A news-based portfolio management system Development and Application of Data Mining and Learning Systems

Sattarov, Timur

Informatik III, Universität Bonn, Germany
timur.sattarov@gmail.com

Abstract

A news-based portfolio management system uses stock price predictions, derived from the news, in order to construct a maximum - profit portfolio, according to a portfolio strategy. For the prediction Support Vector Machines (SVM) is used. The training of SVM is based on taking news and past stock prices as examples and then predict whether the stock price will increase/decrease significantly. For such a prediction, also a confidence degree is provided, which basically tells how strong a certain prediction is.

1 Introduction

The analysis of the stock market is an area of research that causes often a lot of attention. A great number of investors would like to know about the future of the stock market to be able to make successful investments and increase the profit as well. In order to support investors in making a future market decision, different kinds of prediction techniques can be used. During the last several decades this prediction was mainly based on the historical market prices of the stocks, however only the historical data is not enough to make sound predictions. It has been observed that the news are one of the main factors that make the most influential effect on the behavior of the stock market, thus they are necessary to obtain more accurate predictions.

1.1 Motivation

There are different kinds of techniques for predicting the behavior of the stock market, such as: Neural-Networks [6], Genetic Algorithm [1], Support Vector Machines [7],[8]. However one can hardly find a powerful prediction machine for the stock market that could show the future trend of each stock with a certain confidence. There are some recent researches in stock market prediction using SVM as well. For instance, rather good generalization was obtained in [3] using SVM with a composite quasi-linear kernel function to get the stock market prediction. In addition, high accuracy was achived in [5] in proposed classification based method for classifying news data and predict a stock direction. Similar accuracy level was obtained in [2] classifying financial news. A news-based portfolio management system not only predicts the stock trends based on the news, but having the prediction, it yields the most optimal strategy that maximizes the return of each asset with minimum risks at the end of a certain period.

1.2 Goal

The task was analyzed from two perspectives. On the one hand, to make a correct prediction for the given stocks on the market. On the other hand, having some prediction, create several buying and selling strategies for the portfolio and analyze them with respect to the profit they give at the end of a certain period. The next section will cover prediction approach and the techniques that were used. After that, section 3 describes the budget that portfolio can use to buy and sell the stocks during the trading period. Description of all buying and selling strategies will follow after that. The last section will present obtained experimental results, providing also information about the methodology that was followed to obtain them and also the description of the datasets.

2 Prediction

Support Vector Machines (SVM) was selected as one of the "Top Ten Algorithms in Data Mining" [10]. In addition, it was much superior to other machine learning algorithms in prediction of the stock market [4]. It was chosen for the prediction part as it is one of the best classification algorithms with very strong regularization properties and provides a confidence level as well. Moreover, it handles very good with the tasks with high dimensionality and also the cases when number of samples is smaller then the number of features.

An existing python implementation was used, namely the SVM Classification class of the "sklearn" package. For each prediction of the stock, a certain confidence degree is provided with three different values for each predicted class. Based on the fact that confidence degree is in the interval [0,1], it can be used as probability estimate, however it may be inconsistent since it is just calibrated [9] distance from the sample to the separating hyperplane in N-dimensional feature space. The labels for each stock are obtained based on the price movements. If the price of a stock of a current day is less than the price of the same stock on the next day, then the label is "1", if the prices are equal then it is "0", otherwise the label is "-1". The features for each stock are generated based on the historical data of daily news. In order to extract them from daily news, the first step was the preprocessing of the news text. It includes removing of the stop words like conjunctions, particles or common function words. In addition, the stemming is also applied to chop the endings of the words and lemmatization to morphologically analyze the words and bring them to the common lemma. The second step was to rank the words by importance. The frequency of each word and Term Frequency-Inverse Document Frequency (TF-IDF) techniques are used in this step. As a result, each stock will have a separate model.

3 Budget

As one of the initial parameters, the amount of budget for buying stocks into the portfolio should be determined. However, the approach that was used does not imply to spending the whole budget during buying and selling period.

The initial budget is being split into the $MainBudget$ and *TheStockpile*. The $MainBudget$ is used to buy and sell stocks during the whole trading period. For the experiment, the $MainBudget$ was chosen as an arbitrary value of 90% of the initial budget as the starting point. *TheStockpile* takes the rest (10%) of the initial budget but it is used only in a special case, when the level of $MainBudget$ is not enough to buy at least one stock on the market but the current prediction shows that there is at least one rising stock on the market with the confidence higher than 90%. In this case, *TheStockpile* is spent on all the rising stocks with the confidence higher than 90%, according to the current strategy.

