Stock forecasting algorithms and their usage

Marc Misoch
Institute of Computer Science
University of Applied Science
Mannheim, Germany
Email: marc.misoch@gmail.com

David Marquant
Institute of Computer Science
University of Applied Science
Mannheim, Germany
Email: david.ma@live.com

***Abstract*—In this paper, we propose several methods to predict future stock prices. There are a lot of algorithms out there but some are more reliable than others. As expected, in most cases the general rule: *”the more complex, the better the prediction”* is true. However there are some relatively easy algorithms than can predict the stock prices of the future quite well. We are going to present and analyse some of these algorithms and take a deeper look into them. Just to make this clear, there is not a single algorithm, than can predict the future of a stock 100% correctly. The relationships of the global stock market are to complex to bring them down in a single computer program. We will show the experimental results we got with several algorithms and compare them. Also we will take a short look into investment strategies based on these forecasts and take a look at the results.**

***Keywords*—stock, forecast, prediction, paper, algorithms, neuronal-network.**

I. Introduction

A. Introduction to stock forecasting

There is a huge variety of methods that are used in forecasting stock prices. Many of the algorithms that have been made up just use structured data like tables with historic stock quotes in it. The aim of all these algorithms is to maximize the profit while keeping the risk as low as possible.

Since the stock market is global, there are many relationships between stocks. Some of them are paired with others so if the price of one stock goes down, the other stock will follow this trend. Stocks are also influenced by numerous factors and these factors also have relationships between each other. This leads us to the thought, that a good stock prediction has to consider more than just past stock prices. However it is still possible to make reasonably good predictions based on just previous prices.

One very important thing that to observe is influence of international newscasts. If the global economy, or even the economy of a single country is experiencing a financial crisis, the stock market will follow that trend. But those trends are most likely to be relatively slow and easy to predict even looking at the stock prices itself. What is more dangerous are catastrophes. After a catastrophe there is often a low point on the stock market. This low point comes so fast, that only algorithms that observe the news have a chance to prevent huge losses.

Another to note is, that we did all our tests on historic data of the stock market. Even if we *bought* a stock at some specific point of time, we did not really buy this stock on the real market. So our tests did not affect the stock market at all. Because buying and selling stocks on the market while testing, would always affect the received results.

B. Motivation

In this paper we are going to compare some algorithms and their functionality. The main goal of these algorithms is, to predict a rise or fall in the price before the actual rise or fall happens. It is very important to find some kind of pattern in the stock prices of the past to do a accurate prediction of the future prices. We also like to share some of our experimental results we had with the algorithms. We only used algorithms that do not take global news in account because we wanted to see, how good the algorithms were in an isolated environment analysing past prices only. We decided to do it that way, because the algorithms themselves should work properly and examining news would be an easy addition to these working algorithms.

The main goal of the project was, to find a good working stock forecast algorithm to predict future stock trends. Later in this paper, we will introduce parameters and variables, that give the best results combined with our algorithms.

C. Document overview

Section II briefly covers the background information needed for the reader to understand the most important points about our project.

Section III describes the algorithms used in more detail.

Chapter IV is about the test-results received using the algorithms described in chapter III. All algorithms were tested with different inputs and parameters.

In section V the results are interpreted and we propose how our algorithms can be improved to be successful in the real stock market.

In chapter VI some thing extensions to our algorithms are given, which could be done in future.

II. Essentials

We investigated algorithms that bring good results with time series analysis. The computer programs we implemented were specialized making a good prediction only looking at one single stock and its trend. These algorithms were tested on real stock data we fetched from the yahoo finance API.

We used yahoo finance because it gives easy access to numerous stocks and their historical prices. The API yahoo grants is pretty simple. The only thing you have to do to download the data you need is, preparing a link so it fits your needs. For example the url -

http://real-chart.finance.yahoo.com/table.csv?s=IBM&a=00&b=2&c=1962&d=04&e=21&f=2015&g=d&ignore=.csv

- provides the data in a csv-file. The stock that this link is going to download is IBM (International Business Machines Corporation). It is also possible to adjust the timespan and format of the data. This method of getting historical stock market data is pretty easy to implement in various programming languages. It is also possible to download the csv-file to your computer with this link.

After acquiring the necessary data, we started implementing the algorithms that we thought of.

III. ALGORITHMS

We implemented three different algorithms to predict stock prices and maximise the outcome. All three of them only consider one stock at a time and do not take any global news or trends in account. In this section describes the algorithms used in detail

A. *Algorithm using the latest trend*

With this algorithm we wanted to make a precise prediction on the stock market. It analyses the closing prices of a specific stock and gives a forecast how the price will develop. We also wanted the algorithm to be able to predict not only the price for the next day, but for the next few days. In section IV we will demonstrate that we were able to do a good prediction even 4 days in the future. But it must be said that the more in the future the prediction is, the bigger the gap between the real prices and the prediction.

