See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/307802535

Optimized Stock market prediction using ensemble learning

Conference Paper · October 2015

DOI: 10.1109/ICAICT.2015.7338559

CITATIONS
7

READS
249

1 author:

Muhammad Asad
Academia Sinica

5 PUBLICATIONS 9 CITATIONS

SEE PROFILE

All content following this page was uploaded by Muhammad Asad on 21 April 2019.

The user has requested enhancement of the downloaded file.

Optimized stock market prediction using ensemble learning

Muhammad Asad
College of Electrical & Mechanical Engineering, NUST
Rawalpindi, Pakistan
muhammad.asad13@ce.ceme.edu.pk

***Abstract*— Researchers of various fields have always been interested in devising a fault-proof method for the prediction of stock market. Extensive research has been done using Machine Learning Algorithms like SVM, to successfully predict the stock activity in the market using machine learning algorithms mainly Support Vector Machine (SVM). In this paper a scenario in which a portfolio trading strategy is formulated using machine learning algorithms. The strategy will be considered profitable by judging its ability to identify stock indices accurately and consistently, proposing positive or negative returns and in the end it should use a learned model to produce a preferred portfolio allocation. Some of the Technical Indicators like Multiple Regression Analysis (MRA) and different data clustering techniques are used as input to train the system. The learned model is constructed as weighted support vector machine (SVM) classifier, Relevance Vector Machine, random forest classifiers and Multiple Layer Perceptron (MPL). The decision value would be chosen using a majority voting mechanism. The ensemble learning is augmented by a boosting meta-algorithm and feature selection is performed by a supervised Relief algorithm. Stocks listed in Istanbul Stock Exchange (ISE) in Turkey are used to evaluate the performance of the system. A comparison of results obtained using ensemble committee and those using other approaches would show that the ensemble approach has a lower error rate and generates fewer but compact rules.**

***Keywords*—Stock market, Ensemble learning, Stock market prediction, Feature selection.**

I. Introduction

Predicting the trends in Stock market is typically supposed to be a very challenging task due to the uncertainties involved in the market values. Bulls and Bears mainly involves many factors including supply and demand, government influences, general economic conditions, political influences and international transactions. The evident complexity of the problem covers way for the importance of intelligent prediction paradigms. People are using intelligent algorithm and techniques to predict the stock exchange behavior to predict the bulls and bears and making trade decisions. This makes obvious in intelligent problem solving is an important factor in intelligent prediction paradigm.

Our chosen approach to this complex financial forecasting problem is to train an ensemble of classification models on a subset of labeled financial data that are categorized as members of the sets 1+ or 1- according to whether there was a positive or negative shift in stock price from an initial time to a subsequent time. This work offers two important contributions to the field of financial forecasting. The learned model is constructed such that feature selection is conducted a priori and is a fully automated process; this allows the model itself to “discover" which parameters it believes are important to effective prediction rather than being forced to accept human designated explanatory variables. This gives the ensemble an attribute of adaptability in the sense that it may reformulate its relevant parameters according to the GICS or according to location in time. Second, the construction of the ensemble incorporates a probabilistic ranking component that approximates a level of confidence associated with each prediction. Within the financial literature, it is typically the case that portfolios are formulated according to a scoring function that is intended to capture the differences of desirable and undesirable stocks. The capability of the ensemble is similar in concept, yet rather than providing an absolute hierarchy of stock preferences, a probabilistic measure of the desirability of the stock is returned. In uncertain games such as portfolio investment, there are clear benefits to possessing a probabilistic confidence criterion.

II. Methodology And Analysis

In this paper Stocks are categorized by time sequence data sets comprising of procedural variables that imitate market situations in a preceding time interlude, which are used to create binary classification decisions in subsequent interludes.

