Expert Systems with Applications 37 (2010) 1015–1022

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Automatic stock decision support system based on box theory and SVM algorithm

Qinghua Wen *, Zehong Yang, Yixu Song, Peifa Jia

State Key Laboratory on Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Keywords:
Stock data mining
Stock trading system
Box theory
Support vector machine

ABSTRACT

The stock market is considered as a high complex and dynamic system with noisy, non-stationary and chaotic data series. So it is widely acknowledged that stock price series modeling and forecasting is a challenging work. A significant amount of work has been done in this field, and in them, soft computing techniques have showed good performance. Generally most of these works can be divided into two categories. One is to predict the future trend or price; another is to construct decision support system which can give certain *buy/sell* signals. In this paper, we propose a new intelligent trading system based on oscillation box prediction by combining stock box theory and support vector machine algorithm. The box theory believes a successful stock buying/selling generally occurs when the price *effectively* breaks out the original oscillation box into another new box. In the system, two SVM estimators are first utilized to make forecasts of the upper bound and lower bound of the price oscillation box. Then a trading strategy based on the two bound forecasts is constructed to make trading decisions. In the experiment, we test the system on different stock movement patterns, i.e. bull, bear and fluctuant market, and investigate the training of the system and the choice of the time span of the price box. The experiments on 442 S&P500 components show a promising performance is achieved and the system dramatically outperforms buy-and-hold strategy.

1. Introduction

The study of stock market is a hot topic, because if successful, the result will transfer to fruitful rewards. A significant amount of work has been done in this field. Most of the works are the combination of soft computing technology and technical analysis in stock analysis. They can generally be divided into two categories.

One is to predict the future trend or price in the next day or next few days based on the historical prices and the technical indicators. Among them, as claimed by Grudnitski and Osburn (1993), artificial neural networks (ANN) are particularly well suited for finding accurate solutions in an environment characterized by complex, noisy, irrelevant or partial information. So many works focus on applying different neural networks into stock prediction (Chen & Leung, 2004; Kwon & Moon, 2007; Lin, Yang, & Song, 2008). Recently, a new type of learning machine, called support vector machines (SVM), has been receiving increasing attention in areas ranging from its original application in pattern recognition to the extended application of regression estimation. Cao and Francis (2003) show that SVM forecasts significantly better than the BP network in financial time series forecasting. Bao (2005) concludes the support vector machines is a robust technique for stock index regression. Other methods include, hidden markov model (HMM) (Hassan & Nath, 2005), reinforcement learning algorithm (Lee, 2001).

Another high-level research is to develop decision support system which helps to make trading decision. A typical work is (Baba, Inoue, & Yanjun, 2002) in which a decision support system was proposed by utilizing the neural network to make a forecast of the TOPIX in the future and the genetic algorithm to find an effective way of dealing. In more recent work (Li & Chen, 2006), based on the mechanisms of group decision making and cooperative learning, the proposed system combines of Wilson's XCS and neural network and the investment strategy was retrieved from the cooperative learning of 50 agents. The problem of all of these methods is they did not use or used little the practical knowledge and techniques accumulated in stock investment which we believed are critical to design a workable stock expert system.

In our previous work (Bao & Yang, 2008), an expert system which learn trading strategy by markov network probabilistic model from high-level representation of time series turning points and technical indicators was developed, which is based on the idea of price trend turning point in technical analysis. In this paper, the objective of this paper is to exploit the common used box theory of stock and develop an intelligent stock decision

* Corresponding author.
E-mail addresses: wenqh06@mails.tsinghua.edu.cn (Q. Wen), yangzehong@sina.com (Z. Yang), songyixu@sohu.com (Y. Song), dcsjpf@mail.tsinghua.edu.cn (P. Jia).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.05.093

support system based on oscillation box prediction. The basic idea of the box theory is that the stock price is supposed generally oscillates in a certain range in a period of time, which is called price box. The price will fall when it is near to the upper bound of the price box and will rise when it near to the lower bound of the box. The essence of box theory is when the price effectively breaks the upper bound or the lower bound of the oscillation box, the price will enter another oscillation box and it means the price will start an upward or downward trend. So it was the high time to buy or sell the stock (Nicolas, 2007). Box theory is a powerful tool; however, its application is an experience-dependent work. The difficulties are how to identify the price box and how to confirm the breakout is effective.

The proposed system is an automatic decision support system combining box theory and support vector machine algorithm. There are two modules in our trading system: oscillation box prediction module and trading strategy module. The support vector machines algorithm is utilized to make forecasts of the top and bottom of the oscillation box. Then trading strategy based on the box theory is constructed to make trading decisions. In the experiments, we investigate the performance of the supposed system on individual stocks with different movement patterns, test the average rate of profit of nearly all the stocks in S&P 500 and compare with the buy-and-hold strategy. The experiments show a promising performance is achieved with nearly 27% average rate of return, comparing with 6% by buy-and-hold strategy.

The advantage of the system lies in two aspects. First, it proposed a more reasonable trading tragedy which is based on the box theory but overcomes the problem of its poor performance in bearish market. Second, instead of trying to predict the true closing price as other literatures did, our system forecasts the upper bound and the lower bound of the price oscillation box in a certain period of time which are less sensitive to noise than the rough closing price series. This property makes the forecasts easier and more robust.

The rest of the paper is organized as follows: Section 2 describes the architecture and detailed design of the system. Section 3 introduces the theory of support vector machine and the layout of the SVM estimators in the system. Then the experiments and the corresponding analysis are shown in Section 4, and finally some concluding remarks are drawn from Section 5.

