Mining Stock Market Tendency Using GA-Based Support Vector Machines

Lean Yu[1,2], Shouyang Wang[1,2,3], and Kin Keung Lai[3,4]

[1] Institute of Systems Science, Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing 100080, China
[2] School of Management, Graduate University of Chinese Academy of Sciences,
Chinese Academy of Sciences, Beijing 100039, China
{yulean, sywang}@amss.ac.cn
[3] College of Business Administration, Hunan University, Changsha 410082, China
[4] Department of Management Sciences, City University of Hong Kong,
Tat Chee Avenue, Kowloon, Hong Kong
mskklai@cityu.edu.hk

Abstract. In this study, a hybrid intelligent data mining methodology, genetic algorithm based support vector machine (GASVM) model, is proposed to explore stock market tendency. In this hybrid data mining approach, GA is used for variable selection in order to reduce the model complexity of SVM and improve the speed of SVM, and then the SVM is used to identify stock market movement direction based on the historical data. To evaluate the forecasting ability of GASVM, we compare its performance with that of conventional methods (e.g., statistical models and time series models) and neural network models. The empirical results reveal that GASVM outperforms other forecasting models, implying that the proposed approach is a promising alternative to stock market tendency exploration.

1 Introduction

Mining stock market tendency is regarded as a challenging task due to its high volatility and noisy environment. There have many studies using artificial neural networks (ANNs) in this area. The early days of these studies focused on application of ANNs to stock market prediction, such as [1-3]. Recent research tends to hybridize several artificial intelligence (AI) techniques [4]. Some researchers tend to include novel factors in the learning process. Kohara et al. [5] incorporated prior knowledge to improve the performance of stock market prediction. Tsaih et al. [6] integrated the rule-based technique and ANN to predict the direction of the S&P 500 stock index futures on daily basis. Similarly, Quah and Srinivasan [7] proposed an ANN stock selection system to select stocks that are top performers from the market and to avoid selecting under performers. They concluded that the portfolio of the proposed model outperformed the portfolios of the benchmark model in terms of compounded actual returns overtime. Kim and Han [8] proposed a genetic algorithms approach to feature discretization and the determination of connection weights for ANN to predict the stock price index. They suggested that their approach reduced the dimensionality of the feature space and enhanced the prediction performance.

X. Deng and Y. Ye (Eds.): WINE 2005, LNCS 3828, pp. 336 – 345, 2005.

Although a large number of successful applications have shown that ANN can be a very useful tool for stock market modeling and forecasting [9], some of these studies, however, showed that ANN had some limitations in learning the patterns because stock market data has high volatility and noise. ANN often exhibits inconsistent results on noisy data [10]. Furthermore, ANN also suffers from difficulty in trapping into local minima, overfitting and selecting relevant input variables [11].

In order to overcome the above main limitations of ANN, a novel intelligent learning algorithm, genetic algorithm-based support vector machine (GASVM) approach, is proposed in this study. First of all, the SVM, a novel intelligent algorithm developed by Vapnik and his colleagues [12] is used to avoid local minima and overfitting of traditional neural network models. Actually, many traditional neural network models had implemented the empirical risk minimization (ERM) principle; SVM implements the structural risk minimization (SRM) principle. The former seeks to minimize the mis-classification error or deviation from correct solution of the training data but the latter searches to minimize an upper bound of generalization error. In addition, the solution of SVM may be global optimum while other neural network models may tend to fall into a local optimal solution. Thus, overfitting is unlikely to occur with SVM [12]. Subsequently, to select relevant variable and reduce the complexity of SVM, a genetic algorithm is used. Therefore the proposed GASVM approach has two distinct advantages. One is that the computations of GASVM are reduced by the decrease of model inputs and running speed will be accelerated. Another is that GASVM can avoid some defects of neural network models, such as local minima and overfitting.

Although the proposed GASVM has the above advantages, there are few studies for the application of SVM in mining stock market tendency. Kim [13] applied SVM to predict the direction of changes of Korea composite stock price index. Recently, Huang et al. [14] examined the predictability of stock index with SVM. They showed that SVM outperformed the BP networks on the criteria of hit ratios. However, in the existing literature, no studies mentioned the related input variables selection. Our approach fills up the gap in the literature.