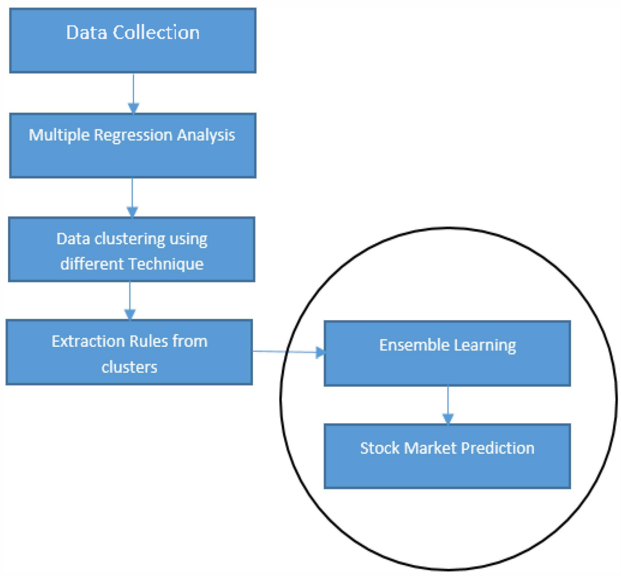

Fig. 1. Purposed Methodology

The learned model is built as a weighted support vector machine classifier, random forest classifiers, a relevance vector machine classifier and Multiple Layer Perceptron (MPL). There are two reasons for choice of these Algorithms : first, there is noticeably little study in financial time-series foretelling that employs learners outside neural networks and clustering algorithms, and this structure deals a feasible substitute; second, this selection integrates an array of methods that have both theoretically optimal classification properties and high empirical success rates in areas separate of economics, in addition to providing a combination of parametric and non-parametric models.

.

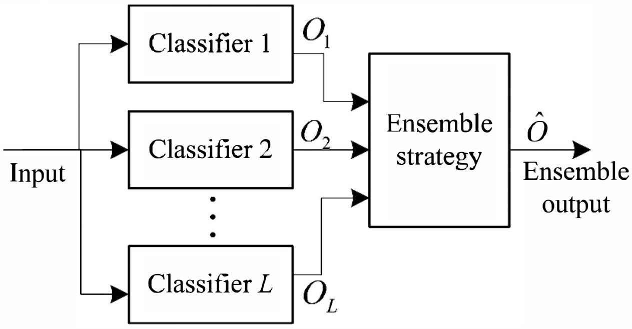

Fig. 2. Ensemble Classifiers

The ensemble committee is amplified by a heightening meta-algorithm and feature selection is achieved by a supervised Relief algorithm. This unit delivers a fleeting introduction to the classification algorithms amalgamated into the ensemble. The section is envisioned for individuals with little knowledge of learning algorithms. See that this section is envisioned just to present the constituent models and to identify how they create a cohesive whole. The constituent algorithms are offered in the order (1) support vector machine classifier, (2) relevance vector machine classifier, (3) random forest classifier, and (4) Multiple Layer Perceptron (MPL). The meta-algorithm of boosting" is also dignified here. In the field of financial trading it is of some earnestness to build models that may be learned and positioned competently, yet must also be comparatively strong to the integrally stochastic nature of stock returns [1]. By means of these critical ideas, we can encourage an ensemble of this form by denoting the alterations between parametric and non-parametric procedures within machine learning [2]. Parametric models are beneficial in the sense that they are firm to learn and install, but characteristically mark strong expectations around the spreading of the data. Non-parametric models evade almost all previous hypotheses regarding the data,

But the difficulty of non-parametric learning procedures inclines to be bigger than that for parameterized learners. Since random forests and is non-parametric learners, and support vector machines and relevance vector machines are parameterized, the ensemble model offered here profits from the rewards of both styles, hitherto still upholds the aptitude to organize the constituent models independently.