2. Architecture of the stock decision support system

To simplify the problem, we assume the price with a fixed time span, i.e. n-day. Then we take the highest value H_i and lowest value L_i of the closing price C n-day in the future as the estimation of the two bounds in the ith day. By the definition of H, L, we transfer the problem of identifying the effectiveness of the box breakout to check the relationship between closing price C_i and H_i, L_i, which consisted of the trading strategy. The basic idea of the trading system is illustrated by the Fig. 1.

The architecture of the trading system shows in Fig. 2. The input data include closing price and technical indicators which listed in (9). *SVMmax* and *SVMmin* are two SVM estimators which forecast the top and bottom of the price box. The model of *SVMmax* and *SVMmin* are learned incrementally from the historical price data. Based on the two forecasts, the system makes decision automatically according to the trading strategy.

2.1. Oscillation box prediction

In our system, we take the highest value H_i and lowest value L_i of the closing price C n-day in the future as the estimation of the two bounds in the ith day.

$$H_i = \max(C_i, C_{i+1}, \ldots, C_{i+n-1})$$
$$L_i = \min(C_i, C_{i+1}, \ldots, C_{i+n-1})$$

Then we obtain two time series H, L corresponding to the rough price series$C$. The reason of the method is twofold. In the first, it can transform the estimation of the price box into time series problem when maintains the basic meaning of price range, then we can employ the data mining method to solve it. Second, by the definition, it can design a more robust trading strategy, which is described in the following part. Forecasting H, L is equal to make a regression for the function which is mined from the historical information.

$$[\tilde{H}_i, \tilde{L}_i] = f(x_i, x_{i-1}, \ldots, x_{i-k}, y_i, y_{i-1}, \ldots, y_{i-k}, \ldots)$$

where $\tilde{H}_i, \tilde{L}_i$ represent the forecasts of the H_i, L_i, x, y are the factors related to the change of H_i, L_i.

The forecast models employ support vector machine algorithm. Two estimators based on SVM algorithm called *SVMmax* and *SVM-*

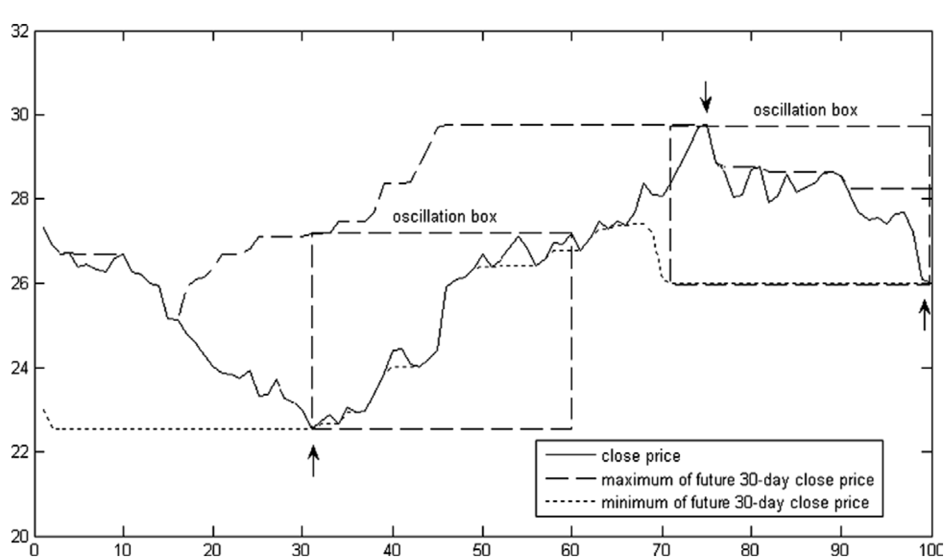

Fig. 1. '↑' means *buy*, '↓' means *sell*, or else do nothing. If the current close price is near to the bottom of the oscillation box and the bottom of the oscillation box is in uptrend, then buy. If the current close price is near to the top of the oscillation box and the top of the box is in downtrend, then sell. The time span n is set to 30.

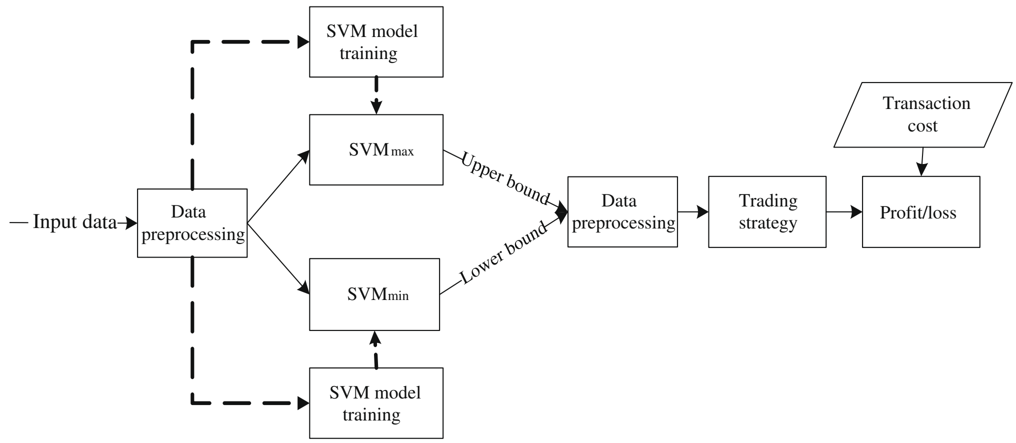

Fig. 2. The architecture of the trading system.

min are used to estimate $\tilde{H}_i, \tilde{L}_i$, respectively. We train the two forecast models by the sliding window method which is effective for time series data that is non-stationary.

