FORECASTING STOCK INDEX MOVEMENT DIRECTION WITH *CPL* LINEAR CLASSIFIER

Jerzy Krawczuk

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: Stocks, indexes, commodities, and precious metals price prediction is a difficult task where many approaches are used: traditional technical analysis, econometric time series or modern data mining techniques. One particular data mining technique - linear classifier is described in this article. Prediction based on linear classifier is done using current market state, which can be described by various data sets (attributes, features). The simplest form of this model could use data from yesterday's price movement. Advanced models are using more historical price movements. Very advanced models include various historical price movements for indexes from other countries and other instruments like currencies, commodities, etc. Using more features requires extended time to estimate model parameters. We build the linear classifier models by the minimisation of a convex and piecewise-linear function which is very efficient comparing to other functions. Computational costs for building the model are similar to linear programming. We also use feature selection method called RLS. Those techniques allow us to explore data with many features. Four scenarios are considered, in each scenario a different amount of market data is used to create a model. In the simplest scenario only one day's change in price is taken, in the most complicated one 421 historical prices of 43 different instruments are taken. Best results were achieved by using middle range of 52 attributes. In this scenario, the model was right 53.19% times. Meaning the directions of daily change in S&P500 index (up or down) were predicted correctly. This doesn't seem a lot, but if those predictions would have been used for investing, they could produce a total profit of 77% in the tested time period from November 2008 to March 2011 (2 years 4 months), or an average of 28% per year.

Keywords: market forecast, market prediction, linear classifier, convex and piecewise-linear function

1. Introduction

The first well known book written on analysis of the stock market was "Confusion of Confusions (1688)" by Joseph de la Vega, who described the way the Amsterdam

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 7, pp. 47-58, 2011.

Stock Exchange worked and gave some hints for price analysis. In Asia during early 18th century, Homma Munehisa described the basics of candlestick techniques [20], which are nowadays a popular charting tool. Price chart analysis techniques, otherwise known as technical analysis [9], are very popular, and are used by traders on a daily basis. It focuses on searching for repeatable patterns in price charts, and on looking at some statistical indicators.

A more modern approach of describing the behaviour of market prices is known as econometric time series analysis [12]. The classic models assumes that current value of price is correlated with previous values (prices are autocorrelated). A stock's price is described as a linear equation of it is own previous values, such a model is called autoregressive. Box and Jenkings [6] describe the methodology to best fit such a model to data for a purpose of forecast. Different class of time series models are used for predicting not the expected value of process but the standard deviation of prediction. Main groups of such models are *ARCH* [11] and *GARCH* [5] also known as heteroskedasticity models. Those models play important role in risk analysis. Robert Engle work on time-varying volatility was awarded with Nobel price in 2003.

Data mining techniques that are developing quickly in recent years, are also being used for predicting market prices. They can be divided into models inspired by nature (neural networks [2], genetic algorithm [21]) and linear models (*SVM* [7], *CPL* [3]). In this article, linear models are presented with more detail, and we explain in an experiment how one such model can be used for market prediction, in this case the next day's move of S&P500 index. The model is built from one year of historical data, and then it is used to make predictions over the next half year. After each half year the model is rebuilt. Results are concluded at the end, and directions for future research are discussed.

2. Data mining techniques inspired by nature

2.1 Neural network

In general, a neural network or artificial neural network is a computer model whose architecture is inspired by human brain. It is build from elements called artificial neurons that process the information in a basic way. It transforms many input signals (real numbers) into one output number. For example if transformation is linear, and output is equal either 0 or 1, depending on some threshold, such *NN* is called a *perceptron* and neuron is a *linear classifier*. The learning power of *NN* lies in hierarchical structure of neurons which is called network. They can model complex non-linear relationships between inputs and outputs.

Nowadays neural networks (ANNs) have been popularly applied to finance problems including stock index prediction. Tokyo stock index was predicted by Kimoto [16] and Mizuno [19], Istanbul Stock Index by Egeli [10]. Other authors use different network architecture (topology), different methods of training and testing. We can find summary of such approaches in Zekic [24]. Authors claims that neural networks gives better results then buy-and-hold strategy.