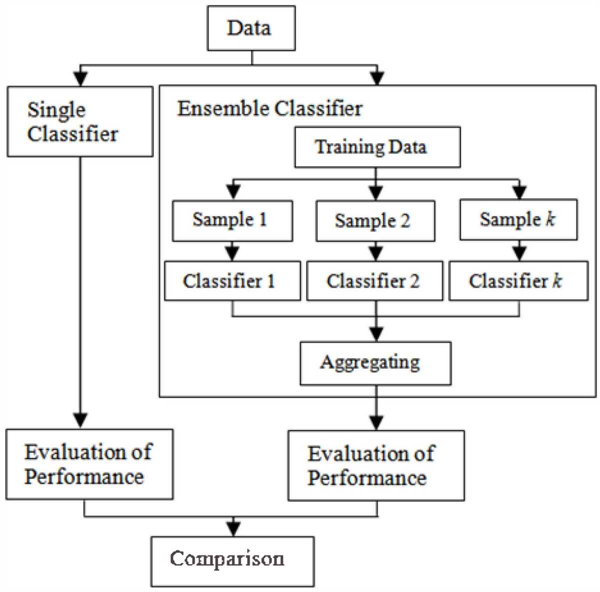

Fig. 3. Comparison of Single and Ensemble Classifiers

A. Artificial Neural Networks (ANN)

In this paper three layer feed forward technique has been used with ten neurons as an input, one each for a technical parameter and a single neuron as an output layer to show the predicted result. For the purpose of updating weights, an adaptive gradient descent is used along with the tangent sigmoid as a transfer function. At the end side output of the model is produced with continuous value which keeps on signifying the predicted values of the stock index. Adaptive gradient descent is used because it allows the change in the learning rate during training process which may provide improvement to the adaptive gradient algorithm. During the process of adaptive gradient descent initial network output and error are calculated followed by the current learning rate which is used to calculate the new weights and biases, first, the initial network output and error are calculated. The current learning rate is used to calculate new weights and biases at each epoch. Based on these new weights and biases each time and if the new error outstrips the old error by a given predefined ratio, the new weights and biases are cast-off [10]. In the similar way learning rate is also decreased. Otherwise, new weights and biases are kept and the learning rate is increased. The procedure ensures that the learning rate is increased only to the extent that the network can learn without large increases in error. This allows to get near optimum learning rate for the local environment. At the same time learning rate is increased as long as stable learning is guaranteed.

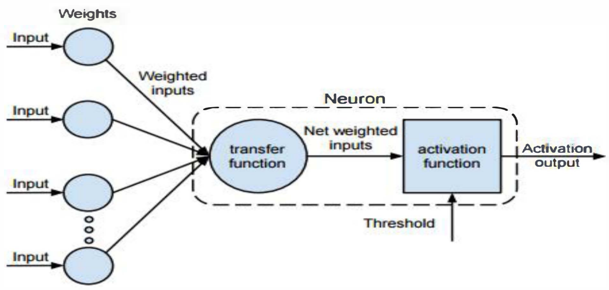

Fig 4. A neuron sums multiple weighted inputs from other neurons and applies it to an activation function

When it is too high to guarantee a reduction in error, it is reduced until stable learning continues. Design parameters of the model are the number of neurons and epochs used in the hidden layers. Inclusive number of testing are conceded out by wavering the parameter values.

B. Random Forest

An arbitrary forest categorizer is a taught collective of decision trees, i.e. the integral learners are "de-correlated" by increasing each tree on an arbitrarily selected subset of all the data trajectories and all the structures [3]. This collective gives out a decision function of the form,

$$f(\vec{x}) = \sum_{i=1}^{m} \frac{1}{m} f_i(\vec{x})$$

Where f_i is the decision function of the i^{th} tree in collective. In increasing an individual tree, the basic purpose is to lessen the contamination of the training data in the nodes subsequent of the candidate splits. There are numerous procedures of such contamination including the mis-categorization rate and entropy, though we designate here using the Gini index measurement, given by,

$$g(\vec{x}) = 1 - \sum_{c \in C} \mathbb{P}(y|\vec{x} = c)^2$$

Where C is the set of possible class labels and in the dual setting we have that C = {+1, -1}. It can be exposed that the minimalizing of Gini index at every node split is comparable to minimalizing the predictable error rate [9]. This specific impurity metric is selected for the decision trees because it gives almost cooperation between entropy and absolute error rate termed in its sensitivity to class likelihoods.

