Final Report —— Stock Price Prediction based on SVM

Yongchi Zhang, Guowei Zhang

YongchiZhang2017@u.northwestern.edu

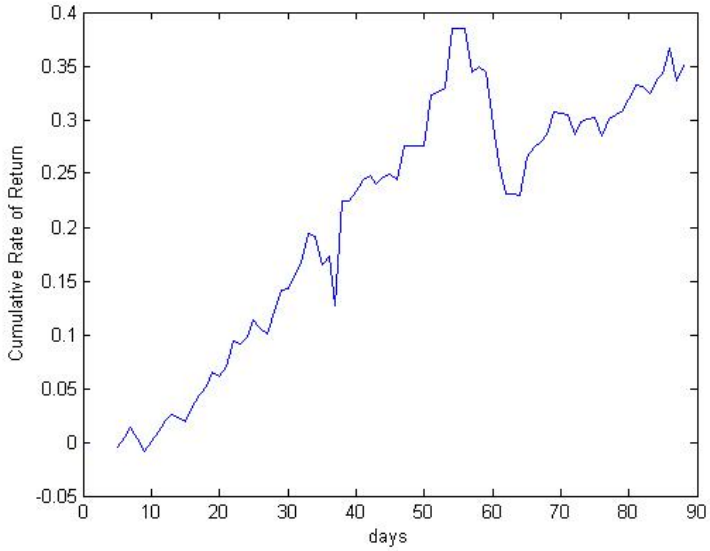
EECS 349 Machine Learning, Northwestern University

Summary

Our project is stock price prediction based on SVM. The task mainly is to forecast the future stock price based on the history data. We want to do this task mainly because both of us had the experience on investing on China stock market. We want to utilize the theory of machine learning to better help us invest on China stock market in the future. The stock is relevant to the country's economy directly, so if you can know some possible changes of stock market before others, you could have more chance to be successful on business.

In our project, we use SVM to forecast stock prices with history data for our input. The attributes we choose to describe the history data include opening price, closing price, highest price, lowest price, volume, price_change, p_change, ma5, ma10, ma20, v_ma5, v_ma10, v_ma20. Our algorithm will predict the stock closing price of next day and output the trading strategy for the next day: buying, selling or doing nothing.

Here is our final result of trading strategy, it performs a rate of return 35% in 90 days, with a success rate of 53.15%:



Report

First of all, we need to acquire our stock history data. Our data comes from Tushare, which is an open-source python stock analysis package. Tushare can only give us the data of past three years, so we get the data from June 1st, 2014 to June 1st, 2017. The entire dataset contains 730 tuples and each tuple has opening price, closing price, highest price, lowest price, volume, price_change, p_change, ma5, ma10, ma20, v_ma5, v_ma10, v_ma20. We think that the attributes we choose can precisely describe the stock data.

During our study, we use LIBSVM, which is a library for support vector machines. It is developed by Prof. Lin Chih-Jen of National Taiwan University. Here is the link:

http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Our algorithm performs SVM (Support Vector Machine) on stock price prediction. It mainly includes three functions from LIBSVM:

- function model = libsvmtrain(train_label, train_data, parameter)

  Function: Used to return the SVM model trained by the train data, train label and parameter.

- function [mse, bestc, bestg] = svmregress(train_label, train_data, cmin, cmax, gmin, gmax, v, cstep, gstep, msestep)

  Function: Used to return the best c, g given the input train_label and train_data, and the result of best c, g will be the input of libsvmtrain as the parameter.

- function predict = libsvmpredict(test_label, test_data, model)

  Function: Used to predict the label according to the test data and the trained SVM model. If you have the test label input, it can compare with that result and tell you the MSE, etc.