2.2 Genetic algorithm *GA*, genetic programming *GP*

This group of methods is using an analogy of evolution processes in order to solve optimisation or search problem. With a help of evolution processes it transforms a set of population (mathematical objects *GA* or computer programmes *GP*) into a new population. Two key mechanism of biological evolution must be mapped in such transformation. First is a natural selection mechanism: those individuals from a population who can solve problem in most efficient way should have bigger chances to survive and reproduce in their environment.The second mechanism is a genetic drift which allow random changes in new population. Individuals are defined by they chromosomes. Transformations between populations is done through changes in chromosomes using genetic operations like inheritance, mutation, selection and crossover. Each population is called a generation. Usually the algorithm stops when either a maximum number of generations has been produced, or a best individual has a satisfactory fitness level.

Since the evolutionary algorithm is a general approach for solving optimisation problem, it can be used in many different ways for purpose of predicting the market. Three most common approaches are:

- finding optimal parameters for a model, usually using the technical analysis,
- feature selection usually with neural network,
- discover trading rules.

First approach is very common in the case of genetic algorithms. Very popular trading platform called MetaTrader is using the genetic algorithm for the purpose of finding optimal parameters for tested investing strategies. User can define a strategy in MQL4 programming language and he can choose from many built-in technical analysis indicators. Such strategy always have some parameters that need to be set before we can start using them. Strategy can be executed on historical data with different values for different parameters. Built-in strategy tester allow for many executions with different parameters. It can do so by Simple Search (searching whole parameters space) or Genetic Algorithm. Some users claimed that with Genetic Algorithms it is possible to find solutions much faster than using other algorithms [14].

Second way of usage - the feature selection is also popular. If we want to describe the market with many features and use for example neural network to build a model, as a first step we may want to select only relevant features. But searching for all $2^N - 1$ subspaces is not possible for to many features N. Many authors argue that genetic algorithms can find very good subspace with reasonable time [23].

Third approach is to use genetic programming, where individuals in populations are represented by computer programmes. Each programme represents simple open or close price, or mathematical operator like add or technical analysis indicators like moving average [21]. From such elements the genetic algorithm is trying to build an optimal formula for the trading signal.

3. Linear models

Bobrowski [3] define linear classifier $LC(w[n], \theta)$ as decision rule:

$$LC(w[n], \theta) = \begin{cases} if\ w[n]^T x[n] \ \geqslant \ \theta,\ then\ x[n]\ is\ located\ in\ class\ \omega^+ \\ if\ w[n]^T x[n] \ < \ \theta,\ then\ x[n]\ is\ located\ in\ class\ \omega^- \end{cases} \tag{1}$$

where $w[n] = w[w_1, w_2, ..., w_n]^T$ is a vector of weights $w_i \in R^1$ and θ is a threshold ($\theta \in R^1$). The creation of predictive rules (1) requires the calculation of the parameter values $w[n]$ and θ. Those parameters can be determined on the basic of learning sets G^+ and G^- containing examples of feature vectors $x_j[n]$ from class $\omega^+ (j \in J^+)$ and from class $\omega^- (j \in J^-)$.

$$G^+ = \{x_j[n] : j \in J^+\}\ and\ G^- = \{x_j[n] : j \in J^-\} \tag{2}$$

In practice it is not always possible to find such $w[n]$ and θ where all feature vectors $x[n]$ are correctly classified. This is not always desirable as well, because of the danger of over-fitting to the data sets (2). The optimal parameters $w^*[n]$ and θ^* of the classification rule (1) can be determined in many ways. *SVM* and *CPL* approaches are introduced in this paper.

3.1 Support vector machine - SVM

SVM approach [7] is to find such optimal parameters $w^*[n]$ and θ^* that represents the largest separation, or margin, between objects from sets G^+ and G^- (2). Such margin could be represented as a two parallel hyperplanes (fig.1) that are in maximum distance but still separating sets G^+ and G^-. They can be written as $wx - \theta = 1$ and $wx - \theta = -1$. It can be proven that a distance between these two hyperplanes is $2/||w||$. Optimisation problem could be defined as finding such optimal parameters $w^*[n]$ and θ^* that minimize $||w||$ and satisfy below constraints:

$$\begin{cases} w[n]^T x_i[n] - \theta \geqslant 1, \; for \;\; x_i \in G^+ \\ w[n]^T x_i[n] - \theta < 1, \; for \;\; x_i \in G^- \end{cases} \tag{3}$$

This problem can be solved by standard quadratic programming techniques. *SVM* approach described above has many extensions. Two main extensions are:

– soft margin: allow for misclassification of the data [7],
– non-linear classification: applying the kernel trick [1].

