

Event-based stock market prediction

Hadi Pouransari, Hamidreza Chalabi

1 Introduction:

There are various studies on the behavior of the market. In particular, derivatives such as futures and options have taken a lot of attentions, lately. Predicting these derivatives is not only important for the risk management purposes, but also for price speculative activities. Besides that accurate prediction of the market's direction can help investors to gain enormous profits with small amount of capital [Tsaih et al., 1998]. Stock market prediction can be viewed as a challenging time-series prediction [Kim, 2003]. There are many factors that are influential on the financial markets, including political events, natural disasters, economic conditions, and so on. Despite the complexity of the movements in market prices, the market behavior is not completely random. Instead, it is governed by a highly nonlinear dynamical system [Blank, 1991].

Forecasting the future prices is carried out based on the technical analysis, which studies the market's action using past prices and the other market information. Market analysis is in contradiction with the Efficient Market Hypothesis (EMH). EMH was developed in 1970 by economist Eugene Fama [Fama, 1965a, Fama, 1965b] whose theory stated that it is not possible for an investor to outperform the market because all available information is already built into all stock prices. If the EMH was true, it would not be possible to use machine learning techniques for market prediction. Nevertheless, there are many successful technical analysis in the financial world and number of studies appearing in academic literature that are using machine learning techniques for market prediction [Choudhry and Garg, 2008].

One way to forecast the market movement is by analyzing the special events of the market such as earnings announcements. Earnings announcement for each stock is an official public statement of a company's profitability for a specific time period, typically a quarter or a year. Each company has its specific earnings announcement dates. Stock price of a company is affected by the earnings announcement event. Equity analysts usually predict the earnings per share (EPS) prior to the announcement date.

In this project, using machine-learning techniques, we want to predict whether a given stock will be rising in the following day after earnings announcement or not. This will lead to a binary classification problem, which can be tackled based on the huge amount of available public data. This project consists of two major tasks: data collection and application of machine learning algorithms. In §2, we will discuss our efforts to collect the required data. In continuation in §3, features definitions and selections are described. We have considered and discussed different machine learning algorithms in §4. In §5, the summary of our results and possible extensions are explained.

2 Data collection

Data collection is one of the most crucial steps in every machine learning problem. For the purpose of this project, we need two sets of data: 1) the daily stock market information, and 2) the earnings calendar data.

There are different websites that provide the daily stock market information (e.g., www.google.com/finance and finance.yahoo.com). There are some available APIs that can be used to fetch the historical data of market information.

The Earnings Calendar data is much more challenging to gather. In the first part of the project, we aimed to collect all the required data from publicly available web-sites. Although there are many available web-sites that provide daily information for the stock prices and earnings announcements, only a few of them provides historical data. In particular, www.nasdaq.com and finance.yahoo.com provides historical data for different companies, but there is no publicly available API to collect the earnings announcements data. Hence, in the first part of the project, we decided to collect these data using web-scraping codes.

We wrote couple of python programs in order to collect the earnings data from aforementioned web-sites. The summary of data is as the following:

- Earnings data from Nasdaq: For each company and for each announcement, we created a dictionary that stores name, fiscal year, market cap, estimated EPS, number of estimations, and EPS.
{SYMB:{DATE:{name, fiscal year, cap, est. EPS, # est., EPS}}}
- Earnings data from Yahoo: For each company and for each announcement, we created a dictionary that stores names and announcement times.
{SYMB:{DATE:{name, time}}}
- Stock prices data from Yahoo: For each company and for every day, we created a dictionary that contains stock volume, high stock value, low stock value, opening stock value, closing stock value.
{SYMB:{DATE:{vol., high, low, open, close}}}

These data are gathered for the period of time between Jan. 1st, 2009 to Nov. 1st 2014 and stored in structured databases. Even though, we have stored all the required data for the purpose of this project, the written codes provide the flexibility to implement the online learning algorithm.

The objective of the project is to predict sign of the *jump*¹ of the stock price right after earnings announcement. Different companies announce their earnings at different time of the day (before market or after market). In order to calculate the price jump right after the earnings announcement, we need to have the announcement time as well. For example, Apple Inc. announced its last earnings on October 20th 2014 after market was closed. Therefore, the correct jump for our consideration is the difference between the opening price on October 21st and the closing price on October 20th. Note that this calculation would be different if the announcement time was “before market”. The earnings time data is not provided by Nasdaq website. Therefore, we have written another crawler to collect the earnings time data from Yahoo website. Finally, we merged the data obtained from these two sources. The overall data (daily stock prices + earnings announcements) obtained for more than 5000 companies over 5 years took about 2GB of memory. This data set is much more complete compared to many similar published works. For instance, [Bao et al., 2004] considers only 140 data points for one company over the period of 6 months.

3 Feature selection

Having gathered a big data set for stock prices and earnings announcements, we are able to define various numerical features out of it. For each company and for each earnings announcement, we consider all the information from a year before the announcement date. This brings about one learning example data. Note that each company creates several learning examples (based on the number of earnings announcements it has).