In our project, we take the closing price of next day as our label to predict, and opening price, closing price, highest price, lowest price, volume, price_change, p_change, ma5, ma10, ma20, v_ma5, v_ma10, v_ma20 of today as our train data. We get the result as follows after we run the SVM algorithm:

optimization finished, #iter = 1501

nu = 0.432491

obj = -3.873112, rho = -1.511400

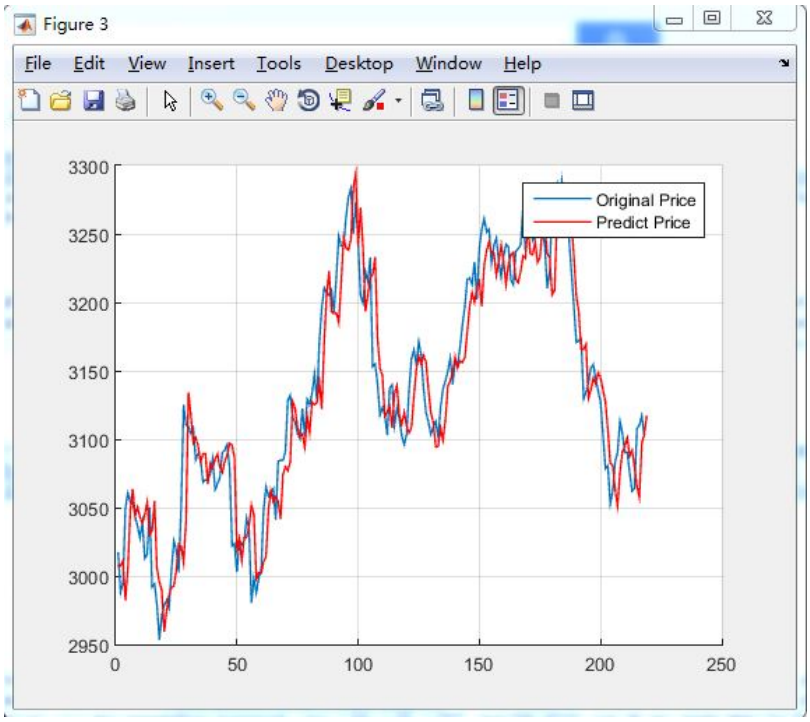
nSV = 257, nBSV = 194

Mean squared error = 4.70029e-05 (regression)

Squared correlation coefficient = 0.936346 (regression)

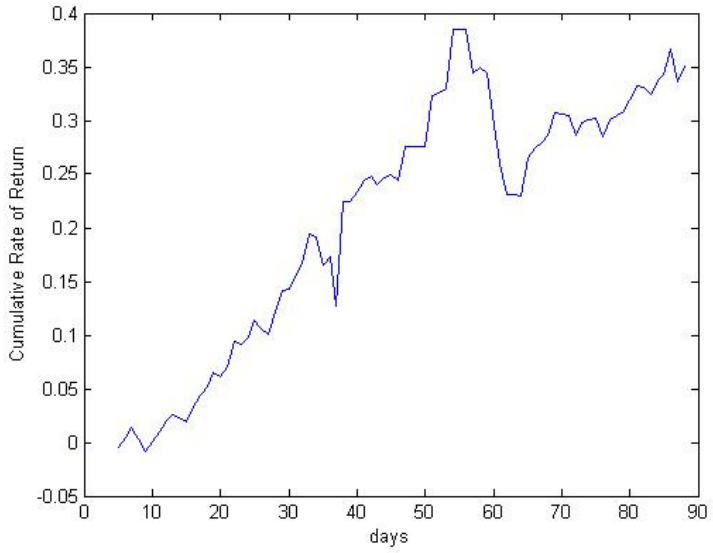
MSE = 4 R = 900%

The predict price and the original price are as follows:



According to the result of prediction, we set up the strategy: if the predict price of (next day) is higher than today's closing price over 0.5%, then we choose to buy in with the price of today's closing price; if the predict price of (next day) is lower than today's closing price over 0.5%, then we choose to sell out with the price of today's closing price; else we choose to do nothing.

We plot the Cumulative Rate of Return of our trading strategy as above. As we can see our trading strategy performs a rate of return 35% in 90 days, with a success rate of 53.15%.



Conclusion

In conclusion, we get a good prediction on stock prices and come up with a trading strategy. From this project, we learn how to use the open-source package libsvm to get a precise SVM model. Moreover, we have a better understanding of the financial stock market.

However, there are still a few points we can improve in our project. Now our trading cycle is one day and we use day K-line. Actually we can make the cycle larger or smaller, like we can use 60-minute K-line or week K-line. That will have a different result. What's more, we can add other statistical indicators like MACD, KDJ, RSI, etc.