The main motivation of this study is to propose a new data mining approach for exploring stock market tendency and to test the predictability of the proposed GASVM model by comparing it with conventional models and neural network models. The rest of the study is organized as follows. The next section will describe the proposed GASVM model building process in detail. In Section 3, we give an experiment scheme and Empirical results and analysis are reported in this section. The concluding remarks are given in Section 4.

2 Model Building Process

In this section, the GASVM model building process is presented in detail. First of all, a basic theory of the SVM in classification is described. Then the genetic algorithm for variable selection will be proposed to reduce the model complexity of SVM. Based on the genetic algorithm and SVM, a GASVM model is built finally.

2.1 The Basic Theory of SVM

The SVM used here is the support vector classification (SVC) proposed by Vapnik [12]. The basic idea of SVM is to use linear model to implement nonlinear class boundaries through some nonlinear mapping the input vector into the high-dimensional feature space. A linear model constructed in the new space can represent a nonlinear decision boundary in the original space. In the new space, an optimal separating hyperplane is constructed. Thus SVM is known as the algorithm that finds a special kind of linear model, the maximum margin hyperplane. The maximum margin hyperplane gives the maximum separation between the decision classes. The training examples that are closest to the maximum margin hyperplane are called support vectors. All other training examples are irrelevant for defining the binary class boundaries.

For the linearly separable case, a hyperplane separating the binary decision classes in the three-attribute case can be represented as the following equation:

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 \tag{1}$$

Where y is the outcome, x_i are the attribute values, and there are four weights w_i to be learned by the learning algorithm. In Equation (1), the weights w_i are parameters that determine the hyperplane. The maximum margin hyperplane can be represented as the following equation in terms of the support vectors:

$$y = b + \sum \alpha_i y_i \mathbf{x}(i) \cdot \mathbf{x} \tag{2}$$

Where y_i is the class value of training example $\mathbf{x}(i)$, $\cdot$represents the dot products. The vector $\mathbf{x}$ represents a test example and the vectors $\mathbf{x}(i)$ are the support vectors. In this equation, b and α_i are parameters that determine the hyperplane. From the implementation point of view, finding the support vectors and determining the parameters b and α_i are equivalent to solving a linearly constrained quadratic programming.

As mentioned above, SVM constructs linear model to implement nonlinear class boundaries through the transforming the inputs into the high-dimensional feature space. For the nonlinearly separable case, a high-dimensional version of Equation (2) is represented as follows:

$$y = b + \sum \alpha_i y_i K(\mathbf{x}(i) \cdot \mathbf{x}) \tag{3}$$

The function $K(\mathbf{x}(i) \cdot \mathbf{x})$ is defined as the kernel function. There are some different kernels for generating the inner products to construct machines with different types of nonlinear decision surfaces in the input space. Choosing among different kernels the model that minimizes the estimate, one chooses the best model. Com Common examples of the kernel function are the polynomial kernel $K(x, y) = (xy + 1)^d$ and the Gaussian radial basis function $K(x, y) = \exp\left(-(x - y)^2 / 2\sigma^2\right)$ where d is the degree of the polynomial kernel and σ is the bandwidth of the Gaussian radial basis function kernel [13]. The construction and selection of kernel function is important to SVM, but in practice the kernel function is often given directly.

For the separable case, there is a lower bound 0 on the coefficient α_i in Equation (3). For the non-separable case, SVM can be generalized by placing an upper bound C on the coefficient α_i in addition to the lower bound [4].

2.2 Feature Vector Selection with Genetic Algorithm for SVM Modeling

In this study, we use genetic algorithm (GA) to extract feature vector of model inputs for SVM modeling. To date, GA has become a popular optimization method as they often succeed in finding the best optimum in contrast to most common optimization algorithms. Genetic algorithm imitates the natural selection process in biological evolution with selection, mating reproduction and mutation, and the sequence of the different operations of a genetic algorithm is shown in the left part of Fig. 1. The parameters to be optimized are represented by a chromosome whereby each parameter is encoded in a binary string called gene. Thus, a chromosome consists of as many genes as parameters to be optimized. Interested readers can be referred to [15-16] for more details. In the following GA for feature variable selection is discussed.