2.2. Trading strategy

By the definition of H, L, we take the relationship between C_i and H_i, L_i as the basic design of the trading strategy. If the current price effective breaks out the original oscillation box, it means the price will start an uptrend and form another price box, the resistance in the original box will become an important support in the new box. According to the definition of H_i, L_i, the current closing price C_i will be close toL_i at this time. The same is true for lower bound breakout. But it have been noted not all C_i close to the two bounds corresponds to a box breakout. In fact every time the price oscillating to the box top and low, regardless breakout following or not, C_i is near toH_i or L_i and is an extreme value locally. Fortunately even when with no break, it is also a good time for short-time trading to win the price difference and we can add constraints to regulate the frequency of short-time trading. Based on the relationship between C_i and H_i, L_i, we define the trading strategy as follows.

If next trade=buy
  If $\frac{|C_i - \tilde{L}_i|}{C_i} \leqslant \sigma$ and $\tilde{L}_i$ is in uptrend
    If *sellprice* $- C_i \geqslant \varphi$
      then Buy, *buyprice* = C_i
        next trade=sell
else If $\frac{|C_i - \tilde{H}_i|}{C_i} \leqslant \sigma$ and $\tilde{H}_i$ is in downtrend
    If $C_i -$ *buyprice* $\geqslant \phi$
      then Sell, *sellprice* = C_i
        next trade=buy
    If $\frac{buyprice - C_i}{buyprice} \geqslant \theta$
      then Sell, *sellprice* = C_i
        next trade=buy

where σ is the transaction rate which varies with the regression accuracy of *SVMmax* and *SVMmin*. When the accuracy is high, σ could be smaller and vice verse. The role of σ is to regulate the trade frequency. The smaller the value, the fewer the number of transactions. φ, ϕ are used to filter out the false operation. θ is called stop-loss rate which is commonly used in stock transaction to minimize the loss. It is very useful especially in bear market. The condition about the trend of $\tilde{H}_i, \tilde{L}_i$ is to make sure the true trend reverse happened.

3. SVMmax and SVMmin

3.1. Overview of SVM regression

The support vector machines (SVMs) algorithm is based on statistical learning theory and structural risk minimization principle. It was first developed by Vapnik and his co-workers in the early 1990s to solve the pattern recognition (classification) problem. With the introduction of loss function such as Vapnik's-insensitive loss function, Huber's loss function, SVMs have been extended to solve regression estimation (function approximation) problems and applied successfully in time series forecasting, non-linear modeling and optimal control problem.

The basic idea of support vector regression is to map the input space into a high-dimension space by a non-linear mapping achieved implicitly by the trick of kernel function and to do a linear regression in the new feature space. Given a time series samples $\{X_i, y_i\} X_i \in R^l$, $y_i \in R$, $i = 1,2,\ldots,n$, where X_i is the input feature variable, y_i is the target value. SVM regression algorithm first maps the data to a high-dimension feature space Γ using a mapping Φ:$R^l \rightarrow \Gamma$. Then in the high-dimension, we find a linear function

$$f(x, w) = w^T \bullet \Phi(x) + b \tag{1}$$

in the condition of minimizing the sum of empirical risk and the complexity term $\|w\|^2$ which enforce flatness in feature space. Where $w \in \Gamma$ is the weight vector, b is a bias. The linear function in the high dimensional feature space corresponds to the non-linear function in the original lower dimensional feature space (Müller et al., 1997). In $\varepsilon - SV$ regression (Vapnik, 1995), our goal is to find a function $f(x)$ that has at most ε deviation from the targets y_i for all the training samples, at the same time is as flat as possible. That is to find (1) under the optimization problem:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2} w^T \bullet w \\ &\text{subject to} \quad \begin{cases} y_i - w^T \Phi(x_i) - b \leqslant \varepsilon \\ w^T \Phi(x_i) + b - y_i \leqslant \varepsilon \end{cases} \end{aligned} \tag{2}$$

Considering the existence of data outside the ε-insensitive tube

$$|y - (w^T \Phi(x) + b)| \leqslant \varepsilon \tag{3}$$

we introduce slack parameter δ_i, δ_i^* to cope with the otherwise infeasible constraints of the optimization problem (2), δ_i, δ_i^* are defined as

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}w^T \bullet w + C\sum_{i=1}^{n}(\delta_i + \delta_i^*) \\ &\text{subject to} \quad \begin{cases} y_i - w^T\Phi(x_i) - b \leqslant \varepsilon + \delta_i \\ w^T\Phi(x_i) + b - y_i \leqslant \varepsilon + \delta_i^* \end{cases} \end{aligned} \tag{4}$$

where $C>0$ is a prescribed parameter to determine the trade-off between the flatness of $f(x)$ and the amount up to which deviations larger than ε are tolerated. To solve this optimization problem, we construct a Lagrange function and introducing a dual set of Lagrange multipliers λ_i, λ_i^* and η_i, η_i^*

$$\begin{aligned} L = &\frac{1}{2}w^T \bullet w + C\sum_{i=1}^{n}(\delta_i + \delta_i^*) - \sum_{i=1}^{n}\lambda_i(\varepsilon + \delta_i - y_i + w^T\Phi(x_i) + b) \\ &- \sum_{i=1}^{n}\lambda_i^*(\varepsilon + \delta_i^* - y_i + w^T\Phi(x_i) + b) - \sum_{i=1}^{n}(\eta_i\delta_i + \eta_i^*\delta_i^*) \end{aligned} \tag{5}$$