C. Support Vector Machine (SVM)

SVMs are kind of extreme boundary categorizers that pursues to find a supreme boundary hyper-plane to separate the classes, i.e., SVMs exploits expanse of hyper-plane from the neighboring training examples. Hyper-plane which is obtained is called the optimal separating hyper-plane (OSH) and the training data samples neighboring the supreme boundary hyper-plane are support trajectories. If the data is linearly divisible, a hyper-plane splitting the dual decision classes in the two attribute case can be signified as the following equation:

$$Y = w_0 + w_1x_1 + w_2x_2 \quad \ldots (1)$$

Where, Y is product, x_i are trait values, w_i are weights to be refined by the learning procedure. Similarly, supreme boundary hyper-plane are signified from the following equation termed with support trajectories:

$$y = b + \Sigma\alpha_i y_i x(i).x \ldots (2)$$

Where y is the class value of training sample x (i), the vectors x (i) are the support vectors, the vector x signifies a test example and · represents the dot product. In the given equation, b and αi are limits of the model that regulate the hyper-plane.

Ruling provision vectors and defining limits b and αi are correspondent to solve the linearly forced quadratic programming problem.

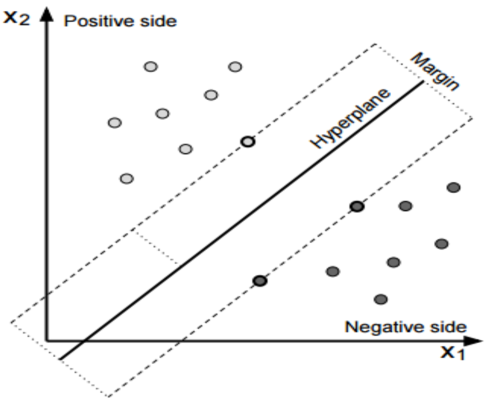

Fig. 5. SVM hyper-plane, in a two dimensional feature space

If no linearly detachable data is available, as in this situation, SVM alters the inputs into the high-dimensional feature space. This can be completed by using the kernel function as follows:

$$y = b + \Sigma\alpha_i y_i K(x(i), x) \ldots (3)$$

There are different diverse kernels available for producing the inner products for building machines with diverse sorts of Non-linear decision surfaces in the input space [4].

K(x; y) = (xy+1) n and the Gaussian radial basis function (RBF)

K(x; y) = exp $^{(-1/\delta 2(x - y)\ 2)}$ where n is the degree of the polynomial kernel and δ2 is the bandwidth of the Gaussian RBF kernel.

An exclusive aspects of Support Vector Machines are that they are resilient to the over fitting problem. It's since although the numerous old-fashioned NN models have been applied for the empirical risk minimization principle [10], SVM gears fundamental risk minimization principle. The previous pursues to minimalize the misclassification error or nonconformity from precise clarification of the training data, but latter hunts to minimalize an upper limit of simplification error [5].