The reason of using such an approach is to try to protect the budget from situations of loss when the portfolio is absolutely empty. The other reason is to make a profit in case the

MainBudget is unavailable but the stocks are predicted with very high confidence. For the experiments, the initially chosen budget was 100000 dollars.

4 Strategies

This section will provide a short description of each buying and selling strategy. In particular, we were interested in finding the best combination of buying and selling strategies, that gives maximum profit at the end of a certain period. Therefore, using some heuristic 3 buying and 6 selling strategies were generated for this experiment. Every strategy has a different approach and results in a different quantity of certificates of a certain stock to buy or sell in a certain market day. In every formula, describing the strategy, n is the number of stocks and $j = 1, ..., n$.

4.1 Buying strategies

In each buying strategy the stocks that are predicted as increasing are chosen to fill in the portfolio. The strategies that have been used for buying are:

- **Buy equally** (buyEqually)

  This type of strategy equally distributes the budget on the stocks, that are predicted to increase, within a certain market day. The amount of the certificates of each "rising" stock in a certain day is calculated according to the formula:

$$quantity_{stock_j} = \frac{currentBudget}{\sum_{i=1}^{n} price_{stock_i}} \tag{1}$$

- **Buy relative to the confidence** (buyRelConf)

  This strategy distributes the budget based on the confidence of the stocks predicted to increase. The portion of the budget to spend on each stock is calculated by dividing the confidence of the stock to the sum of all confidences of the "rising" stocks. As a result, the quantity of the certificates of each "rising" stock in this strategy is the portion of the budget to spend on this stock divided by the current price of this stock.

$$quantity_{stock_j} = \frac{currentBudget * \frac{confidence_{stock_j}}{\sum_{i=1}^{n} confidence_{stock_i}}}{price_{stock_j}} \tag{2}$$

- **Buy top K relatively to the confidence** (buyTopKRelConf)

  This type of the strategy is almost similar to the **buyRelConf**, the difference is only that it first takes top **K** of the "rising" stocks according to the confidence and then applies **buyRelConf**.

4.2 Selling strategies

In each selling strategy the stocks that are predicted as decreasing are chosen to be sold from the portfolio. The strategies that have been used for selling are:

- **Sell all stocks** (sellAll)

  This type of selling strategy empties the portfolio by selling all the certificates of all the stocks, which are currently in the portfolio, no matter what prediction and confidence they have.

- **Sell all falling stocks** (sellFalling)

  This strategy sells all the certificates of the stocks which predicted as "falling down".

- **Sell relatively to the confidence** (sellRelConf)

  This type of strategy is also very similar to the **buyRelConf**. However the quantity of the certificates of a "falling down" stock is calculated differently. It is also highly dependent on the confidence of the current stock and on the confidences of the other "falling down" stocks. The amount of certificates of each stock is calculated according to the formula:

$$quantity_{stock_j} = \frac{confidence_{stock_j}}{\sum\limits_{i=1}^{n} confidence_{stock_i}} * Portfolio.quantity_{stock_j} \tag{3}$$

- **Sell top K relatively to the confidence** (sellTopKRelConf)

  This type of the strategy is similar to the **sellRelConf**, the difference is only that it takes top **K** of the "rising" stocks and then applies **sellRelConf**.

- **Sell all top K** (sellAllTopK)

  This strategy takes all the stocks that are currently in the portfolio, pick only those that are predicted as "falling down", sort them according to the confidence from the highest to the lowest and sell all the certificates of the top **K** stocks.

- **Sell relatively to the confidence and current portfolio** (sellRelConfPortf)

  This strategy is a bit more advanced. It is also dependent on the amount of all certificates of all stocks currently in the portfolio. The portion of the quantity of the stock to sell is calculated again according to the confidences. But then this portion is multiplied by the sum of all certificates currently in the portfolio. That kind of approach allows selling much more stocks, which are more likely to fall down in comparison with **sellRelConf**. In this strategy the quantity of certificates of each stock, which need to be sold is exactly the portion of the confidence degree, with respect to the sum of all certificates. For instance, if there are three stocks in the portfolio with the amount of 5,10 and 15 stock certificates, and if the portion of confidence degree gives 10%, 20%, and 30% for each stock, then the quantity of certificates of each stock, which should be sold, will be 3,6 and 9 respectively, according to the formula below:

$$quantity_{stock_j} = \frac{confidence_{stock_j}}{\sum\limits_{i=1}^{n} confidence_{stock_i}} * \sum_{i=1}^{n} Portfolio.quantity_{stock_i} \tag{4}$$

5 Experiments

5.1 Datasets

The news datasets were taken from ”Bloomberg” and ”Reuters” news agencies. The ”Bloomberg” dataset contains the news of a period from the 3 January 2011 till 30 November 2012. The ”Reuters” dataset contains news from 1 January 2011 till 19 April 2013.