where $\lambda_i, \lambda_i^*, \eta_i, \eta_i^* \geqslant 0$. This function has a saddle point which corresponds to the solution of the optimization problem. Solving (4) we get the optimal w, b and $f(x)$:

$$w^* = \sum_{i=1}^{n}(\lambda_i - \lambda_i^*)\Phi(x_i)$$

$$b^* = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^{*T} \bullet \Phi(x_i) \mp \varepsilon), \quad 0 < \lambda_i^{(*)} < C \tag{6}$$

$$f(x) = \sum_{i=1}^{n}(\lambda_i - \lambda_i^*)\Phi(x_i) \bullet \Phi(x) + b^* \tag{7}$$

The Lagrange multipliers λ_i, λ_i^* in (6) are sparse. Only when x_i is outside or on the ε-insensitive tube, they are non-zero. These points are called support vector. The training points in the tube are useless for $f(x)$. The dot products in $f(x)$ is computed in the new high dimensional space which is usually intractable. This problem is tackled by substituting the dot products with a kernel function which satisfies Mercer's conditions. Any symmetric kernel function satisfying Mercer's condition corresponds to a dot product in some feature space (Müller et al., 1997), then the computation of dot products in the high dimensional transforms to the simpler computation in the original low feature space. There are three commonly used kernel function, they are polynomial, RBF and sigmoid function. RBF kernel is most recommended, it defined as:

$$K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2g^2}\right) \tag{8}$$

when using the RBF kernel, there are two important parameter C, g is needed to be carefully chose in the SVM algorithm.

There are some deformations of SVM algorithm such as Least Squares support vector machines (LSSVM), Linear Programming support vector machines (LPSVM) and Bayesian support vector machines (BSM). For the detail refer relative literatures.

3.2. Setup of SVMmax and SVMmin

3.2.1. Input feature

In the system, we try to estimate function (1). Due to the target values related to the next n-day prices, so intuitively, the input data should at least contain the information of n-day before. The determination of n is set to 30 except for special note in the following experiments. The input data selected in the system totally include 240 features. For *SVMmax (SVMmin)*, they are:

$$\begin{aligned} &C_k, MA_k, RSI_k, ROC_k, FastK_k, SlowK_k, SlowD_k, \\ &\tilde{H}_{k-30}(\tilde{L}_{k-30}) \quad k = i, i-1, \ldots i-30 \end{aligned} \tag{9}$$

where MA_k, RSI_k, ROC_k, $FastK_k$, $SlowK_k$, $SlowD_k$ are technical indicators computed from closing price. The computations can be found in Martin (2002). The data pre-processing is crucial to the SVM algorithm. All of the input data is scaling to [-1,1] by

$$x_{scaled} = -1 + 2\frac{x - \min(x)}{\max(x) - \min(x)} \tag{10}$$

3.2.2. Determination of parameters

When applying SVM algorithm, another important thing that needs to be considered is what kernel function is to be used. As the dynamics of financial time series are strongly non-linear, it is believed that using non-linear kernel functions could achieve better performance than the linear kernel. In this study, the radial basis function as formula (8) is used as the kernel function of SVM. The parameter C, g are optimized by cross-validation and grid search. The training set is divided into five folds. One fold was taken as the validation set, others taken for training. The grid point with the best accuracy of predicting is used as the value of the two parameters.

3.2.3. Training

The sliding window validation is a train and test technique which is much more suitable for time series data that is slow varying or non-stationary. So the sliding window method is employed to train the two estimators. We divide the whole data set into overlapping training-test set. The length of window is 1050 in which the 1000 data points are taken as the training set, the 50 data points following are taken as the test set.

4. Experiment

In order to test the feasibility of the trading system, we experiment in several typical stock movements such as bull market, bear market, fluctuant market and so on. The test criteria include the trade profit/loss and the accuracy of SVM regression. The training of SVM algorithm and the parameter n of the system is also investigated. All of experiments run in the MATLAB environment and all of the data are gathered from the Yahoo's financial web site. The SVM algorithm employed in the experiments is the LibSVM developed by Lin, for the details about the LibSVM refer to Chang and Lin (2001).

4.1. Performance evaluation

MSE (mean squared error) and SCC (squared correlation coefficient) is used to measure the accuracy of SVM regression. MSE reflects the local fitness of the regression and SCC reflects the global fitness. MSE and SCC defined as follows

$$\begin{aligned} MSE &= \frac{1}{N}\sum_{i=1}^{N}(y_i - y_i^*)^2 \\ SCC &= \frac{\left[\sum_{i=1}^{N}(y_i - \bar{y}_i)(y_i - \bar{y}_i^*)\right]^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2 \sum_{i=1}^{n}(y_i^* - \bar{y}_i^*)^2} \end{aligned} \tag{11}$$

where y_i is the actual output and y_i^* is the estimate, $\bar{y}_i, \bar{y}_i^*$ are their averages.

Rate of profit is defined as (12), Y_0 is the initial fund, Y is the final return after trading strictly according to the system decision during certain period.

$$rate\ of\ profit = (Y - Y_0)/Y_0 \times 100\% \tag{12}$$

In the trading process, suppose \$ 1000 initial fund and trade all funds/stocks at each operation. For the convenience to compare with other method the last trading decision should be *sell*, if not, we sell the stock at the end of the trading period. To simulate the real stock transaction, a 0.5 percent transaction cost for each trading is assumed.