D. Relevance Vector Machine

Relevance vector machine (RVM) assumes a form similar to the SVM, but is capable of providing a probabilistic interpretation to predictions. For target values y {0, 1} (remapping the input training labels from {+1, -1} is trivial), then predictions are of the form,

$$\hat{y}(\vec{x}) = \sum_{i=1}^{n} w_i k(\vec{x}, \vec{x}_i) + b$$

Whereas *b* is essentially analogous to the bias parameter in the SVM. In this case, the wi are constructed through an iterative optimization process where there is an initial assumption,

$$\mathbb{P}(\vec{w}|\vec{\alpha}) = \prod_{i=1}^{m} \mathcal{N}(w_i|0, \alpha_i^{-1})$$

And the αi is precision parameters corresponding to the w_i. Rather than taking the sign of the linear function, as was the case in the SVM situation, a classification decision is constructed by considering a logistic sigmoid function as follows:

$$\mathbb{P}(y|\vec{x}) \equiv \sigma\left(\sum_{i=1}^{n} w_i k(\vec{x}, \vec{x}_i) + b\right)$$

Such that, In particular, if P (y|x) > 4/5 then a classification decision of +1 is returned, whereas if the converse is true, then a classification decision of -1 is returned (after remapping into the original target- space). The threshold of 4/5 was selected by experimentation as delivering strong predictive results. This selection can be justified intuitively in some sense by the observation that high confidence is preferable in stock prediction, and that one is more likely to invest correctly when the probability of class membership is 0.80 rather than 0.50. The RVM is then capable of providing an interpretation that a given stock input has a particular class membership probability above the 0.80 threshold, which the user can incorporate into further analyses.

Summarily speaking, the RVM is an advantageous algorithm to use due to its probabilistic properties, and the value of the sigmoid function represents a degree of confidence that a given feature vector belongs to the class of stocks with positive returns from an initial quarter to a subsequent quarter.

E. Boosting Classifier Performance

Financial time-series foretelling agonizes from somewhat of a bad status, being founded on raucous data for which it is frequently obstinate to learn an operative model. Methods exist in machine learning to dismiss this issue and inside the possibility of this effort we implemented the technique of boosting for cultivating a foundation set of “weak learners" to produce a more accomplished forecaster.

The vital impression behind the boosting algorithm is that if the first classifier completes powerfully and precisely foretells the stock return consequence, then it is less significant for subsequent classification algorithms to also mark the identical precise forecasts, and consequently their influence on the extrapolation task becomes less expressive and less essential. A course for future study in this area would be in presentation to the detection of procedures for integrating a weighted objective function that more exceptionally poises the constituent models (consider for example an increase of Adaptive Boosting). In some sense of the word, this application of boosting is a “hacky" solution, trusting primarily on stimulus from a wholly streamlined version of AdaBoost and on an empirical assessment of modestly what works well. However, it was found experimentally that this procedure learns weighting coefficients of similar correctness to a wearing grid search of parameters on the testing data [6, 7]. Consequently, however the algorithm absences a hypothetical framework, its usefulness in practice validates its occurrence [8].

F. Technical Analysis (TA)

Technical analysis is a method which involves study of market action using previous values and trading trends with regards to predicting forthcoming price trends. TA considers that stock values relocate trends, and the detail which affects prices pass in the market over a predetermined stretch of time. TA challenges the long kept Efficient Market Hypothesis (EMH). EMH situations that market values track a random walk and can't be predicted based on their own past behavior. According to EMH, all information that enters the market affects the prices instantly. If the EMH were true, it would not be possible to make use of Artificial Intelligence methods to forecast the marketplace. Conversely, because of achievement of technical analysts within the financial world and a several studies appearing academics works effectively using Artificial Intelligence approaches to forecast the market, EMH is widely considered to be a null hypothesis right now. The technical analysts use procedural meters, which are mathematical preparations which give us clues concerning the trend of the marketplace. An example of a technical sign is the famous stochastic oscillator %K

$$\%K = (P(c) - P(l)) / (P(h) - P(l))$$

Where, P(c), P (h), and P (l) represents the lowest price, the highest price and the closing price respectively, of a security more than any given time period. Technical experts generally use a number of such judgment gained & indicators from knowledge to decide which design a particular tool imitates at a given time, and what the interpretation of that pattern should be. Technical analysts may disagree among themselves over the explanation of a given graphic representation. These practical indicators have been successfully used as input features to AI techniques.