We have defined 54 numerical features for each learning example. In order to select the most useful features among all of these 54 features, we used **filter feature selection** with scores assigned as the absolute value of the correlation between features and the objective.

Figure 1 illustrates the scores of different features.

Here is the list of top 10 features according to their scores:

1. Surprise factor[0] > 0².
2. EPS[0] > EPS[-1].
3. EPS[-2] - 2 EPS[-1] + EPS[0] > 0.
4. Earning Jump[-1] > 0.
5. Standard deviation((High - Low)/close) / Mean((High - Low)/close) in last 90 days.

¹Jump is defined as the difference between the opening stock price and the closing price of the previous day.

²Surprise factor is defined as (EPS - estimated EPS)/EPS. This parameter quantifies how unexpected the announced EPS is.

6. Standard deviation((High - Low)/close) / Mean((High - Low)/close) in last 10 days.
7. 1 Billion < Market cap < 10 Billions.
8. Standard deviation((High - Low)/close) in last 90 days.
9. Earning Jump[-3] > 0.
10. Earning Jump[-2] > 0.

Note that our objective function is:

$$y = \text{Earning jump}[0] > 0$$

In the above definition the number inside the brackets shows the number of periods before the date of learning example. For instance, Jump[-1] means the price jump in the last earnings announcement prior to the current one.

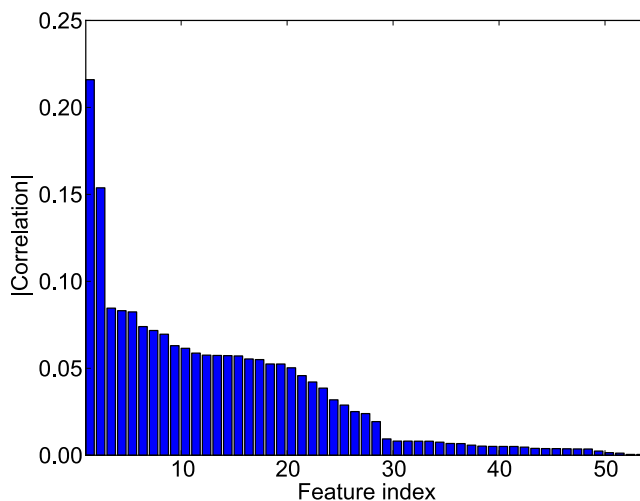


Figure 1: Scores of features based on their correlations with the objective function.

4 Machine learning algorithms

We have decomposed our data into two data sets. The first data set which is used as the *training set* includes all the data prior to April 1st 2014, and the second one that is used as the *test set* consists of the data from April 1st 2014 to November 1st 2014. Basically, this corresponds to a **cross-validation** method with about 85% training set and 15% test set. In the following we are going to investigate different machine learning algorithms for our prediction goal.

4.1 Logistic regression with regularization

We first implemented the Logistic regression by incorporating the regularization. Here, we are seeking for a vector θ that maximizes the log-likelihood function. To do this, we performed stochastic gradient descent algorithm:

$$y = [\theta^\top x > 0]$$

$$\theta := (1 - \gamma)\theta + \alpha(y^{(i)} - h(\theta^\top x^{(i)}))x^{(i)} \quad , \quad \alpha = \frac{10^{-4}}{\sqrt{\#\text{of iteration}}}$$

	Gaussian	Linear	Polynomial (3rd degree)	Sigmoid
Training error	37.7%	39.5%	37.7%	46.0 %
Test error	36.0 %	37.2%	36.0%	46.2%

Table 1: Obtained errors using SVM with different kernels as applied to **full data set** (40k training examples).

The convergence plot for our stochastic gradient descent (SGD) algorithm is shown in Figure 2. The obtained training and test errors are 37.8% and 36.4%, respectively. Please note that as explained in §1, predicting the market movement is a very difficult and challenging task, and our accuracy is in fact pretty reasonable. Many of the other previous works have obtained similar (or less) accuracies [Bao et al., 2004, Lee, 2009, Kim, 2003, Choudhry and Garg, 2008]. For further verification of the results of the SGD, we also implemented the

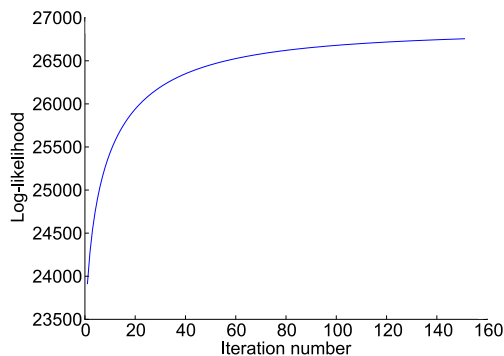


Figure 2: Convergence of the stochastic gradient descent method for maximization of the log-likelihood function for the logistic regression method.