4.2. Typical stock movement and their trade

Fig. 3 shows a movement in a fluctuant and bull market and long period trade (almost 4 years) with transaction rate σ set to 0.05 and stop-loss rate θ set to 14% and φ, ϕ set to 0, 5, respectively. In the experiment, our trading system is able to profit up to 121.63% while the market gains about 15.3%. The MSE and SCC for *SVMmax* are 0.00661 and 0.7109 while for *SVMmin* they are 0.00842 and 0.6677. Test data set are sampled from March 27, 1990 to March 7, 1994.

Fig. 4 shows a movement in a fluctuant and bear market and short period trade (less than 2 years) with transaction rate σ set to 0.12 and stop-loss rate θ set to 15% and φ, ϕ set to 0, 2, respectively. The profit is up to 21.04% while the stock loss is about 11.3%. The MSE and SCC for SVMmax are 0.00890 and 0.8370 while for SVMmin they are 0.00842 and 0.6677. Test data are sampled from Jun 13, 1970 to Jan 11, 1972.

In other word, our system can outperform buy-and-hold strategy in the fluctuant market.

Movement in Fig. 5 is an overall bull market and 400 days short trade with transaction rate σ set to 0.09 and stop-loss rate θ set to 15% and φ, ϕ set to 0, 2, respectively. The experiment result shows our trading system achieves a perfect excellent trade with profit 139.5% while the stock gains about 80.6%. The result benefits from the good regression of the two SVM estimators. The MSE and SCC for SVMmax are 0.00662 and 0.8806 while for SVMmin they are 0.00776 and 0.90364. Test data are sampled from Mar 17, 2004 to Oct 17, 2005.

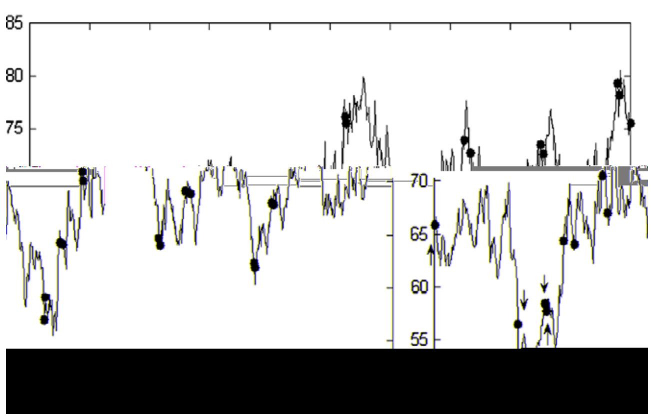

Fig. 3. Trading log('↑': buy, '↓': sell) of ALCOA INC for 4 years.

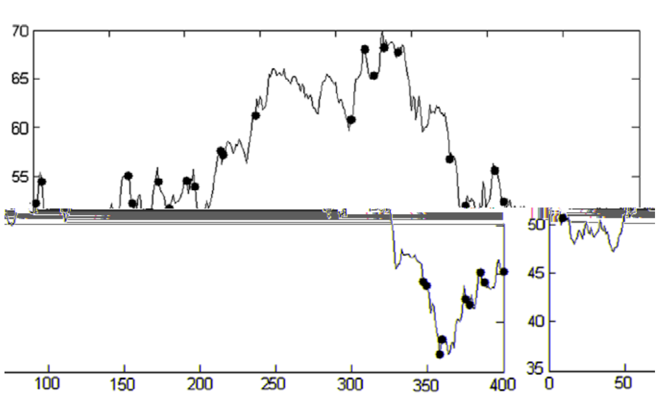

Fig. 4. Trading log of ALCOA INC for almost 2 years.

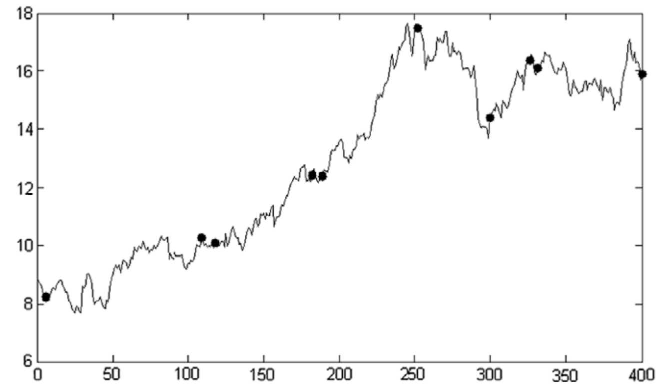

Fig. 5. Trading log of AES CP INC for 400 days.

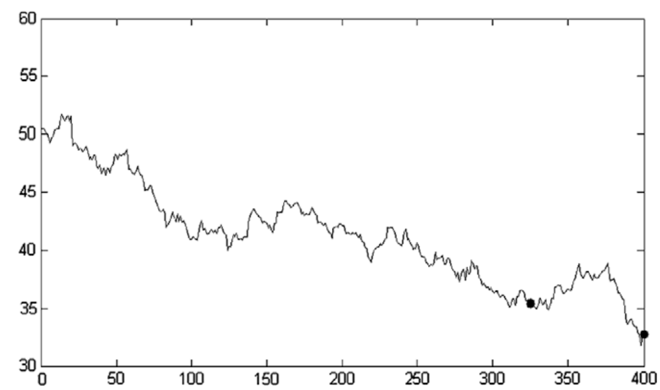

Fig. 6. Trading log of TRIBUNE INC for 400 days.