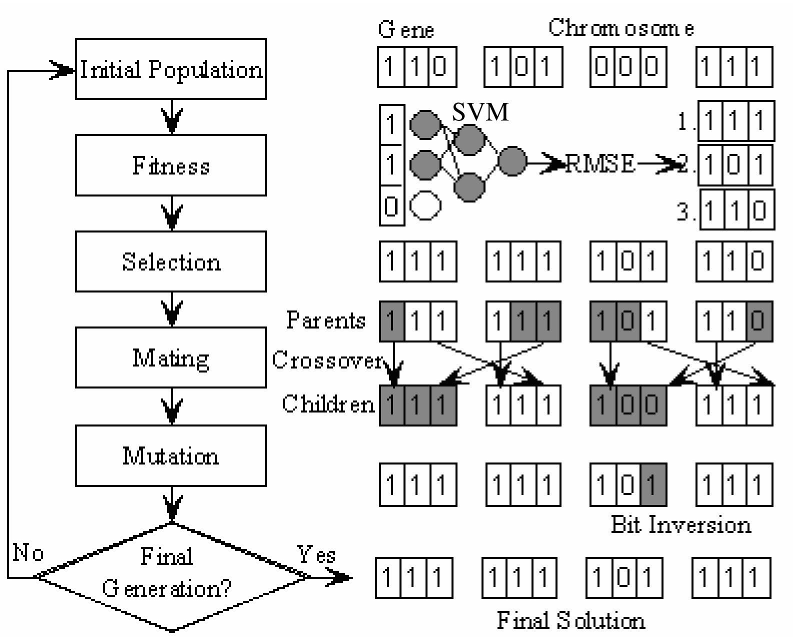

Fig. 1. The variable selection with the genetic algorithm for SVM

First of all, a population, which consists of a given number of chromosomes, is initially created by randomly assigning "1" and "0" to all genes. In the case of variable selection, a gene contains only a single bit string for the presence and absence of a variable. The top right part of Fig. 1 shows a population of four chromosomes for a three-variable selection problem. In this study, the initial population of the GA is randomly generated except of one chromosome, which was set to use all variables.

The binary string of the chromosomes has the same size as variables to select from whereby the presence of a variable is coded as "1" and the absence of a variable as "0". Consequently, the binary string of a gene consists of only one single bit. The subsequent work is to evaluate the chromosomes generated by previous operation by a so-called fitness function, while the design of the fitness function is a crucial point in using GA, which determines what a GA should optimize. Here the goal is to find a small subset of variables from many candidate variables. In this study, the SVM is used for modeling the relationship between the input variables and the responses. Thus, the evaluation of the fitness starts with the encoding of the chromosomes into SVM model whereby "1" indicates that a specific variable is used and "0" that a variable is not used by the SVM model. Then the SVM models are trained with a training data set and after that, a testing data set is predicted. Finally, the fitness is calculated by a so-called fitness function f. For a prediction/classification problem, for example, our fitness function for the GA variable selections can use the following form:

$$f = 0.3 RMSE_{training} + 0.7 RMSE_{testing} - \alpha (1 - n_v / n_{tot}) \tag{4}$$

where n_v is the number of variables used by the SVM models, n_{tot} is the total number of variables and *RMSE* is the root mean square error, which is defined in Equation (5) with N as total number of samples predicted, y_t as the actual value and $\hat{y}_t$ as the predicted value:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\hat{y}_t - y_t)^2} \tag{5}$$

From Equation (4), we find that the fitness function can be broken up into three parts. The first two parts correspond to the accuracy of the SVM models. Thereby $RMSE_{training}$ is based on the prediction of the training data used to build the SVM models, whereas $RMSE_{testing}$ is based on the prediction of separate testing data not used for training the SVM models. It was demonstrated in [17] that using the same data for the variable selection and for the model calibration introduces a bias. Thus, variables are selected based on data poorly representing the true relationship. On the other hand, it was also shown that a variable selection based on a small data set is unlikely to find an optimal subset of variables [17]. Therefore, a ratio of 3:7 between the influence of training and testing data was chosen. Although being partly arbitrary this ratio should give as little influence to the training data as to bias the feature selection yet taking the samples of the larger training set partly into account. The third part of the fitness function rewards small models using only few variables by an amount proportional to the parameter *a*. The choice of *a* will influence the number of variables used by the evolved SVM. A high value of results in only few variables selected for each GA whereas a small value of a results in more variables being selected. In sum, the advantage of this fitness function is that it takes into account not only the testing error of test data but also partially the training error and primarily the number of variables used to build the corresponding SVM models.