The quantity of the price datasets, that were used, are also two. One of them contains the prices from the stocks, that are listed in ”Standard & Poor's”. Another one is the dataset with the prices of randomly chosen stocks. Both contain the prices of the stocks during the period of 8 years from 1 January 2004 till 28 December 2012. Each dataset has 100 stocks of NASDAQ and NYSE stock exchange. Some of the values of the prices in the datasets are ”NaN” or zero, because of the weekends, holidays, etc. In this case, these values are replaced by the same price of the previous day. If the previous day doesn't have any price, then the current price is zero.

5.2 Metrics

In order to have a certain picture about the accuracy of the prediction, several measurements were chosen, such as:

- Precision, which shows the hit rate of correctly classified stocks in the prediction and impacts the profit/loss at every moment. It is calculated using the formula: $\frac{tp}{(tp+fp)}$, where tp is the number of true positives and fp is the number of false positives.
- Recall, which shows the proportion of correctly classified stocks, with respect to all the stocks, that should be predicted of a certain label. The calculation is the ratio $\frac{tp}{(tp+fn)}$ where tp is the number of true positives and fn is the number of false negatives.
- F1-score is a combination of precision and recall, which gives a broad outline about the correctness of the prediction, based on the formula: $2 * \frac{(precision*recall)}{(precision+recall)}$

5.3 Methodology

Each experiment is done during a certain fixed period. After initializing the strategy instance, it is associated with a list of all possible combinations of buying and selling strategies. After that, each combination of strategies is applied on the portfolio to obtain the profit results at the end of the period.

The general framework of the experiments. The experiment was divided into two parts: real prediction and randomly generated prediction. The real prediction is the prediction made by the SVM approach using the real stock news and stock prices. Randomly generated prediction just creates the random values for the prediction and the confidence. According to the recent researches, that were mentioned in 1.1, SVM-based prediction approach gives 70%-80% accuracy, which is rather good result. Thus we would like to see what will be in the opposite case and considered 20% of correct prediction as a good classifier. As a result, we left 20% as correct prediction with confidence degree between [0,1] and generate 80% of the prediction randomly, labeling them 1,0 or -1 and giving them a confidence level between [0,0.3] in order to get closer to the real life values.

Before gathering all the statistics of the experiments we had to determine optimal parameters for the framework such as: find the **Best K** in buying and selling strategies and find an appropriate **Price Difference** value. These parameters will be used during the whole process of the experiment. More detailed explanation of these parameters are described in the next subsections.

Presentation of the experimental results. In presenting the results tables, pie charts, plots and boxplots were preferred. Boxplot is considered appropriate for this situation since it shows not only where most of the results are situated, but also what the variance of the experimental results is. The scripting language R was used to generate the boxplots (using the command 'boxplot'), and they depict the median as a bold horizontal line, 50% of the samples are represented with boxes, and the rest of the samples (excluding outliers) are represented with the so-called "whiskers".

5.4 Best K

Some of the strategies that have been generated for the experiments contain the part, where the **Top K** stocks should be chosen. In order to find most suitable **K**, several experiments have been made with all possible values of **K**. Thus, for each buying or selling strategy containing the part with finding stocks with **Top K** confidences, a list of profits is generated according to the parameter **K** from the interval [1,100] where $\mathbf{K} \in \mathbb{Z}$. The experiments were done using random prediction for the stock market.

Plots 1 and 2 illustrate the relation between the profit and different values of **K** during **buyTopKRelConf** and **sellAllTopK** strategies. These plots give the most clear representation of the **Top K**, out of all results. It should be emphasized, that **K** gives maximum profit in the interval from 4 to 10. The rest of the plots with all combinations of the strategies depict almost the same picture, thus it can be assumed, that the range between 4 and 10 is the best value for **K** for the majority of the strategies.