Movement in Fig. 6 is an overall bear market and 400 days short trade with transaction rate σ set to 0.01 and stop-loss rate θ set to 15% and φ, ϕ set to 0, 2, respectively. In the experiment the stock losses totally 35.2% in the period, our trading system losses only 7.6%. The MSE and SCC for SVMmax are 0.00157 and 0.9921 while for SVMmin they are 0.00218 and 0.8336. Test data are sampled from Mar 17, 2004 to Oct 17, 2005.

4.3. Trade on S&P500

In this experiment, we evaluate the average performance on 500 stocks (S&P 500 components) and check whether the average profit outperforms the S&P 500 index.

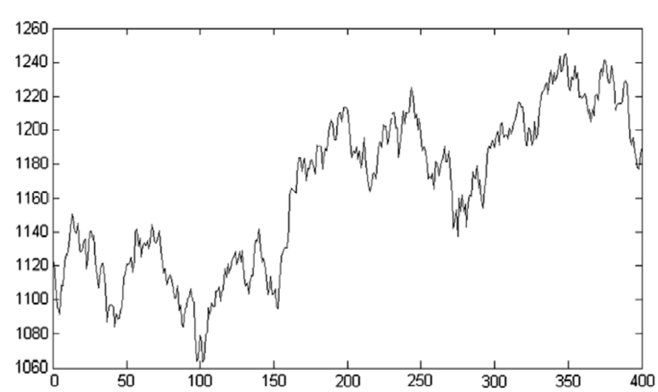

Fig. 7. S&P 500 index from Mar 17, 2004 to Oct 17, 2005.

Fig. 7 shows the S&P 500 index in 400 trading days from Mar 18, 2004 to Oct 17, 2005. The overall profit is 6.57%.

In the experiments, 442 stocks which have over 3000 daily data among the S&P 500 components are selected. In the SVM regression training process, the cost C is set to 550 and 350 for SVMmax and SVMmin, respectively. The parameter g of RBF kernel function is set to 0.00002125 for both estimators. The stop-loss rate θ is set to 15% and φ, ϕ set to 0, 2, respectively. The average profit is 25.94% while the profit of buy-and-hold strategy is 5.77% for the 400 trading days, the detailed statistical result is in Table 1. That means the profit of the trading system is much better than the S&P 500 index. For convenience of the experiments, it is needed to note the set value of C, g is not optimal for all of the stocks.

In an effort to enhance the performance, we expand the proportion between the number of training set and of test set from 3:1 to 5:1 with other parameters same for the 57 stocks which underperform the buy-and-hold strategy in uptrend and the 29 stocks which loss in downtrend. The overall average profit rises to 27.23%. The statistical result is in Table 2. According to the Table 2, most stocks can profit only with 14 stocks loss due to the overall downtrend during the test period and 39 stocks underperform the buy-and-hold strategy due to the overall uptrend during the test period in which obviously the buy-and-hold is the best trade strategy.

Only 14 stocks in the 422 stocks loss due to the overall downtrend during the test period in which there is no change to profit. 39 stocks in the 422 stocks underperform the buy-and-hold strategy due to mainly the overall uptrend during the test period in which obviously the buy-and-hold is the best trade strategy.

Due to the good performance of the box theory in bull market, all the bullish stocks investigated profit during the period of time and the average profit in bull market is as high as 39.11%. We also can find even in the bear market, the system has 13.79% average profit when the average loss is 21.04% by buy-and-hold. That means the system in bearish market still has a good performance.

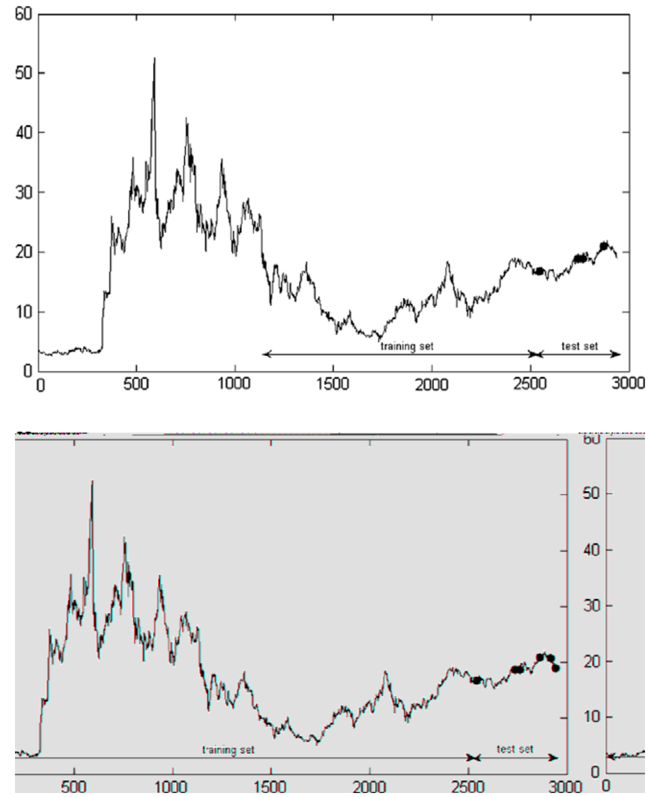

Fig. 8. Trading log of AES CP INC for 400 days from Mar 17, 2004 to Oct 17, 2005.with transaction rate σ set to 0.25. Top: There is 1400 data training points, the profit is 23.92%. Bottom: the training set extend to 2492 points, the profit is 14.02%.

