 of mean-squared-error (MSE), are shown in Figure 2, and its seasonality components are shown in Figure II-B.

B. Intra-day Model

The Intra-day model has a LSTM-based structure, shown in Figure II-B. We train a intra-day model on 2009/6/1 to 2009/8/31 and then test it on 2009/9/1 to 2009/9/30. During test phase, it predicts one day price at first, then makes this day predictions new features for next day forecasting until completes whole test. It is worth mention that each day price series are the initial price obtained from day model plus cumulative sum of raw predictions from intra-day model. The test results are of 0.2 MSE, shown in Figure 3. However, this relatively good predictions are highly reliant on performances of the day model. These dependencies, sometimes, are vital for building final trading strategy.

III. Task Two : Trading via Statistical Arbitrage

In fact, there exist many problems in time-series prediction based trading strategy:

Firstly, the prediction highly depends on the last month modality, i.e. if last month is downward, the prediction is most probably downward of flat. However, the future trend is uncertain, influenced by many factors, far more than just the latest trend;

Secondly, there are more than 500 stocks with highly different characteristics, which means model trained on single or several stocks can not be promoted to other items;

Thirdly, stock price series in each month are not identical distributed. Hence the model that has learned representations of the past series can fail in prediction due to change of the series distributions.

Therefore, we decided not to build strategy via predictions from the prediction model built in task 1 but via method what is called *Statistical Arbitrage* [3].

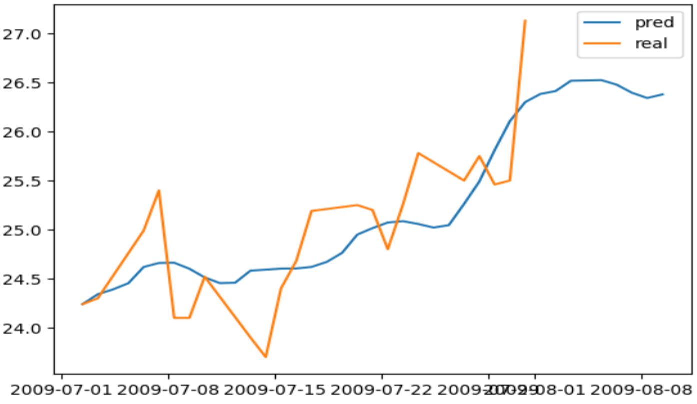

Fig. 2. Day Model: Predictions

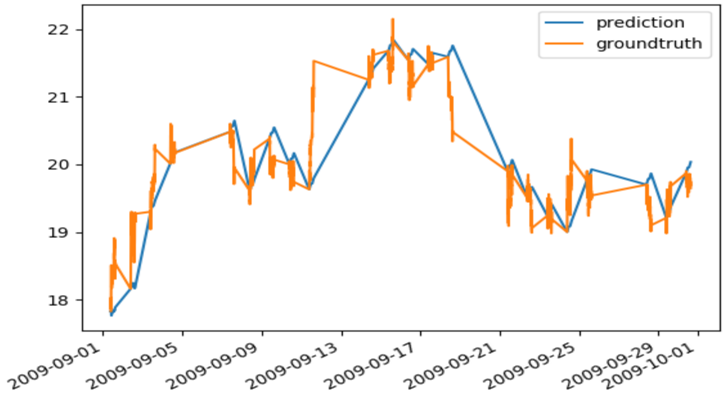

Fig. 3. Intra-day Model: Predictions

A. Arbitrage & Cointegration: Fundamentals

In finance, statistical arbitrage is a class of short-term financial trading strategies that employ mean reversion models involving broadly diversified portfolios of securities (hundreds to thousands) held for short periods of time (generally seconds to days). The *Cointegration* means relationship between two stocks S_1 and S_2, that is, their spread price $D = S_2 - S_2$ often fluctuates around a fixed value and with high probability within a two-sigma interval. This relationship was firstly provided statistical evidence by [4]. To find this interval, one simple method is by linear regression:

$$S_2 = w * S_1 + const \tag{4}$$

from which we get a spread price series $S_p = S_2 - w * S_1$. Then, the *Dickey–Fuller test* [5] can be implemented to testing whether the S_p is stationary and hence estimating significance in cointegration of the pair (S_1, S_2). The upper and lower bounds of this interval are computed by:

$$\begin{aligned} S_p^{upper} &= mean(S_p) + std(S_p) \\ S_p^{lower} &= mean(S_p) - std(S_p) \end{aligned} \tag{5}$$