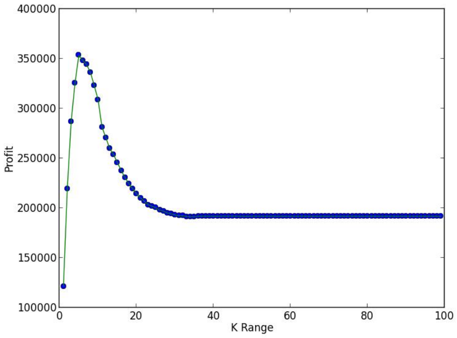


Figure 1: Best K in buyTopKRelConf and sellRelConfPortf

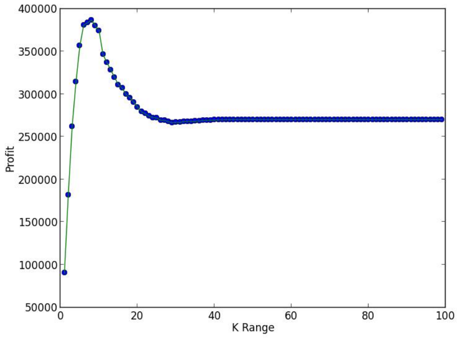


Figure 2: Best K in sellAllTopK and buyTopKRelConf

5.5 Price Difference

The Price Difference parameter is the difference between the prices of the two consequent days of a certain stock. The experiments were done counting on the real SVM prediction. The reason of determining this parameter is to try to minimize the trading risk and make the profit more effective. This parameter influences on the labels for the training of SVM. As a result, the prediction for “rising”/”falling” stocks will be only the stocks that will increase/decrease within a certain percent, the rest will be predicted as “same”. Since in [2] this parameter was used with the value of 0.5% and obtained the accuracy of around 70%, we would like to start with 1% and increment it on 1 percent in each new experiment until SVM will not be able to predict at least one “rising”/”falling” stock.

To have more clear picture about the results, a co-called ”classification report” was made. It shows the precision, recall and f1-score of each of the predicted class with the average value below. The support column shows the number of labels of each class.

However, at first it is better to take a look at the classification report with the price difference of 0 percent, in order to see how the real prediction and prices are distributed. The results are shown in Table 1. The duration of the stock market for gathering the statistics is 506 days (from 01.01.2011 till 28.12.2012).

	Precision	Recall	f1-score	support	Precision	Recall	f1-score	support
fall (-1)	0.48	0.13	0.21	23990	0.33	0.02	0.04	10384
same (0)	0.00	0.00	0.00	34	0.58	0.99	0.73	28963
rise (1)	0.53	0.77	0.62	26576	0.31	0.01	0.01	11253
avg / total	0.50	0.47	0.43	50600	0.47	0.57	0.43	50600
	0% of price difference				1% of price difference			

Table 1: Classification report with 0% and 1% of price difference

Table 1 also shows the classification report with 1 percent of the price difference parameter. Although, the average/total results in both cases are almost identical, recall and f1-score in case of 1 percent of the price difference are very low in “rising” and “falling” stocks. They both are even less than 5 percent in both labels, which means that a very small amount of stocks were predicted for these labels.

Even though recall and f1-score don’t show promising results with the 1 percent of price difference, these results are the best in comparison to the rest of the price difference parameters. The reason is that starting from the value of 2 the results are similar. Table 2 shows the classification report of the price difference parameter of 2 percent. Here the average/total value is much higher than in the previous tables, but it is due to the fact that ”same (0)” class got a rather high value and balanced the rest, which are totally zero. If we try to increase the price difference parameter, this will increase the average/total value in all metrics up to 99 percent, but the ”fall (-1)” and ”rise (1)” classes will always remain zero. As a result, none of the portfolio strategies will make a decision to buy or to sell even one stock and the profit will always remain zero. This occurs due to the fact that the classification algorithm has very low amount of training examples of “rising” and “falling” stocks. In case when the price difference parameter is two, it is even less than 10 percent of the whole quantity of examples (”fall (-1)” - 4314 and ”rise (1) - 4524). And when this parameter is higher than 2 percent, these values are even more less. At the value of 5 percent of price difference they both are around 1 percent of the whole number of labels. As a result, there is not much information to make a correct classification for these stocks and almost all the instances are classified as the majority.

5.6 Results

At first we have tried to reduce the dimensionality of the feature space to two in order to see how the distribution looks like in two-dimensional feature space. Principal Components Analysis (PCA) approach was used to deal with this. Fig. 3 and 4 show the 2D feature space of two

	Precision	Recall	f1-score	support
fall (-1)	0.00	0.00	0.00	4314
same (0)	0.83	1.00	0.90	41762
rise (1)	0.00	0.00	0.00	4524
avg / total	0.68	0.83	0.75	50600

Table 2: Classification report with 2% of price difference

stocks: DELL and AMAZON with original labels of classes. The price difference parameter was chosen as 1 percent, in order to see more labeled points on the plot, and whether they are separable or not. The rest of the stocks give almost the same picture. It should be emphasized that the points on the plots are divided into 4 groups (clusters), but with different labels in each group.