4.3.1. Proportion between the number of training set and test set

The experiment in Table 2 raised the question of what is proper proportion between the number of training set and test set and whether the expanding of training set will enhance the performance generally. To identify this problem, in this experiment, we expand the proportion to 5:1 for all 422 stocks and all of the variable values remain unchanged. The result shows in Table 3. Compared with the experiment in Table 1, although the number of stocks underperforming buy-and-hold strategy decreases from 58 to 53 and the number of loss stocks decreases from 29 to26 slightly, the average profit is also decreases from 25.94% to

Table 1
Average performance of trading 422 stocks for 400 days.

Market pattern	Number of stock	Less than buy-and-hold	Less (%)	Number of loss	Loss (%)	Average profit (%)	Average profit of buy-and-hold (%)
Bull	224	57	25.44	0	0	37.69	29.47
Bear	198	1	0.51	29	14.65%	12.65	−21.04
Total	422	58	13.12	29	6.87%	**25.94**	**5.77**

Table 2
Enhanced average performance of trading 422 stocks for 400 days.

Market pattern	Number of stock	Less than buy-and-hold	Less (%)	Number of loss	Loss (%)	Average profit (%)	Average profit of buy-and-hold (%)
Bull	224	39	17.41%	0	0	39.11	29.47
Bear	198	0	0	14	7.07%	13.79	−21.04
Total	422	39	9.24%	14	3.32	**27.23%**	**5.77**

Table 3
Average performance of trading 422 stocks for 400 days with extended training set.

Market pattern	Number of stock	Less than buy-and-hold	Less (%)	Number of loss	Loss (%)	Average profit (%)	Average profit of buy-and-hold (%)
Bull	224	51	22.77	0	0	36.29	29.47
Bear	198	2	1.01	26	13.13%	10.33	−21.04
Total	422	53	12.56	26	8.53%	**24.11**	**5.77**

Table 4
Average performance of trading 50 stocks by varying *n*.

	3/3	5/5	8/8	10/10	15/15	20/20	30/30	40/40	50/50
Average profit (%)	35.5	39.49	**40.79**	39.83	34.35	35.86	**37.88**	37.01	34.89
number of transaction	9.8	8.7	8.1	7.7	7.3	8.4	8.8	7.9	7.7
Average SCC of *SVMmax*	0.969	0.945	0.914	0.888	0.843	0.792	0.706	0.623	0.562
Average SCC of *SVMmin*	0.959	0.925	0.881	0.852	0.786	0.732	0.609	0.523	0.462

Table 5
Performance of trading microsoft.

σ	0.005	0.01	0.015	0.02	0.025	0.03	0.035	0.04	Buy-and-hold
Number of transactions	4	5	7	8	8	8	6	6	1
Profit (%)	−2.77	19.03	53.37	73.94	**74.00**	72.17	29.37	15.41	5.16

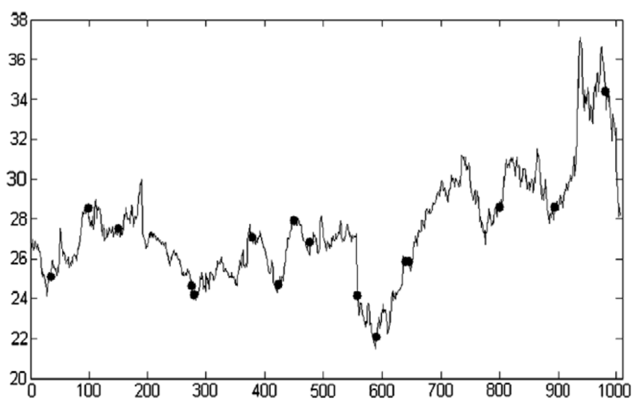

Fig. 9. Trading log of Microsoft for 4 years from Feb 12, 2004 to Feb 12, 2008 with trade ratio σ set to 0.025 and profit 74.00%. The profit is 5.16% by the buy-and-hold strategy.

24.11%. We also find from the experiment that the performances improve only when the expanded training set have the same moving pattern with the test set, otherwise, the expanding of training set often leads to deterioration of performance. An example shows in Fig. 8.

4.3.2. Choice of n

The width of the oscillation box *n* is another parameter in this trading system need to be identified. Intuitively, the *n* smaller, the higher the accuracy of the two estimators *SVMmax* and *SVMmin* will be got, but the trade decision may be less optimal due to the forecasts concerning less days ahead and vice verse.

In order to identify *n*, this experiment is carried on 50 stocks (the alphabetically first 50 stock in S&P 500). The average profit is investigated by varying *n*. the *n* changes from 3 to 50. The result shows in Table 4. */* means to predict the upper/lower bound of the oscillation box * days ahead using the information * days before. Note that the information of stock in a day includes eight features as (9).

From the result, we can find the best performance achieved in *n* =8 and 30. Further studying the movements of the 50 stocks, the performance is better with *n* = 8 when the stocks are dramatically fluctuant during the test period, the better performance is achieved with *n* = 30 when the stocks change gently.

According the experiment, n set to 8 is recommended when the market is in dramatic fluctuation, otherwise *n* set to 30 is recommended.

4.4. Trade on individual stock

This experiment is carried on two commonly selected stocks Microsoft and IBM in many literatures. *Number of transactions*is the number of the buy-and-sell operation.

Table 5 shows the performance of trading Microsoft. The time span *n* is set to 30 and the stop-loss rate θ is set to 10% and all of other parameters have the same values as the experiment in Table 1. The MSE and SCC for *SVMmax* are 0.00890 and 0.8370 while for *SVMmin* they are 0.00842 and 0.6677. When transaction rate σ is set to 0.025 the system achieves the best profit with 74.00%. The trade log shows in Fig. 9.