III. RELATED RESEARCH

The use of prediction algorithms is in contradiction with one of the basic rules in finance, the Efficient Market Hypothesis (EMH) [11]. This hypothesis states that if one can get an advantage from analyzing past returns, the entire financial market will notice this advantage and as a consequence the price of the share will be corrected. This means that no abnormal returns can be obtained by examining past prices and returns of stocks. Although EMH is generally accepted, it was initially based on traditional linear statistical algorithms [11]. Many researchers have already rejected the hypothesis by using algorithms that can model more complex dynamics of the financial system [12, 13]. Since methods handling the complex and non-linear financial market are yielding positive results, researchers still try to invent better techniques. There are three major methodologies to predict the stock price behavior: (1) technical analysis, (2) time series forecasting and (3) machine learning and data mining [14]. The first category uses charts and plots as a principal tool. Analysts use these plots to make a buy or sell decision. The second category aims at predicting future stock prices by analyzing past returns on stock prices. Among the common methods are: auto-regressive technique (AR), the moving average model (MA), the autoregressive-moving average model (ARMA) and threshold auto-regressive model (TAR). The third category is “the science of extracting useful information from large data sets or databases” [15]. Popularity of data mining in the financial world has been growing since the main problem with predicting stock price direction is the huge amount of data. The data sets are too big to handle with non-data mining methods such that they obscure the underlying meaning and one cannot obtain useful information from it [16, 17]. Several algorithms have been used in stock price direction prediction literature. Simpler techniques such as the single decision tree, discriminant analysis, and Naïve Bayes have been replaced by better performing algorithms such as Random Forest, Logistic Regression and Neural Networks. General-purpose solvers such as Genetic Algorithms [18] have also been used but generally perform worse and are computationally more expensive. The majority of stock price direction prediction literature has focused on Logistic Regression, Neural Networks, K-NN and SVMs. Ensemble methods such as Radom Forest, (Stochastic) AdaBoost and Kernel Factory are still much unexplored in the domain of stock price direction prediction. In Table 1 we provide an overview of those algorithms used for predicting stock price direction in literature (we excluded single Decision Trees, Naïve Bayes, Discriminant Analysis and Genetic Algorithms because they have been superseded by newer and better methods discussed above). LR stands for Logistic Regression, NN stands for Neural Networks, KN stands for K-nearest neighbors, SVM stands for Support Vector Machines, RF stands for Random Forest, AB stands for AdaBoost and KF stands for Kernel Factory. It is clear from Table 1 that our study is the first to include all seven algorithms in one benchmark. Using suboptimal algorithms may hinder scientific progress in that important patterns in the data might be missed. In our study we will benchmark ensemble methods against single classifier models. The ensemble methods mentioned above all use a set of individually trained classifiers as base classifiers. We believe that the ensemble methods will outperform the individual classification models because they have proven to be very successful in several other domains such as face recognition [19], gene selection [20], and protein structural class prediction and credit scoring [21]. In stock price direction prediction literature both SVM and Random Forest (RF) have proven to be top performers [22, 23]. However, there is no consensus on which algorithm is best with SVM outperforming RF in [22] and vice versa in [23]. AdaBoost has also been shown to perform well, albeit not as well as Random Forest [24]. In an effort to help provide clarity in which algorithm is best, this study will benchmark SVM, AB, RF and four other algorithms.