After evolving the fitness of the population, the best chromosomes with the highest fitness value are selected by means of the roulette wheel. Thereby, the chromosomes are allocated space on a roulette wheel proportional to their fitness and thus the fittest chromosomes are more likely selected. In the following mating step, offspring chromosomes are created by a crossover technique. A so-called one-point crossover technique is employed, which randomly selects a crossover point within the chromosome. Then two parent chromosomes are interchanged at this point to produce two new offspring. After that, the chromosomes are mutated with a probability of 0.005 per gene by randomly changing genes from “0” to “1” and vice versa. The mutation prevents the GA from converging too quickly in a small area of the search space. Finally, the final generation will be judged. If yes, then the optimized subsets are selected. If no, then the evaluation and reproduction steps are repeated until a certain number of generations, until a defined fitness or until a convergence criterion of the population are reached. In the ideal case, all chromosomes of the last generation have the same genes representing the optimal solution.

2.3 GA-Based SVM Model in Data Mining

Generally, SVM cannot reduce the input information. When the input space dimension is rather large, the process of solving SVM problem will require too much time. It is therefore necessary for SVM to preprocess the input feature vectors. In this study, the genetic algorithm is used to preprocess the input feature vectors. Then the processed feature vectors are sent to SVM model for learning and training. Thus, a novel forecasting approach, GA-based SVM (GASVM) model integrating GA and SVM, is formulated for data mining, as illustrated in Fig. 2.

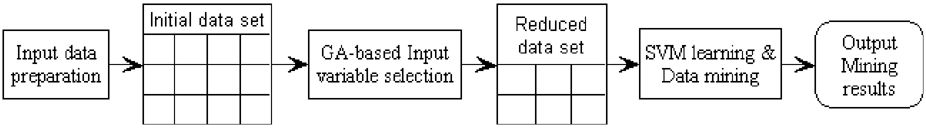

Fig. 2. The main process of GASVM for data mining

As can be seen from Fig.2, the GASVM data mining model comprises three phases: data preparation, input variable selection, and SVM learning and mining. Generally, for a specific problem, the first step is to collect and prepare some input data (initial data set) in terms of factor analysis. The second step is to select some typical feature variables using GA method mentioned previously, and a reduced data set can be obtained. The final step is to train SVM model for mining purpose and to output the mining results accordingly.

3 Empirical Study

3.1 Research Data

The research data used in this study is technical indicators and the direction of change in the daily S&P500 stock price index. Since we attempt to mine the stock index movement tendency, technical indicators are used as input variables. This study selects 18 technical indicators to make up the initial attributes, as determined by the review of domain experts and prior research [13]. The descriptions of initially selected attributes are presented in Table 1.