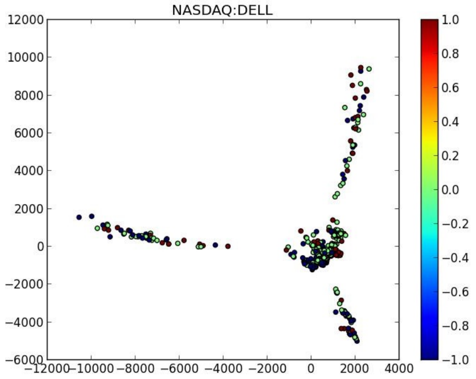


Figure 3: NASDAQ:DELL in 2D

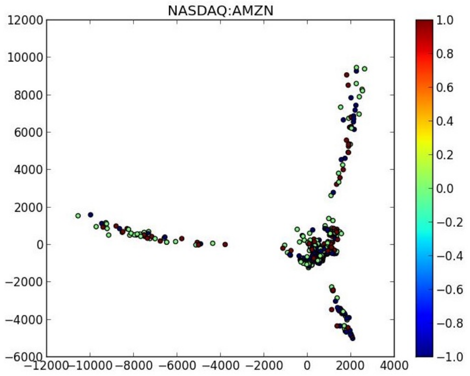


Figure 4: NASDAQ:AMZN in 2D

Randomly generated prediction. The chosen period for the randomly generated prediction was of two years, starting from 01.01.2011 till 28.12.2012. As it was described before, the prediction in this case is generated absolutely randomly and does not use SVM approach.

The boxplot 5 represents the profits of all combinations of buying and selling strategies with randomly generated prediction. Each experiment is repeated 100 times in order to gather the statistics. On the boxplot, the boxes are divided into 3 groups, each of a certain buying strategy. The colors represent different selling strategies.

The boxplot shows that none of the strategies have negative profit, however some strategies are much more effective than others. In the first group, where **buyEqually** is a buying strategy, the mean of the best profit is around 50000 dollars (buyEqually and sellAll). The mean of the best strategy of the second group is somewhere between 50000 and 100000. In fact, this can be considered as a rather good result because reaching the point of 100000 dollars means getting the profit of 100% just in two years, having an initial budget of 100000 dollars according to the section 3. However, the third group with the **buyTopKRelConf** as the buying strategy shows the best result with the average profit even a bit higher than 100000 dollars. Moreover, the majority of the strategies in this group such as **sellAll**, **sellAllTopK**, **sellFalling** and **sellRelConfPortf** have the highest average profits in comparison to any combination of the strategies in any other groups. Thus, this group of the strategies with **buyTopKRelConf**