B. Arbitrage Strategy: Technical Details

According the theory of cointegration, we can build a strategy which raises signals when the S_p reaches upper or lower bounds. The first challenge lies on finding enough significant pairs to constituting our portfolio pool. We design a scheme to mining out the pairs, estimating coefficents and building strategy. Suppose here is a stock S_1:

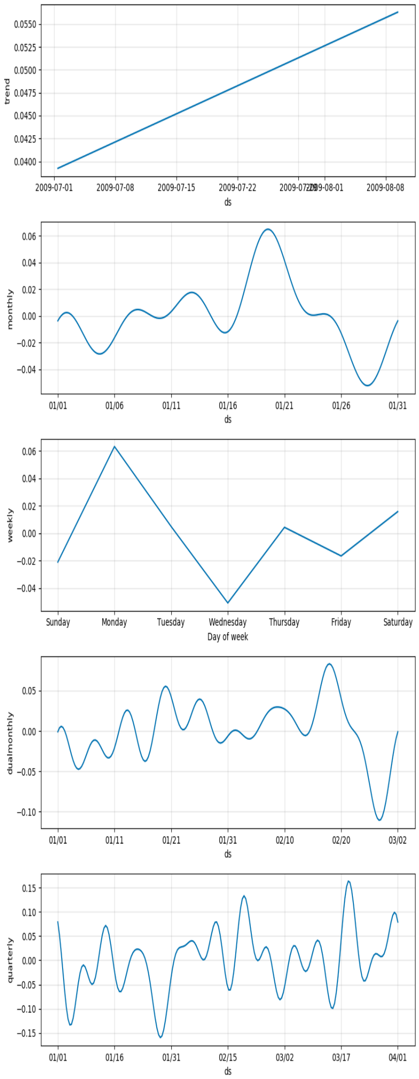

Fig. 4. Day Model: Seasonalities

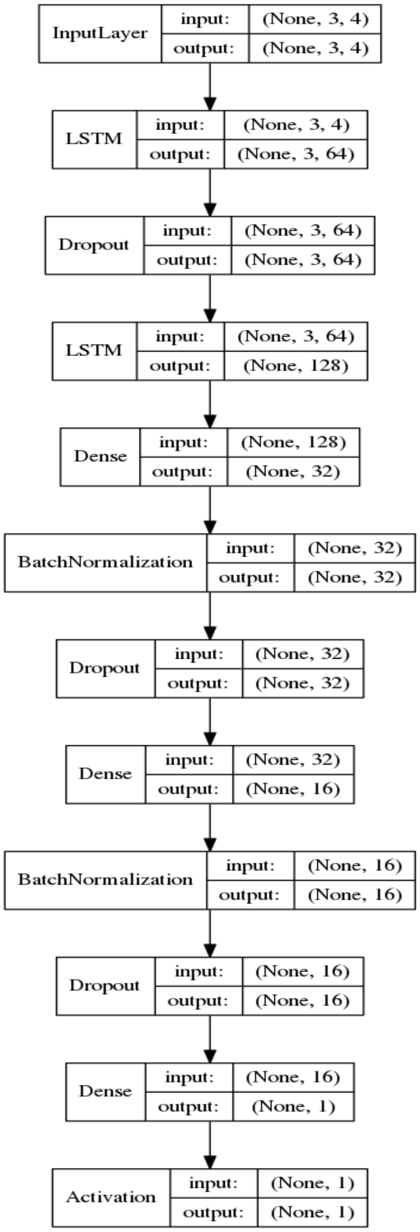

Fig. 5. Intraday Model: Structure

Step1: Finding $S_2, S_3, \ldots$ which have large correlation coefficient with S_1;
Step2: Finding cointegration relationship among $S_1, S_2, \ldots$, obtain significant pairs (S_i, S_j);
Step3: Doing linear regression on (S_i, S_j) via $S_j = w * S_i + const$, then compute bounds according to the formula (5).
An example of how the pairs information stored is shown in table II. When we have a significant pair $(600015, 600016)$, we can add a rule in trading strategy like the following:
Doing linear regression: $S_2 = -3.1 + 0.91 * S_1$ then $S_p = S_2 - 0.91 * S_1$;
Computing coefficients: $S_p^{upper} = mean(S_p) + std(S_p) = -2.46$ and $S_p^{lower} = mean(S_p) - std(S_p) = -3.73$;
Building strategy: If $S_p > -2.46$, buy S_1 and sell out S_2; if $S_p < -3.73$, buy S_2 and sell out S_1.
From the Figure 6, we can see that the S_p is in the interval during most of time except only little period of time, which proves the significance of cointegration.