Table 6 lists the performance of trading IBM. The time span *n* is set to 30, the cost *C* is set to 1000 and 2000 for *SVMmax* and *SVMmin*, respectively while *g* is set to 0.00002125 for both estimator. The stop-loss rate θ is set to 15% and φ, ϕ set to 0, 2, respectively. The MSE and SCC for *SVMmax* are 0.00016 and 0.9111 while for *SVMmin* they are 0.00020 and 0.8706. When transaction rate σ is set to 0.015 the system achieves the best profit with 53.31%. The trade log shows in Fig. 10.

5. Conclusion and future work

In this paper, we present a novel implementation of intelligent stock decision support system based on oscillation box prediction. Two estimator based on SVM regression algorithm is used to forecast the upper and lower bound of the oscillation box respectively. We also investigated the choice of the width of the box and the training of the SVM algorithm in financial time series. In order to examine the feasibility, we test the system on typical stock movements and S&P 500 components and typical individual stocks. The experiments show a promising performance and dramatically outperform buy-and-hold strategy.

The prediction algorithms with a single algorithm are often fragile because of the complexity of the stock movement, and there is a wide acceptance of benefit of the synergy effect. In the system,

Table 6
Performance of trading IBM.

σ	0.005	0.01	0.015	0.02	0.025	0.03	0.035	0.04	Buy-and-hold
Number of transactions	2	2	3	4	4	6	10	13	1
Profit (%)	10.10	47.38	**53.31**	20.83	24.33	5.97	15.33	−13.63	7.28

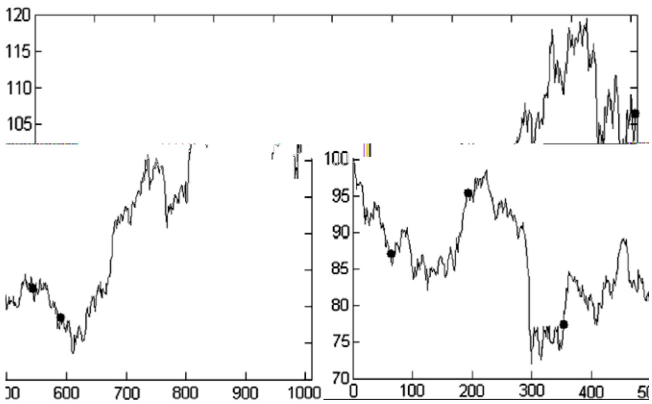

Fig. 10. Trading log of IBM for 4 years from Feb 12, 2004 to Feb 12, 2008 with trade ratio σ set to 0.015 and profit 53.31%. The profit is 7.28% by the buy-and-hold strategy.

the accuracy of the forecasts for the upper and lower bound of the oscillation box is also the bottleneck of the performance. So the future work includes building more robust estimators to improve the accuracy of the forecasts by combining other soft computing techniques together.

References

Baba, N., Inoue, N., Yanjun, Y. (2002). Utilization of soft computing techniques for constructing reliable decision support system for dealing stocks. In *IJCNN'02: Proceedings of the 2002 international joint conference on neural networks, Honolulu, Hawaii*.

Bao, Y.-K. (2005). Forecasting stock composite index by fuzzy support vector machines regression. In *Proceedings of the fourth international conference on machine learning and cybernetics, Guangzhou, 18–21 August 2005*.

Bao, D., & Yang, Z. (2008). Intelligent stock trading system by turning point confirming and probabilistic reasoning. *Expert Systems with Applications, 34*(1), 620–627.

Cao, L. J., & Francis, E. H. (2003). Tay, support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks, 14*(6).

Chang, C.-C., Lin, C.-J. (2001). LIBSVM: A library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

Chen, A. S., & Leung, M. T. (2004). Regression neural network for error correction in foreign exchange forecasting and trading. *Computers and Operations Research, 31*, 1049–1068.

Grudnitski, G., & Osburn, L. (1993). Forecasting S&P and gold futures prices: an application of neural networks. *Journal of Futures Market, 13*, 631–643.

Hassan, M. R., Nath, B., (2005). Stock market forecasting using hidden Markov model: A new approach. In *Proceedings of the fifth international conference on intelligent systems design and applications* (pp. 192–196).

Kwon, Y.-K., & Moon, B.-R. (2007). A hybrid neurogenetic approach for stock forecasting. *IEEE Transaction on Neural Networks, 18*(3).

Lee, J. W. (2001). Stock price prediction using reinforcement learning. In *IEEE international joint conference on neural networks* (pp. 690–695).

Li, J.-B., Chen, A.-P. (2006). Refined group learning based on XCS and neural network in intelligent financial decision support system. In *Proceedings of the sixth international conference on intelligent systems design and applications (ISDA'06)*.

Lin, X., Yang, Z., Song, Y., Washio T., et al. (2008). The application of echo state network in stock data mining. In *PAKDD 2008: Vol. 5012. LNAI* (pp. 932–937).

Martin, P. (2002). Technical analysis explained (4th ed.). Paperback. McGraw-Hill Company. ISBN0071226699.

Müller, K.-R., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V. (1997). Predicting time series with support vector machines. In *Proceedings, international conference on artificial neural networks. Lecture notes in computer science* (p. 999). Springer.

Nicolas, D. (2007). *How I made two million dollars in the stock market*. BN Publishing.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.