Table 1. Initially selected feature indicators and their formulas

Feature indicators	Formulas
Price (P)	x_t, $(t = 1, 2, \ldots, n)$
Stochastic oscillator (SO)	$((x_t - x_l(m))/(x_h(m) - x_l(m)))$
Moving stochastic oscillator (MSO)	$\frac{1}{m}\sum_{i=t-m+1}^{t}(so_{t-i})$
Slow stochastic oscillator (SSO)	$\frac{1}{m}\sum_{i=t-m+1}^{t}(MSO_{t-i})$
Rate of change (ROC)	x_t/x_{t-m}
Momentum (M)	x_t-x_{t-m}
Moving average (MA)	$\frac{1}{m}\sum_{i=t-m+1}^{t} x_i$
Moving variance (MV)	$\frac{1}{m}\sum_{i=t-m+1}^{t}(x_i - \overline{x}_t)^2$
Moving variance ratio (MVR)	MV_t^2 / MV_{t-m}^2
Exponential moving average (EMA)	$a \times x_t + (1 - a) \times x_{t-m}$
Moving average convergence & divergence (MACD)	$\sum_{i=t-m+1}^{t} EMA_{20}(i) - \sum_{i=t-m+1}^{t} EMA_{40}(i)$
Accumulation/ distribution oscillator (ADO)	$((x_l(m) - x_t)/(x_h(m) - x_l(m)))$
Disparity5 (D5)	x_t / MA_5
Disparity10 (D10)	x_t / MA_{10}
Price oscillator (OSCP)	$(MA_5 - MA_{10}) / MA_5$
Commodity channel index (CCI)	$(M_t - SM_t)/0.015D_t$ where $M_t = x_h(t) + x(t) + x_l(t)$, $SM_t = \sum_{i=t-m+i}^{t} M_i / m$, $D_t = \sum_{i=t-m+i}^{t} \lvert M_i - SM_t \rvert / m \cdot$
Relative strength index (RSI)	$100 - \frac{100}{1 + RS}$ where $RS = \frac{\sum_{i=t-m+1}^{t}(x(i) - x(i-1))^{+}}{\sum_{i=t-m+1}^{t}(x(i) - x(i-1))^{1}}$
Linear regression line (LRL)	$\frac{m \times \sum_{i=t-m+1}^{t} i \times x(i) - \sum_{i=t-m+1}^{t} i \times \sum_{i=t-m+1}^{t} x(i)}{m \times \sum_{i=t-m+1}^{t} i^2 - \left(\sum_{i=t-m+1}^{t} i\right)^2}$

In order to evaluate the forecasting ability of the proposed GASVM model, we compare its performance with those of conventional methods, such as statistical and time series model, and neural network model, as well as individual SVM model without GA preprocessing. Typically, we select random walk (RW) model, autoregressive integrated moving average (ARIMA) model, individual back-propagation neural network (BPNN) model and individual SVM model as the benchmarks.

For RW and ARIMA models, only the foreign exchange series P (i.e., price) is used. Each of the RW and ARIMA models is estimated and validated by in-sample data. The model estimation selection process is then followed by using an empirical evaluation, e.g., RMSE, which is based on the out-of-sample data.

For individual BPNN model, the BPNN has 18 input nodes because 18 input variables are employed. By trial and error, we select 36 hidden nodes. Training epochs are 5000. The learning rate is 0.25 and the momentum term is 0.30. The hidden nodes use sigmoid transfer function and the output node uses the linear transfer function.

Similarly, individual SVM model has also 18 input variables, the Gaussian radial basis function are used as the kernel function of SVM. In SVM model, there are two parameters, i.e., upper bound C and kernel parameter σ, to tune. By trial and error, the kernel parameter σ is 10 and the upper bound C is 70.

For GASVM model, the GA is firstly used as preprocessor for input variable selection. Then the reduced variables are sent to the SVM model for learning and forecasting. In this study, eleven variables, i.e., price (P), stochastic oscillator (SO), rate of change (ROC), moving average (MA), moving variance ratio (MVR), moving average convergence & divergence (MACD), disparity5 (D5), price oscillator (OSCP), commodity channel index (CCI), relative strength index (RSI) and linear regression line (LRL), are retained by GA-based variable selection. Accordingly, some other parameters settings are similar to the individual SVM model.

This study is to mine and explore the tendency of stock price index. They are categorized as “0” and “1” in the research data. “0” means that the next day’s index is lower than today’s index, and “1” means that the next day’s index is higher than today’s index. The entire data set covers the period from January 1 2000 to December 31 2004. The data sets are divided into two periods: the first period covers from January 1 2000 to December 31 2003 while the second period is from January 1 2004 to December 31 2004. The first period, which is assigned to in-sample estimation, is used to network learning and training. The second period is reserved for out-of-sample evaluation.

3.2 Experiment Results

Each of the models described in the last section is estimated and validated by in-sample data. The model estimation selection process is then followed by an empirical evaluation based on the out-of-sample data. At this stage, the relative performance of the models is measured by hit ratio. Table 2 reports the experimental results.