Table 1: Algorithms for stock price direction prediction used in literature

	Predicted Method						
	LR	NN	KN	SVM	AB	RF	KF
Kim and Lee(2004)		X					
Rodriguez & Rodrigues(2004)	X	X			X	X	
Huang et al. (2005)		X		X			
Kumar and Thenmozhi (2006)	X	X		X		X	
Lunga and Marwala (2006)					X		
Wu et al. (2006)							
Wang and Chan (2007)	X						
Huang et al. (2008)	X	X	X	X			
Senol and Ozturan (2008)	X	X					
Lai, Fan, and Chang (2009)							
Lee (2009)		X		X			
Ou and Wang (2009)	X	X	X	X			
Kara, Boyaciogly and Baykan (2011)		X		X			
Wei and Cheng (2011)		X					
Subha and Nambi (2012)	X		X				
Lin, Guo, and Hu (2013)				X			
De Oliveira, Nobre, and Zárate (2013)		X					
Chen, Chen, Fan, and Huang (2013)		X					
Rechenthin et al. (2013)			X	X	X		
Ji, Che, and Zong (2014)		X					
Bisoi and Dash (2014)		X					
Zikowski (2015)				X			
Hafezi, Shahrabi, and Hadavandi (2015)	X						
Patel et al. (2015)		X		X		X	

IV. Results

The primary aim is to fit a model to the stock market dataset. We develop a linear model motivated from single layer perceptron to fit to the dataset. UCI Istanbul Stock exchange dataset issued for experimentation. In the first step, we train our linear model using the training data taken from the Istanbul stock exchange dataset. After train the model for 67 iterations, we obtained the weights for all the attributes as shown in Table 2.

Attributes	Weights
ISE	0.2034
ISE-500	0.3503
DAX	-0.1287

Table.2 Represents weights for respective attribute

It is observed that the Istanbul stock exchange National 100 index and the stock market return index of Germany have high correlation as shown in fig. 5

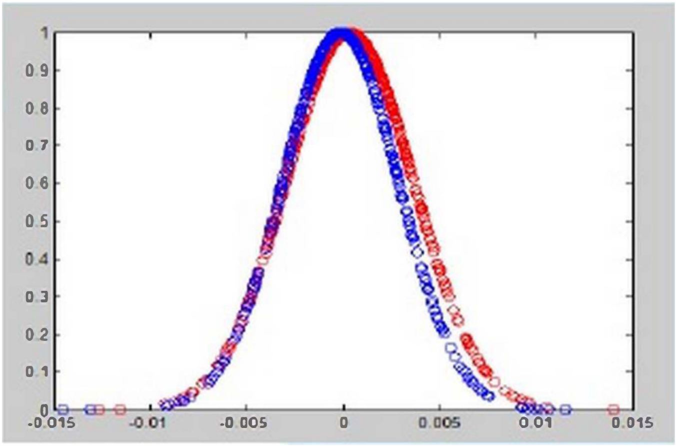

Fig. 6. Correlation between DAX & ISE 100

V. Conclusions

We have offered an architecture for learning to foretell stock price returns by bearing in mind a binary classification problem, where positive return forecasts are represented by the class label +1 and negative predictions -1. This architecture depends on an ensemble committee model of random forests, relevance vector machines, support vector machines, and a k-nearest neighbor constituent ensemble. The ensemble architecture we offered has descriptive power over direct billets that have been established to fall in the array of roughly 70% precision on testing data. The model can infrequently over-fit the data, turning to poor presentations on the test set, but in exercise this occurs in marginal conceivable presentations of the committee. Further research in the field of monetary modeling should, unquestionably, be fortified and followed. On the way, we commend reconnoitering the applications of supposed "Deep Learning" practices to stock price forecast. This primarily includes learning weight coefficients on a large directed and layered graph. Models of this procedure have, previously, proven problematic to train and improve, but recent advances in only the last few months have led to resurgence in deep learning.