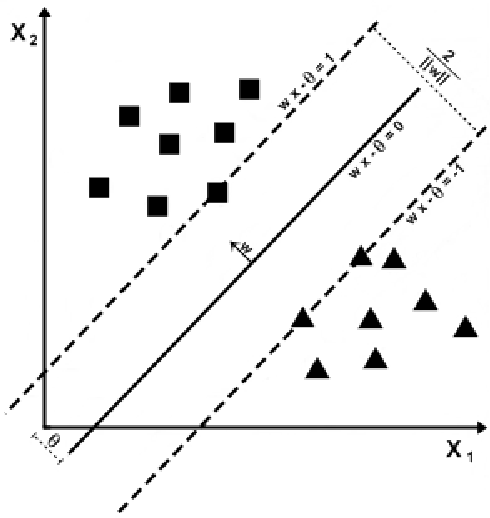


Fig. 1. Optimal hyperplane w constructed with two support vectors on the margin of sets G^+ and G^-. Source: own elaboration

One of the earliest studies on financial forecasting using support vector machines could be found in [15]. Author compared *SVM*s with other popular data mining techniques, including case-based reasoning and backpropagation neural networks. *SVM* outperformed other methods in prediction of future direction of Korea stock index. Results were in range of 50%-57%, depends on used parameters. In other study [13] author predicted a weeks direction of change for a Japanese index NIKKEI with as high as 73% hit ratio for SVM.

3.2 Convex and piecewise-linear penalty function - CPL

Other way of finding optimal parameters $w^*[n]$ and θ^* of the classification rule (1) is proposed by Bobrowski [3] [17]. He define a convex and piecewise-linear (*CPL*) criterion functions in the below manner:

$$\phi_j^+(w[n],\theta) = \begin{cases} \theta + 1 - w[n]^T x_j[n] & if \;\; w[n]^T x_j[n] < \theta + 1 \\ 0 & if \;\; w[n]^T x_j[n] \geqslant \theta + 1 \end{cases} \tag{4}$$

$$\phi_j^-(w[n],\theta) = \begin{cases} \theta - 1 + w[n]^T x_j[n] & if \;\; w[n]^T x_j[n] > \theta - 1 \\ 0 & if \;\; w[n]^T x_j[n] \leqslant \theta - 1 \end{cases} \tag{5}$$

And the perceptron criterion function $F(w[n],q)$ as the weighted sum of the penalty functions (4) and (5):

$$\Phi(w[n],\theta) = \sum_{j \in J^+} \alpha_j \phi^+(w[n],\theta) + \sum_{j \in J^-} \alpha_j \phi^-(w[n],\theta) \tag{6}$$

where non-negative parameters α_j represent prices linked to particular feature vectors $x_j[n]$. The minimization of the criterion function $\Phi(w[n],q)$ (6) allow us to find the optimal parameters $w[n]^*$ and q^* of the prediction rule (1).

3.3 Feature selection method *RLS* (Relaxed Linear Separability)

It is *CPL embeded* feature selection method. Relaxed linear separability (*RLS*) methodology was introduced in [4][17]. It is defined by additional costs γ related to particular features x_i added to the penalty function (6) :

$$\Psi(w[n],\theta) = \Phi(w[n],\theta) + \lambda \sum_{i \in I} \gamma_i |w_i| \tag{7}$$

where λ is the cost level, and $I = \{1,...,n\}$. In accordance with the RLS method, a gradual increase of the cost level λ value in the criterion function (7) allows successive reduction of features x_i. In the result a descended sequence of feature subspaces can be generated. The quality of each subspace is measured and best subspace is selected. Quality measure is a classification accuracy calculated by the leave-one-out methodology. Each feature vector $x_j[n]$ is classified by the linear classifier (1) build on all other vectors except one which is classified. This method allow to reduce bias of the classifier accuracy estimation.