From Table 2, the differences between the different models are very significant. For example, for the GBP test case, the D_{stat} for the RW model is only 51.06%, for the ARIMA model it is 56.13%, and for the BPNN model D_{stat} is only 69.78%; while for the proposed GASVM forecasting model, D_{stat} reaches 84.57%, which is higher than the individual SVM, implying that the GA-based variable selection has a significant impact on SVM forecasting.

In addition, in the experiments, we also find that the GASVM computing is faster then BPNN and SVM. The main reason is that the GA-based variable selection procedure reduces the model input space and thus saves the training time and speeds the SVM learning. Therefore, the proposed GASVM model can have some comparative advantages relative to individual SVM and BPNN. First of all, there is less parameter to tune for GASVM than for BPNN. Second, the GASVM can overcome some shortcomings of BPNN, such as overfitting and local minima. Third, the input space of the GASVM is smaller, and the learning speed of GASVM is faster than the individual SVM model.

Table 2. The prediction performance comparison of various models

Mining models	Hit ratios (%)
RW	51.06
ARIMA	56.13
BPNN	69.78
SVM	78.65
GASVM	84.57

4 Conclusions

This study proposes using a GA-based SVM data mining model that combines the genetic algorithm and support vector machine to predict stock market tendency. In terms of the empirical results, we find that across different forecasting models for the test cases of S&P 500 on the basis of same criteria, the proposed GASVM model performs the best. In the proposed GASVM model test cases, the D_{stat} is the highest, indicating that the nonlinear ensemble forecasting model can be used as a viable alternative solution for mining stock market tendency.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China; Chinese Academy of Sciences; Key Laboratory of Management, Decision and Information Systems and SRG of City University of Hong Kong (No. 7001806).

References

1. Kamijo, K., Tanigawa, T.: Stock Price Pattern Recognition: A Recurrent Neural Network Approach. In: Proceedings of the International Joint Conference on Neural Networks, San Diego, CA (1990) 215-221
2. Yoon, Y., Swales, G.: Predicting Stock Price Performance: A Neural Network Approach. In: Proceedings of the 24th Annual Hawaii International Conference on System Sciences. Hawaii (1991) 156-162
3. Trippi, R.R., DeSieno, D.: Trading Equity Index Futures with a Neural Network. Journal of Portfolio Management 19 (1992) 309-317
4. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, San Francisco, CA (2000)
5. Kohara, K., Ishikawa, T., Fukuhara, Y., Nakamura.: Stock Price Prediction Using Prior Knowledge and Neural Networks. International Journal of Intelligent Systems in Accounting, Finance and Management 6 (1997) 11-22
6. Tsaih, R., Hsu, Y., Lai, C.C.: Forecasting S&P 500 Index Futures with a Hybrid AI system. Decision Support Systems 23 (1998) 161-174
7. Quah, T.S., Srinivasan, B.: Improving Returns on Stock Investment through Neural Network Selection. Expert Systems with Applications 17 (1999) 295-301

8. Kim, K., Han, I.: Genetic Algorithm Approach to Feature Discretization in Artificial Neural Networks for the Prediction of Stock Price Index. Expert Systems with Applications 19 (2000) 125-132
9. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with Artificial Neural Networks: the State of the Art. International Journal of Forecasting 14 (1998): 35-62
10. Hann, T.H., Steurer, E.: Much Ado about Nnothing? Exchange Rates Forecasting: Neural Networks vs. Linear Models Using Monthly and Weekly Data. Neurocomputing 10 (1996): 323-339
11. Yu, X.H.: Can Backpropagation Error Surface not have Local Minima? IEEE Transactions on Neural Networks 3 (1992): 1019–1021
12. Vapnik, V.N.: The Nature of Statistical Learning Theory. New York: Springer, (1995)
13. Kim, K.J.: Financial time series forecasting using support vector machines. Neurocomputing 55 (2003) 307-319
14. Huang, W., Nakamori, Y., Wang, S.Y.: Forecasting Stock Market Movement Direction with Support Vector Machine. Computers & Operations Research 32 (2005) 2513-2522
15. Holland, J.H.: Genetic Algorithms. Scientific American 267 (1992): 66-72
16. Goldberg, D.E.: Genetic Algorithm in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA (1989)
17. Kupinski, A.M., Giger, M.L.: Feature selection with limited datasets. Medical Physics 26 (1999): 2176-2182