REFERENCES

[1] Yoo, P.D.; Kim, M.H.; Jan, T., "Machine Learning Techniques and Use of Event Information for Stock Market Prediction: A Survey and Evaluation," Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce , vol.2, no., pp.835,841, 28-30 Nov. 2005

[2] Brzezinski, D.; Stefanowski, J., "Reacting to Different Types of Concept Drift: The Accuracy Updated Ensemble Algorithm," Neural Networks and Learning Systems, IEEE Transactions, vol.25, no.1, pp.81,94, Jan. 2014

[3] Hendahewa, C.; Pavlovic, V., "Analysis of Causality in Stock Market Data," Machine Learning and Applications (ICMLA), 2012 11th International Conference, vol.1, no., pp.288-293, 12-15 Dec. 2012

[4] Zhen Hu; Jie Zhu; Ken Tse, "Stocks market prediction using Support Vector Machine," Information Management, Innovation Management and Industrial Engineering (ICIII), 2013 6th International Conference, vol.2, no., pp.115,118, 23-24 Nov. 2013

[5] Mianji, F.A; Ye Zhang, "Robust Hyperspectral Classification Using Relevance Vector Machine," Geoscience and Remote Sensing, IEEE Transactions on , vol.49, no.6, pp.2100,2112, June 2011

[6] Friedman, J. H. (2002). Stochastic gradient boosting. Computational Statistics & Data Analysis, 38, 367–378.

[7] Xie Jun; Yu Lu; Zhu Lei; Xue Hui, "Boosting decision stumps to do pairwise classification," Electronics Letters, vol.50, no.12, pp.866-868, June 5 2014

[8] Hendahewa, C.; Pavlovic, V., "Analysis of Causality in Stock Market Data," Machine Learning and Applications (ICMLA), 2012 11th International Conference on, vol.1, no., pp.288, 293, 12-15 Dec. 2012

[9] Zhen Hu; Jie Zhu; Ken Tse, "Stocks market prediction using Support Vector Machine," Information Management, Innovation Management and Industrial Engineering (ICIII), 2013 6th International Conference on , vol.2, no., pp.115,118, 23-24 Nov. 2013

[10] W. Kreesuradej, D. Wunsch, and M. Lane, Time-delay neural network for small time series data sets, in World Cong. Neural Networks, San Diego, CA, June 1994.

[11] Malkiel, B. G., & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work∕. The Journal of Finance, 25(2), 383–417.

[12] Lo, Andrew W., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. The Journal of Finance, 55(4), 1705–1770.

[13] Malkiel, Burton G. (2003). The efficient market hypothesis and its critics. The Journal of Economic Perspectives, 17(1), 59–82.

[14] Hellström, T., Holmströmm, K. (1998). Predictable Patterns in Stock Returns. Technical Report Series IMa-TOM-1997-09

[15] Hand, D. J., Mannila, Heikki, & Smyth, Padhraic (2001). Principles of data mining. MIT Press.

[16] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. Communications ACM, 39(11), 27–34.

[17] Widom, J. (1995). Research problems in data warehousing. In Proceedings of the fourth international conference on information and knowledge management, CIKM ’95 (pp. 25–30). New York, NY, USA: ACM. 10.1145/221270.221319.

[18] Kuo, R. J., Chen, C. H., & Hwang, Y. C. (2001). An Intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network. Fuzzy Sets and Systems, 118(1), 21–45.

[19] Tan, X., Chen, S., Zhou, Z.-H., & Zhang, F. (2005). Recognizing partially occluded, expression variant faces from single training image per person with SOM and Soft k-NN ensemble. IEEE Transactions on Neural Networks, 16(4), 875–886.

[20] Diaz-Uriarte, R., & de Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. BMC Bioinformatics 2006, 7, 3.

[21] Paleologo, G., Elisseeff, A., & Antonini, G. (2010). Subagging for credit scoring models. European Journal of Operational Research, 201(2), 490–499.

[22] Kumar, M., & Thenmozhi, M. (2006). Forecasting Stock index movement: A comparison of support vector machines and random forest. SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, January 24, 2006.

[23] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. Expert Systems with Applications, 42, 259–268.

[24] Rodriguez, Pedro N., & Rodriguez, A. (2004). Predicting stock market indices movements. In C. Brebia (Ed.), Computational Finance and its Applications. Southampton: Marco Constantino, Wessex Institute of Technology.

View publication stats