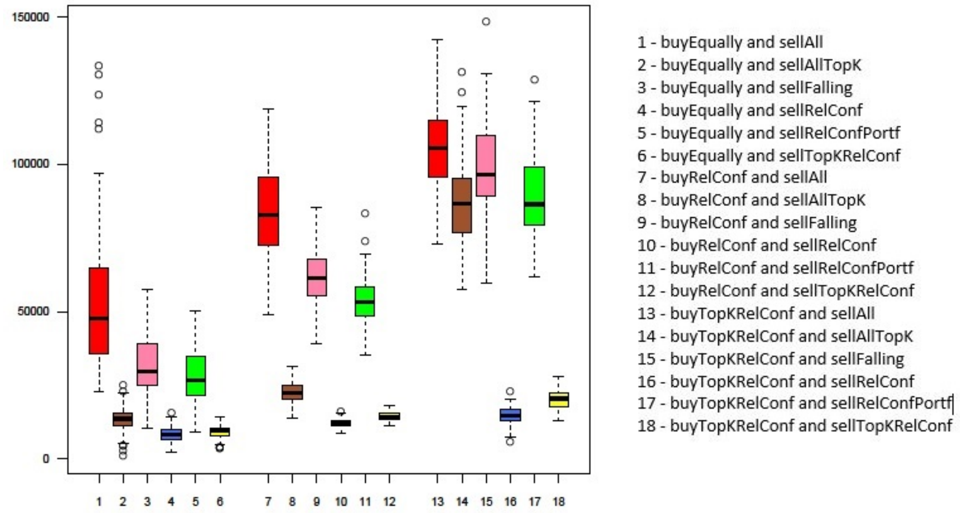


Figure 5: Profits of all strategies with description

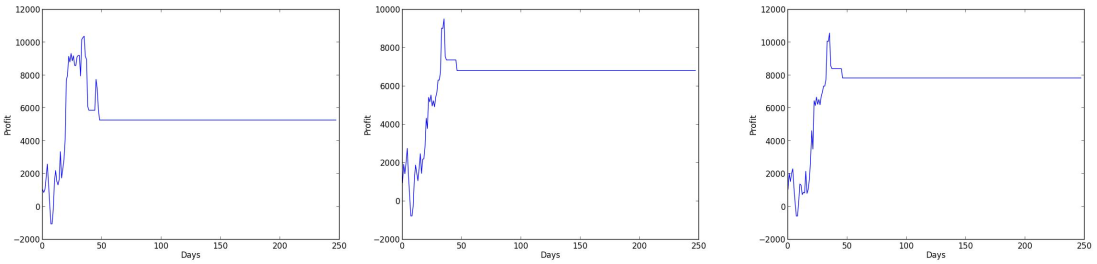
looks much more superior than **buyRelConf** and **buyEqually**. Theoretically, this fact that **buyTopKRelConf** should give higher profit than **buyRelConf** or **buyEqually** should be correct. The reason is that **buyTopKRelConf** chooses the **Top K** stocks with the highest confidences and spend the whole current budget only in this stocks. Thus, the probability of getting more profit is much higher than in case of **buyRelConf** when the strategy just spreads the whole current budget to all the rising stocks according to their confidence. In this case the stocks which have a low level of confidence will also be bought into the portfolio and can be losing. Even worst is the case of **buyEqually** when all the stocks are bought equally, no matter what confidence they have. Therefore, even large amount of stock with low confidence can be bought into the portfolio resulting in wasting the budget.

In addition, boxplot shows that some selling strategies, such as **sellRelConf** and **sellTopKRelConf** give the lowest profit, with any combination of buying strategy. Probably, this is due to the fact that in these strategies the amount of each stock to sell is highly dependent on the confidence level of all falling down stocks which are in the portfolio at the moment. In fact, if there is a case when several stocks predicted as "falling", it is rather difficult to line up such confidences for them, to sell all of them or at least more than 50% of their quantity.

Real SVM prediction. The training period for the SVM was chosen one year starting from the beginning of 2011 year and till the end of it. The prediction phase started after the end of year 2011, so the process of buying and selling stocks was during the whole 2012-th year, around 250 days. As it was mentioned already in the previous sections, SVM prediction gives rather small amount of classes with "rising up (1)" or "falling down (-1)" labels, even though the price difference parameter was only 1 percent. Moreover, having the price difference parameter more that 1 percent, prediction of SVM gives only "0" labels for all the stocks during the whole period (all the predictions have the value of zero). It is possible to continue the experiment with price difference parameter of zero and have a high accuracy as in Table 1, but one of the main goals was minimizing the risks and this parameter mostly influences on it.

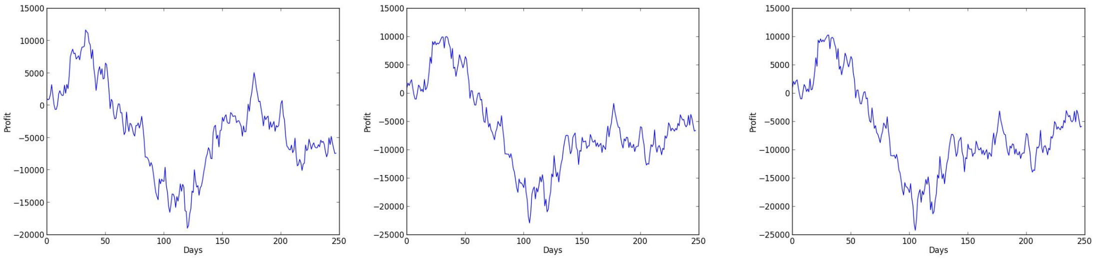
One of the possible reasons which can result to such a behavior is also the fact that the feature space is very large - 5504 dimensional (number of words) and the number of examples considered are only - 253 instances (number of days). Probably, there is not enough information for the classifier to make a correct prediction. However, the results that have been obtained are very interesting because when **sellAll** strategy is used as a selling strategy, all 3 cases show almost the same graphic of the profit. It can be seen on the plot 6, where the profits of each buying strategy with respect to **sellAll** strategy are depicted. All three graphics are similar and have a constant profit after the 50-th day till the end of the whole process. The reason is, that after this day, there is no ”rising up (1)” labeled stocks in the prediction which SVM gives. Thus, none of the buying strategies makes a decision to buy at least one stock. In addition, **sellAll** strategy in this case just empties the portfolio by selling all the current stocks which are in the portfolio. After that, each new day none of the stocks are bought and none of the stocks are obviously sold, thus the value of the 50-th day's profit stays till the end of whole the period.