4. Experiment

The *CPL* linear classifier with *RLS* feature selection method was used in the experiment. One day move of S&P500 US stocks index was predicted. In this approach we do not predict the exact tomorrows value for the index, we predict direction of change (either the index will move down or up). Forecast task is defined as a classification approach [8] not a regression.

Daily market data (open and close prices) was used, with data set starting from November 2007 till end of February 2011. Each day was described by the feature vector $x[n]$ and class ω. Vector $x[n]$ could be assigned to one of two classes. To class ω^+ if in the next day index rose, and ω^- if index fell. Four scenarios were used, in each scenario feature vector $x[n]$ describing current market situation was constructed in a different way. Each scenario have different number of features, starting from only 1 through 10, 52 to 421. For the purpose of this article we call them Simple (1), Normal (10), Big (52) and Huge(421). All features used in the Simple model was used in Normal, all used in Normal was used in Big, and all in Big was used in Huge. The features are:

– Simple: only overnight gap for SPY (change between yesterday close and today open).
– Normal: only historical prices for SPY:
  - open price,
  - gap, percent change from yesterday close,
  - daily change, percent change from yesterday open,
  - 2 days change, percent change from open to open,
  - 5 days change, percent change from open to open,
  - yesterday daily change, change from open to open at yesterday open,
  - 2 days back daily change,
  - 9 days moving average (close prices),
  - 12 days moving average (close prices),
  - 26 days moving average (close prices).
– Big: all features used for SPY in *normal* scenario, and also gaps of other 41 instruments plus VIX previous day close.
– Huge: all 10 features for all 42 symbols plus VIX level.

VIX is the Volatility Index. It measures the market's expectation of near term volatility based on options prices of S&P500 stock index.

Calculations were done using training and test data sets. All data was divided into 5 groups of corresponding training and test data sets in a way showed in (2).

The training set was build using 252 features vectors, one vector for each day. This is approximately 1 year of data. The test set was build using 126 vectors representing days following the training period. Such approach can by used by investor to trade on real markets. Starting point would be November 2008. At this time investor could use 1 year historical data and build a decision rule (1) to use it for a next 6 months.

Lets consider *Big* scenario to check what kind of results investor would see. After building a *CPL* linear classifier (1) with *RLS* feature selection procedure at November 2008 results would be as follow:

- *RLS* method will choose only 11 features,
- *CPL* linear classifier on those 11 features will have accuracy of classification measured by leave one out methodology of 67.98%.

If investor would be satisfied with those results on historical data, he could start using the model to trade on real markets. If so, in the next half year he would be successfully in 56.35%[2] cases predicting day index move (up or down). If he can bet on both market up and down his profit would be as high as 22.47%. Making profit on moving down market could be achieved by investing in futures contracts like E-mini[3], or fund like SPY[4].

After half year the investor could decide that his model is out of date and it has to be rebuilt using more up-to-date data. Construction of the new model could be again done using 1 year historical data, this time going back to May 2008. New model would use 42 features and have a little higher accuracy of 69.57% measured on training set, but in the next 6 months it would not produce a profit. Days on good position would be only 48.41% that would transfer to lose of 7.10%. All results till beginning of March 2011 are presented in 2, results for different scenarios are presented in 4..

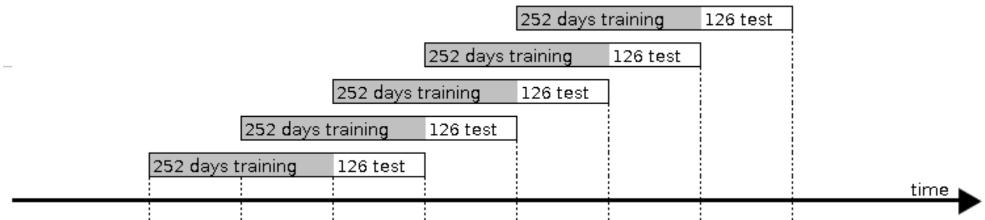
5. Conclusion

These results show that it is hard to achieve a better prediction of the next day's market price change then 50% which in fact does not give better results than throwing a coin. Never the less the best results of 53.91% could give some advantage to the investor. Such advantage could actually return an average profit of 15% in 6 months,