The algorithm itself uses only the last few days of a stock. The number of days used for this algorithm is not static. The amount of days that give the best result may vary for different stocks. Nevertheless if the number of days stays within a specific range, the results are pretty accurate.

$$M = \sum_{n=0}^{D} C_{n+1} - C_n$$

$$prediction = \frac{M}{D}$$

$$stock_{predicted} = stock_{today} + prediction$$

Where D is the number of days the prediction is based on and C is the closing price of the nth day. After you calculated the prediction, the only thing that is left to do, is adding the prediction to the actual stock price. The predicted stock price here is for the next day only.

To get the prediction for the day after that, do this calculation again but take the predicted stock into account.It is possible to do this over and over again for as many days in the future as you would like. Though it is not meaningful for more than a few days because the prediction gets more and more unreliable. For detailed results an the amount of days that are rational, see chapter IV on page 3.

B. *Algorithm using long term moving average*

This algorithm is based on analysing the moving average of a stock. It does not make a prediction about the precise prise of a stock the next day. The only thing it does, is giving an advice when to buy a stock and when to sell it. If you follow the advices of this algorithm, it is possible to have a positive outcome even if the trend of a stock is overall going down.

The first thing we had to do was calculating the moving average of a stock. This can be computed as below-

$MA_n = \frac{C_D + C_{D-1} + \ldots + C_{D-(n-1)}}{n}$

Where C is the closing price of a stock on a specific day D.

Now we need to calculate MA_{50}[1] and MA_{200}[2] of a specific stock we want to invest in.

If the condition $MA_{50} > MA_{200}$ is fulfilled the algorithms gives the advice to buy the stock on this day. After the stock has been bought, the algorithm checks every day if the requirement is still met. As soon as $MA_{50} < MA_{200}$ the algorithm recommends to sell the stock. This algorithm seems to be very simple but it is worth noticing, that the outcome was pretty solid in the test we made. For detailed results, see chapter IV on page 3.

C. *Algorithm using supervised learning techniques*

In forecasting future stock trends it is common to not use the stock prices itself. Instead other values based on the stock prices are first calculated and then used as predictor variables for the learning process. One possible value is the daily return which is:

$Return_i = \frac{Close_i - Close_{i-1}}{Close_{i-1}}$

For this algorithm three different techniques are used to learn the model of the stock trends. Those are Logistic Regression, Linear Discriminant Analysis and Quadratic Discriminant Analysis. To test them the returns of the past two days are used as predictor variables.

Through Logistic Regression the probability of a following day being categorized as ”Up” or ”Down” is modelled. The function used for logistic regression is the following:

$p(Y = U|L_1, L_2) = \frac{e^{\beta_0 + \beta_1 L_1 + \beta_2 L_2}}{1 + e^{\beta_0 + \beta_1 L_1 + \beta_2 L_2}}$

Where $p(Y = U)$ is the probability that the next day's stock price is increasing. L_1 and L_2 are the lagged returns of the past two days.

Linear and Quadratic Discriminant Analysis both treat their predictor variables independently at first and combine them later using Bayes' theorem.

[1] Moving average of the last 50 days
[2] Moving average of the last 200 days

IV. RESULTS

This chapter is about the results we had with each algorithm individually. We will also discuss which parameters gave the best and worst results and why that happened.

A. Algorithm using the latest trend

How this algorithm works in detail is described in section III. This algorithm is all about predicting the exact price of a stock for the next days. With this algorithm we tried to predict if a stock would go down or up on the next day. We also tried to predict more than just one day in the future. Later in this chapter we will show how good our predictions were on the first and on the nth day in the future. It is understandable that, the more you go into the future, the more unreliable the predictions become.

To get better and more reliable results, we did not test this method on single stocks. We wanted to get a big amount of predictions and see whether they were good. Because if you only predict one single price it is not really significant, but if you predict many prices and take the average of these predictions you get meaningful results. That is why we took 150 stocks for our algorithm to predict. We tested 3 sets of data with 150 stocks each.

Also a very important thing is the amount of days the algorithm uses to make its prediction. We got the best results using something in the span of 10-15 past days. If you use more than 15 days, it does not really take the latest trend into account and if you use less than 10 days the timespan to get a good prediction is to short.

Days in the future	Set 1	Set 2	Set 3
Day 1.	0.043 $	0.012 $	0.028 $
Day 2.	0.049 $	0.010 $	0.033 $
Day 3.	0.059 $	0.013 $	0.041 $
Day 4.	0.089 $	0.017 $	0.038 $
Day 5.	0.092 $	0.024 $	0.049 $
Day 6	0.126 $	0.035 $	0.062 $
Day 15.	1.740 $	0.092 $	0.107 $

Table 1. Difference between prediction and real data

You can see in this table, that the prediction for the first 3-4 days is relatively accurate. However the time after that the difference between prediction and real data gets bigger. Nevertheless in the first 6 days its still in an acceptable scope. But there are relatively big differences between the sets. These differences come with the fact that, if a stock is going smooth out prediction algorithm is pretty accurate. The algorithm is not good on predicting big price drops so if that happens the prediction gets pretty inaccurate. We suppose that in our first test set, there were more stocks with price spices than in the other to sets. That explains, that the predictions of the first set are not as accurate as the other two sets. So we consider our algorithm to be pretty accurate if the stock market goes smooth and has no big spikes because of some external reason. A span of a few cents in a prediction of 150 stocks is pretty accurate in our opinion.