C. Experiments

The process of mining these pairs, actually, can be done automatically via scripts, such that we build a script. As there are 500 stocks in database, traversing on all pairs requires experiment numbered $C_{500}^2 = 500 * 499/2 = 124750$. Due to the time

TABLE II
STORED PAIRS INFORMATION

S1	S2	p-value	down	up	w
600156	600566	1.50E-07	0.6	1.1	1.5
600103	600012	6.50E-06	1.3	1.5	1
600130	600242	1.50E-05	-2.2	-1.3	2.1

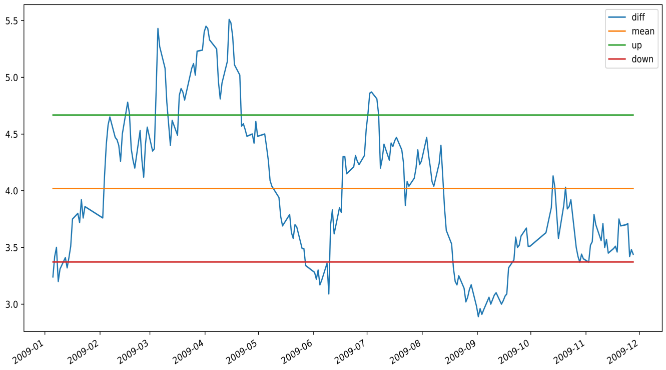

Fig. 6. This OLS spread series of pair $(600015, 600016)$ from 2009/1/1 to 2009/11/30

limits, we ran our scripts in 12 hours and get 137 pairs. In the next mock trading, we use top 10 significant pairs, which allows us get return around 10% profit in one month.

IV. CONCLUSION

In this project, we achieve time-series prediction in task 1 and build a arbitrage based strategy in task 2. After experiments in prediction, we recognize that the time-series prediction is pretty tough as all our method has an assumption that the future is a repeat of history, such that the model learns the series distribution based on history data. However, stock price is not predictable since it is influenced by numerous participants, the people who are involved in the game. Besides, even though we get 0.22 MSE on day model and 0.2 MSE on intra-day model separately, to assuring the sufficient tuning, the model is often conducted in single stock but there are 500 stocks wait for test, which leads great challenges.

On the other side, a stock itself represents a substantial company, which has a relatively stable states and these stable features can be dug out through many different statistical methods. In our task 2 solutions, we make use of statistics called cointegration to building our strategy. After 12 hours mining, we acquire 137 pairs and select top 10 significant ones, doing mock trading in 2009/12, the final return is around 10%, which proves efficiency of our method in practice.

ACKNOWLEDGMENTS

The authors would like to acknowledge the TA Fei Ma 's efforts in API construction and server maintenance during the period of ranking in mock trading. The brilliant courses during this semester and insights in advice after our presentation, given by Prof. Lin Zhang, are also genuinely appreciated.

REFERENCES

[1] F. Hayashi, *Econometrics*. Princeton Universitry Press, 2000, no. ISBN 0-691-01018-8.
[2] R. David, H. Geoffrey, and R. Williams, “Learning representations by back-propagating errors,” *Parallel Distributed Processing*, 1986.
[3] B. J. Jacobsen, *Statistical Arbitrage*. Handbook of Finance, 2008.
[4] C. Plosser, “Trends and random walks in macroeconomic time series: Some evidence and implications,” *Journal of Monetary Economics*, vol. 10, no. 2, pp. 139–162, 1982.
[5] D. A. Dickey and W. A. Fuller, “Distribution of the estimators for autoregressive time series with a unit root,” *Journal of the American Statistical Association*, vol. 74, no. 366, pp. 427–431, 1979.