Newton’s method to obtain the θ of logistic regression. The runtime of the Newton’s method was about the same as SGD for the current size of feature vector (10 dimension + 1 for interception). The training and test errors were 37.8% and 36.1%, respectively.

4.2 SVM with different kernels

Many similar works have used Support Vector Machines (SVMs) in order to obtain a right balance between the empirical error and the VC-confidence interval [Bao et al., 2004]. Generally, this results in a better generalization performance compared to the logistic regression method. We used [Pedregosa et al., 2011] in order to implement SVM with l_2 -regularization and different kernels. In table 1, training and test errors using the full data set for different kernel choices are tabulated. Similar to [Choudhry and Garg, 2008], this table shows that the Gaussian kernel outperforms the others.

The regularization factor in SVM provides a trade-off between bias and variance errors. In Figure 3(left) we have shown how the training error drops as C coefficient increases. Note that to obtain this figure we have used all 54 features (not only the top 10 ones). This shows us that SVM is capable of completely overfitting the data, whereas the test error increases for large values of the coefficient C . Moreover, in Figure 3(right) we have demonstrated the learning curve achieved by $C = 50$. The plot shows a reasonable trend: higher(lower) training(test) error as the training set size increases.

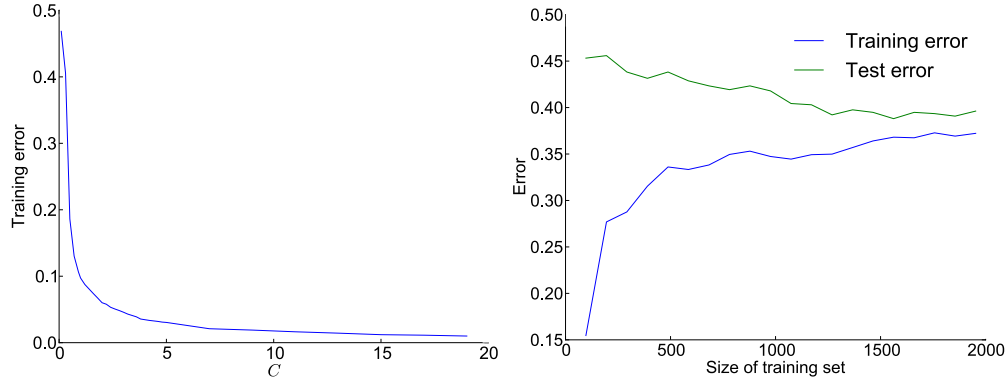


Figure 3: Left: Variation of the training error with the regularization coefficient using SVM with Gaussian kernel. Right: The learning curve obtained from a portion of the data set as our training set.

5 Conclusion and future work

In general, the problem of stock market prediction is very challenging, and very high accuracies is not achievable. Nevertheless, machine learning techniques can provide reasonable market movement predictions, that can be used by investors. The calculated results show that using support vector machines with Gaussian kernel and regularization outperforms the logistic regression, and SVM with other kernels. In this study, we were able to get a prediction with accuracy of 64%. We should emphasize that the current dataset is very valuable, and many extensions to the current algorithm can be applied to it.

First of all, we could expand the feature space to higher dimensions, and try other feature selection methods such as wrapper model feature selection. Secondly, as a natural extension to our work, one can predict the value of the price jump, rather than just the sign of it. Moreover, one can categorize companies based on their mutual influence on each other, and find an individual learning parameter, θ , for each category (e.g., IT companies, energy related companies, health care companies, etc.). Additionally, for each feature, one can compute the correlation matrix of that feature between different companies. This can be used as an extra information for prediction purposes.

References

- [Bao et al., 2004] Bao, Y., Lu, Y., and Zhang, J. (2004). Forecasting stock price by svms regression. In *Artificial Intelligence: Methodology, Systems, and Applications*, pages 295–303. Springer.
- [Blank, 1991] Blank, S. C. (1991). “chaos” in futures markets? a nonlinear dynamical analysis. *Journal of Futures Markets*, 11(6):711–728.
- [Choudhry and Garg, 2008] Choudhry, R. and Garg, K. (2008). A hybrid machine learning system for stock market forecasting. *World Academy of Science, Engineering and Technology*, 39:315–318.
- [Fama, 1965a] Fama, E. F. (1965a). The behavior of stock-market prices. *Journal of business*, pages 34–105.
- [Fama, 1965b] Fama, E. F. (1965b). Random walks in stock market prices. *Financial Analysts Journal*, pages 55–59.
- [Kim, 2003] Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1):307–319.
- [Lee, 2009] Lee, M.-C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, 36(8):10896–10904.
- [Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- [Tsaih et al., 1998] Tsaih, R., Hsu, Y., and Lai, C. C. (1998). Forecasting s&p 500 stock index futures with a hybrid ai system. *Decision Support Systems*, 23(2):161–174.