However, these plots prove the theory that **buyTopKRelConf** gives more profit than **buyRelConf** or **buyEqually**. According to the profits of the last day on Fig.6, in each case, this fact can have place, because **buyTopKRelConf** has reached the value of 8000 points when the rest two buying strategies could reach only 7000 and 6000.



(a) buyEqually (b) buyRelConf (c) buyTopKRelConf

Figure 6: Profits of sellAll strategy with respect to the buying strategies



(a) buyEqually (b) buyRelConf (c) buyTopKRelConf

Figure 7: Profits of all the rest strategies with respect to the buying strategies

Concerning the rest of the selling strategies, the situation is almost the same. Each buying strategy with any combination of selling strategy, except **sellAll**, gives exactly the same profit graph. To understand it more clear we can have a look at the plot 7 where each graph represents the profit of a certain combination of the buying strategy with any selling strategy, except **sellAll**. As in a previous section, here all 15 combinations of strategies are divided into 3 groups.

Even though all three graphs illustrate the same behavior, there is a tiny difference between them. This small difference again proves the theory that **buyTopKRelConf** gives more profit at the end of the period, in comparison to **buyRelConf** and **buyEqually**. Although all the plots show a negative return at 250-th day, **buyTopKRelConf** gives less lose of initial budget at the end of the period and Table 3 proves it.

Strategy	Profit (points)	Profit (%)
buyEqually	-7389.34	-7.3%
buyRelConf	-6584.22	-6.5 %
buyTopKRelConf	-5864.36	-5.8%

Table 3: Comparison of profits of buying strategies

Since all the graphs in each group of strategies are absolutely the same, it is necessary to look more deeply into the portfolio to find out what kind of stocks and their amount of certificates, each combination of strategies buys and sells every day. The result of the analysis is illustrated in the Fig.8, where each buying strategy is depicted with two pie charts representing the portfolio at the beginning of the period from the left and portfolio at the end of the period from the right. In all the cases portfolio was filled by same stocks: **NYSE:NOV, NYSE:HAL, NYSE:APC, NYSE:BHI**. The only difference is in the amount of each stock in each combination of strategies. As it is visible from the charts, the amount of the stocks which are in the portfolio in the beginning and in the end of the period, in each strategy, are almost the same. That means since the portfolio was filled with a certain amount of stocks at the beginning of the period, it almost didn't change during the whole period. The reason of such a behavior is that there was not so many "falling down (-1)" labeled classes in SVM prediction during the whole period. Even those stocks that were predicted as decreasing ones were not in the portfolio at the moment. As a result, none of selling strategies could make a decision to sell any of the stocks which were in the portfolio, therefore these stocks remained there till the end of the period. That's why these graphs on the Fig.7 are not like a constant line because the portfolio was not empty and they show the profit as a movement of the prices of the stocks in the portfolio. Table 4 shows the prices of these stocks at the beginning and at the end of the period and their price improvement between these two dates.

Stock	Price at 03.01.2012	Price at 27.12.2012	Increased/ Decreased (%)
NYSE:NOV	70.87	66.97	-5.5%
NYSE:BHI	51.02	40.07	-21.4%
NYSE:APC	78.65	73.77	-6.2%
NYSE:HAL	34.15	34.60	1.3%

Table 4: Comparison of the prices of the stocks in portfolio

Even though all these stocks give rather good profit during the first 40 days and plots 6 and 7 prove that, not all of them remained "rising" at the end of the period. According to the Table 4 only **NYSE:HAL** was not "falling" stock at the end of the period. But unfortunately this stock didn't have so high confidence as **NYSE:APC** had. Therefore, in case of **buyRelConf** and **buyTopKRelconf** around 80% of the budget was spent on **NYSE:APC**.