[1] In other words *ex post* error of prediction is equal 43.65%

[2] [http://www.cmegroup.com/trading/equity-index/us-index/e-mini-sandp500_contract_specifications.html]

[3] SPY - fund corresponds to the price and yield performance of the S&P 500 [https://www.spdrs.com/product/fund.seam?ticker=spy]



	Nov2007		Nov2008	May2009	Nov2009	May2010	Nov2010	Mar2011	
									Average
Used attributes			11	42	24	36	11		28,25
Training LOO			67,98%	69,56%	63,64%	70,36%	66,01%		67,89%
Test accuracy			56,35%	48,41%	59,52%	62,70%	42,55%		53,91%
P&L			22,47%	−7,10%	21,18%	47,29%	−6,36%		15,49%

Fig. 2. Moving window of building model on 1 year of training data and testing it on next half year of the data from Nov 2007 till Feb 2011. Source: own work

Table 1. Results for the CPL classifier with Relax method of features selection on different set of attributes

Model	Simple	Normal	Big	Huge
No of attributes	1	10	52	421
Attributes after features selection	1	3.6	24.8	92
LOO accuracy on training sets	55.20%	55.41%	67.51%	91.86%
Average accuracy on test sets	51.07%	50.84%	53.91%	48.93%
Average profit or lose on test sets	-1.72%	5.01%	15.49%	4.90%

giving a total of 77% profit in the entire time period tested (November 2008 - March 2011). It could be argued that such a result from using the *CPL* linear classifier for predicting the one day movements of a market index is promising.

The practical aspect of this research is actual investing in the market. We could have a bad decision from 1 model resulting in different levels of loses. A loss could be as low as 0.1% or as high as 3.0%. In both cases the model will be wrong, but the consequences of each loss differ greatly. The question that could be asked here is: can we account for this variation during optimisation? It seems that the *CPL* penalty function $\Phi(w[n], \theta)$ 6 could easy account for it by α_j parameters. More research in this are could be done in the future.

Other area of future research could be to look for explanation of observed differences between classification accuracy on training and test data sets. For example for biggest space with 421 attributes, accuracy measured by leave one out method was equal 91.86% but on test set it was only 48.93%. This difference get lower when less features were used. Probably it could be explained by the features subset selection bias [18][22], but more research need to be done to verify this hypothesis.

References

[1] Aizerman M., Braverman E., Rozonoer L., Theoretical foundations of the potential function method in pattern recognition learning, Automation and Remote Control 25, 1964.

[2] Bishop M., Neural networks for pattern recognition, Oxford University Press, 2005.

[3] Bobrowski L., Eksploracja danych oparta na wypukłych i odcinkowoliniowych funkcjach kryterialnych, Wydawnictwa Politechniki Białostockiej, 2005.

[4] Bobrowski L., Łukaszuk T., Feature selection based on relaxed linear separability, Biocybernetics and Biomedcal Engineering, Volume 29,Number 2, 2009, pp. 43-59.

[5] Bollerslev T., Generalized Autoregressive Conditional Heteroskedasticity, Journal of Econometrics, 31, 1986, pp. 307-327.

[6] Box G.E.P, Jenkins G.M., Analiza szeregów czasowych, Państwowe Wydawnictwo Naukowe, 1983.

[7] Cortes C. Vapnik, V., Support-Vector Networks, Machine Learning, 20, 1995.

[8] Duda O.R., Hart P.E., Stork D.G., Pattern Classification, J. Wiley, New York, 2001.

[9] Edwards R.D., Magee J., Technical Analysis of Stock Trends, AMACOM, 7th edition, 1997.

[10] Egeli B., Ozturan M., Badur B., Stock Market Prediction Using Artificial Neural Networks, Proceedings of the 3rd International Conference on Business, 2003.

[11] Engle R.F., Autoregressive Conditional Heteroskedasticity with the Estimates of the Variance of U.K. Inflation, Econometrica, 50, No. 4, 1982, pp. 987-1007.

[12] Hamilton J.D., Time Series Analysis, Princeton University Press, 1994.

[13] Huang W., Nakamoria Y., Wangb S.Y., Forecasting stock market movement direction with support vector machine, Computers & Operations Research Vol. 32, Issue 10, 2005, pp. 2513-2522.