B. Algorithm using long term moving average

As described in Chapter III this algorithm uses two different MA's to make a good prediction when to buy and when to sell a stock. It is not an algorithm to predict an exact price but rather to make profit without knowing the exact price the stock is going to have the next days.

Fig. 1. Moving Average of MSFT (Microsoft Corporation)

In Figure 1 you can see the development the stock of Microsoft Corporation™made, in the last 11 years. If you look closely, there is also a red and green line in this Figure. The green line is the MA of the last 200 days of every point in this graph. The red line marks the MA of the last 50 days. Now as described in section III, we used these two things to figure out when to buy and when to sell a stock.

We now take a closer look at a single passage of the stock to explain our approach in more detail.

Fig. 2. MSFT stock zoomed

Figure 2 shows only a small segment of the stocks development. You can see now, that the real stock price (blue line) is way different to both of the MA's. If you look closely, there is a red and a green dot on the stock price. This dots mark the spots, where the 200 day MA and the 50 day MA cross each other. Every time this happens, it is best to either sell of buy the stock. If the dot is green, the algorithms says, that now is the right time to buy the stock for the current price. And if it is red, the best thing is to sell the stock. The dots are on the blue line, because it marks the real price of the stock at this point of time. With this algorithm, we had good results as shown in the table below. We only investigated the last 1000 days of a stock for our test because we thought this was enough data to work with. So we tested our algorithm and bought and sold stocks based on the algorithm. In the first column of the table below you see the stocks we tested the algorithm with. In the second column, we wrote down, how much profit we made buying and selling only **one** instance of the stock. Just to make this clear, if you would by two instances of the stock you would make double the profit of loss we made. Remember that this are only examples, this results are not generalizable in any way. But we did not only do test with single stocks as they are not to meaningful. That is why we generated 100 random stocks and used our algorithm on all of them at the same time to get more significant results.

Stock(s) used	Outcome
MSFT[3]	13.39 $
MCD[4]	2.71 $
YHOO[5]	20.05 $
NVDA[6]	-13.70 $
INTC[7]	5.70 $
AMZN[8]	128.87 $
Random combination 1.	34.53 $
Random combination 2.	-17.32 $
Random combination 3.	56.94 $
Random combination 4.	15.74 $

Table 2. Results of LTMA

As you can see in the table, our results with this algorithm were pretty good. While the single stock outcome is not very expressive, the results of the random combination of 100 stocks is also fine. If we only take the combinations of stocks into account, we made a total profit of **89.89 $**. However this are only our test results with the amount of stocks we tested in a special period of time. Even with these test results it is not ensured, that our algorithm would always or even most of the time make profit.

C. *Algorithm using supervised learning techniques*

The algorithms were implemented in python utilizing the scikit-learn library.

The following table shows the results of applying different machine learning techniques to stock data. All used methods achieve similar results. The accuracy of the prediction is barely higher than the probability of a coin flip. But none of the cases do have a prediction accuracy below 50%. The algorithm, as it is, should not be used without further improvements and refinement.

	^AXJO[9]	^FCHI[10]	^FTSE[11]	^GDAXI[12]	^IXIC[13]
LDA	0.501	0.541	0.527	0.555	0.571
LR	0.501	0.541	0.527	0.555	0.571
QDA	0.541	0.513	0.519	0.539	0.575

Table 3. Results of Supervised Learning Algorithms

V. Conclusion

The conclusion of this paper is, that you can get pretty good results in forecasting stock prices and try to get profit out of the stock market even with algorithms that do not consider the global news. However the stock market is pretty fragile and fast-paced, so while you are trying to make profit, it is always a possibility that the exact opposite happens. But with the right algorithms it can be worth the risk of investing into the market.

VI. Future Work

We implemented good working algorithms that can predict the stock price. But if these algorithms would take the global news into account there prediction could be better. With our algorithms we can never have a perfect result because if something global happens that affects the stock market, our algorithms are not fast enough to react and sell all stocks we bought. So the addition of a newscrawler would be a good idea for future work because it would help making the algorithms even more reliable in unforeseeable circumstances.

Appendix

List of Abbreviations

MA - Moving average
LTMA - Long term moving average Algorithm
LTA - Latest trend algorithm
LR - Logistic Regression
LDA - Linear Discriminant Analysis
QDA - Quadratic Discriminant Analysis

Acknowledgment

The authors would like to thank Prof. Dr. Joern Fischer for his help and support while researching in this subject

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[3] Microsoft Corporation™
[4] McDonald's Corp™
[5] Yahoo! Inc.™
[6] NVIDIA Corporation™
[7] Intel Corporation™
[8] Amazon.com Inc.™
[9] Australia ASX-200
[10] Paris CAC 40
[11] London FTSE-100
[12] Frankfurt DAX
[13] NASDAQ Composite