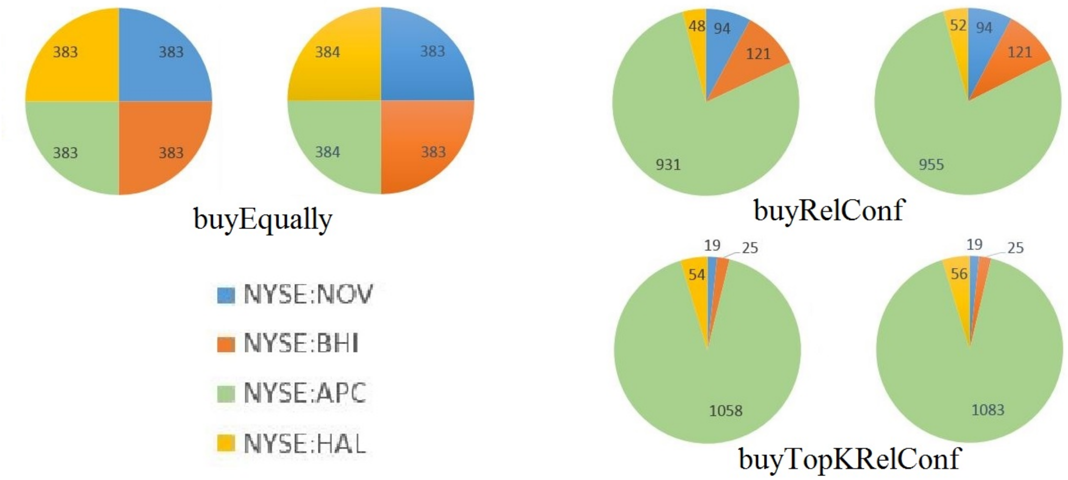


Figure 8: Portfolio of the buying strategies in the beginning and at the end of the period

6 Conclusion

Different scenarios have been explored in order to analyze gathered information. A number of different experiments have been done, such as finding the value of **Top K** in the strategies, finding the most suitable price difference parameter, generating SVM and random prediction and then analysis of all possible combinations of different strategies.

Regarding the value of **Top K**, experiments have shown that portfolio reaches the highest profit at the value between 4 and 10. And this fact seems logical since it is more reasonable to spend money only in 6 or 7 stocks with high confidence than in 20 or 30 stocks with lower confidence.

Comparing to achieved accuracy of SVM prediction of around 70% in [5] and [2], the accuracy we obtained is not so impressive. The reason is that we used price difference parameter as 1%, which showed the best results, while [5] and [2] used 0% and 0.5% respectively. Another reason of the low accuracy may also be the fact that there was not so much information for SVM to make more correct predictions. The training phase of SVM only for one year also plays an important role and maybe after training it for more than 5 years it will give more relevant prediction.

Even though, that SVM prediction didn't give so effective results, some of the theories were proved practically. According to all of the gathered statistics, among all buying strategies, **buyTopKRelConf** could be considered as the strategy that gives the highest profit. It was shown both by the random prediction and by the real SVM prediction. Concerning the selling strategies, in case of the real SVM prediction it was impossible to make any conclusions. But if we take into consideration the case of random prediction, it can be summarized that **sellAll** was always superior in its group. In contrast **sellRelConf** and **sellTopKRelConf** gave always the worst results. In addition, random prediction has proved that even having a classifier with 20% of correct prediction it is enough to have a positive profit at the end of a certain period.

For the future research, it is possible to extend the system by adding more advanced strategies, which will provide higher return. Additionally, it would be better to increase the training phase of SVM to obtain higher accuracy with higher price difference parameter. Determining the best parameter for the $MainBudget$ and $TheStockpile$ is also essential in order to maximum ensure the portfolio from the cases of loss.

References

[1] Kyoung-jae Kim and Ingoo Han. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, 2000.

[2] Rama Bharath Kumar, Bangari Shravan Kumar, and Ragiri Shiva Sai Prasad. Financial news classification using svm.

[3] Yuling Lin, Haixiang Guo, and Jinglu Hu. An svm-based approach for stock market trend prediction. In *IJCNN*, pages 1–7. IEEE, 2013.

[4] Phichhang Ou and Hengshan Wang. Prediction of stock market index movement by ten data mining techniques, 2009.

[5] Ms. D. Preetha and Mrs K. Mythili. Kenerl based svm classification for financial news.

[6] Tong-Seng Quah and Bobby Srinivasan. Improving returns on stock investment through neural network selection, 1999.

[7] R. Rosillo, D. de la Fuente, and J. A. L. Brugos. Forecasting s&p500 index movement with support vector machines.

[8] Rafael Rosillo, Giner Javier, Puente Javier, and Borja Ponte. Different stock market models using support vector machines, 2013.

[9] Ting-Fan Wu, Chih-Jen Lin, and Ruby C. Weng. Probability estimates for multi-class classification by pairwise coupling. *J. Mach. Learn. Res.*, 5:975–1005, December 2004.

[10] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1):1–37, December 2007.