[14] Khatimlianskii A., Genetic Algorithms vs. Simple Search in the MetaTrader 4 Optimizer, [http://articles.mql4.com/361]

[15] Kim K.J., Financial time series forecasting using support vector machines, Neurocomputing Vol. 55, Issues 1-2, 2003, pp. 307-319.

[16] Kimoto T., Asakawa K., Yoda M., Takeoka M., Stock market prediction system with modular neural network, Proceedings of the International Joint Conference on Neural Networks, 1990, pp. 1-6.
[17] Krawczuk J., Bobrowski L., Short term prediction of stock indexes changes based on a linear classifier, Symulacja w badaniach i rozwoju, Vol. 1 nr 4/2010.
[18] Miller A., Subset selection in regression, Chapman & Hall/CRC, 2002.
[19] Mizuno H., Kosaka M., Yajima H., Komoda N., Application of Neural Network to Technical Analysis of Stock Market Prediction, Studies in Informatic and Control, Vol. 7, No. 3, 1998, pp. 111-120.
[20] Nison S., Japanese Candlestick Charting Techniques, Second Edition, Prentice Hall Press, 2nd edition, 2001.
[21] Poli R., Langdon W.B., McPhee N.F., A Field Guide to Genetic Programming, Lulu.com, 2008.
[22] Singhi S.K., Liu H., Feature Subset Selection Bias for Classification Learning, Proceedings of the 23 rd International Conference on Machine Learning, Pittsburgh, PA, 2006, pp. 849-856.
[23] Yang J., Honavar V., Feature subset selection using a genetic algorithm, Intelligent Systems and their Applications, IEEE, 1998.
[24] Zekic M., Neural Network Applications in Stock Market Predictions - A Methodology Analysis, Proceedings of the 9th International Conference on Information and Intelligent Systems, 1998, pp. 255-263.

PROGNOZOWANIE KIERUNKU ZMIANY INDEKSÓW GIEŁDOWYCH ZA POMOCĄ KLASYFIKATORA LINIOWEGO TYPU CPL

Streszczenie Prognozowanie cen akcji i wartości indeksów giełdowych jest zadaniem trudnym, gdzie używanych jest wiele technik takich jak: analiza techniczna, ekonometryczne szeregi czasowe, techniki eksploracji danych. Artykuł ten przedstawia jedna z metod eksploracji danych - klasyfikator liniowy. Klasyfikator ten w przeprowadzonym eksperymencie został użyty do prognozowania wartości indeksu giełdy amerykańskiej. Prognozowanie takie oparte jest o dane opisujące obecny stan giełdy. Stan giełdy można opisać różną ilością danych (atrybutów, cech). W najprostszym przypadku może to być tylko jednodniowa zmiana ceny prognozowanego indeksu. W bardziej rozbudowanym modelu można użyć wielu cen historycznych. W modelu jeszcze bardziej rozbudowanym można użyć danych z innych giełd, kursów walut, cen towarów jak np. ropa. Użycie dużej ilości danych wymaga dłuższego czasu obliczeń parametrów modelu. W prezentowanym podejściu klasyfikator liniowy budowany jest w oparciu o minimalizację wypukłej i odcinkowo-liniowej funkcji kryterialnej. Metoda ta jest bardzo wydajna o koszcie zbliżonym do programowania liniowego. Dodatkowo użyta została metoda selekcji cech RLS. Techniki te pozwoliły na efektywną eksplorację danych o wielu wymiarach. W artykule przedstawiono cztery scenariusze o różnej ilości danych opisujących giełdę. W najprostszym użytku tylko jednej danej, w najbardziej rozbudowanym 421 danych o 43 instrumentach finansowych. Najlepsze wyniki uzyskano dla pośredniego modelu o 52 cechach, w którym model przewidział prawidłowo 53.19% kierunków dziennych zmian indeksu S&P500. Otrzymany wynik nie wydaje się być wysoki, jednak gdyby inwestowano w indeks zgodnie z modelem zysk z takich inwestycji wyniósłby 77% w okresie od października 2008 do marca 2011, dając średnio 28% zysku rocznie.

Słowa kluczowe: klasyfikator liniowy, prognozowanie giełdy, funkcje wypukłe i odcinkowo-liniowe