Optik 125 (2014) 1439–1443

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Real estate price forecasting based on SVM optimized by PSO

Xibin Wang [a], Junhao Wen [a,b,*], Yihao Zhang [a], Yubiao Wang [a]

[a] College of Computer Science, Chongqing University, Chongqing 400030, China
[b] College of Software Engineering, Chongqing University, Chongqing 400030, China

ARTICLE INFO

Article history:
Received 15 April 2013
Accepted 1 September 2013

Keywords:
Real estate price forecasting
Support vector machine
Particle swarm optimization
Parameters optimization

ABSTRACT

The real estate market has a close relationship with us. It plays a very important role in economic development and people's fundamental needs. So, accurately forecasting the future real estate prices is very significant. Support vector machine (SVM) is a novel type of learning machine which has been proved to be available in solving the problems of limited sample learning, nonlinear regression, as well as, better to overcome the "curse of dimensionality". However, the selected parameters determine its learning and generalization. Thus, it is essential to determine the parameters of SVM. Compared to ant colony algorithm, grid algorithm, genetic algorithm, particle swarm optimization (PSO) is powerful and easy to implement. Therefore, in the study, real estate price forecasting by PSO and SVM is proposed in the paper, where PSO is chosen to determine the parameters of SVM. The real estate price forecasting cases are used to testify the forecasting performance of the proposed PSO–SVM model. The experimental results indicate that the proposed PSO–SVM model has good forecasting performance.

1. Introduction

With the rapid economic development, the real estate prices have been showing an upward trend. Forecasting the real estate prices exactly is vital to implement the macroscopic regulation efficiently under the market economy. So far, there are some studies on the quantitative analysis of real estate prices. Malpezzi (1999) [22] estimated house price changes for 133 US metropolitan areas using the time series and cross-section data regression analysis model, which was built based on housing price index and macroeconomic indicators. He believed that housing prices are not random changes, which can be partly predicted. Anglin [1] set up a VAR model to forecast Toronto real estate prices, with the predictors' three lags for the average real estate price growth rate, CPI, mortgage rates and unemployment rate. Case and Shiller [2] forecasted the real estate price and excess returns in the real estate market using the percentage change in real per capita income, real construction costs, adult population, marginal tax rate and housing starts. In addition, there are some scholars using gray model, BP neural network and regression method for quantitative analysis of real estate prices. Tough gray model can be constructed by only a few samples, it only depicts a monotonously increasing or decreasing process [3–5]. In the case of small samples, the neural network prediction accuracy is low, convergence speed is not ideal, and the hidden layer nodes are difficult to determine. Regression analysis method requires that premises and assumptions must be correct, and there would exist a question that simultaneously examine multiple dependent variables is impossible. Considering the characteristics of the data set(high dimension, nonlinear, small samples), a new model based on support vector machine(SVM), which is aiming at the existing problems in the above methods, was proposed in this paper for real estate price forecasting.

SVM is a type of machine learning method, which adopts the structure risk minimization principle. Based on the structured risk minimization principle, SVM seeks to minimize an upper bound of the generalization error instead of the empirical error as in conventional neural networks [6–8]. It can better solve the small samples, nonlinear, over learning, high dimension, local minima, and so on, and has high generalization. But its performance can be still affected, if the parameters selected improper. To obtain an optimal SVM forecasting model, it is important to choose a kernel function, set the kernel parameters and determine a soft margin constant C, ε-insensitive loss parameter [9,10]. Thus, the choice of the parameters has a heavy impact on the forecasting accuracy [9]. At present, many techniques have been used for the parameters optimization. The most common optimization techniques are ant colony algorithm method [11], grid algorithm method [12] and genetic algorithm [13]. But ant colony algorithm method has disadvantages such as initial pheromone scarcity, long-time searching and local best solution; grid algorithm method is computationally intensive, time consuming and low learning accuracy; genetic algorithm is operation complex and different issues need to design different crossover or mutation. The particle swarm optimization (PSO)

* Corresponding author at: College of Software Engineering, Chongqing University, Chongqing 400030, China.
E-mail address: Jhwen@cqu.edu.cn (J. Wen).

0030-4026/$ – see front matter © 2013 Elsevier GmbH. All rights reserved.
http://dx.doi.org/10.1016/j.ijleo.2013.09.017

was found to have the extensive capability of global optimization for its simple concept, easy implementation and fast convergence [14–16]. Therefore, instead of using the above algorithms, a new method, PSO is proposed to optimize the SVM parameters in this study.

2. The regression theory of SVM

SVM was originally used for classification but its principle was extended to the task of regression and forecast as well [17–19]. In this study, we focus on support vector regression (SVR) for regression and forecast.

The SVM model can be described as following:

Let the training set as: $\{(x_1, y_1), \ldots, (x_l, y_l)\} \in R^n \times R$.where x_i is the input vector, y_i is the output value and n is the total number of the sample data.

The regression model defines the relation between x_i and $f(x_i)$ as:

$$y = f(x_i) = (\omega \cdot x_i) + b \tag{1}$$

where ω is the weight vector, b is the threshold. In addition, the coefficients ω and b are estimated by the following linear optimization problem:

$$\min_{\omega, b, \xi^{(*)}} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \tag{2}$$

$$s.t \begin{cases} ((\omega \cdot x_i) + b) - y_i \le \varepsilon + \xi_i, & i = 1, \ldots, l \\ y_i - ((\omega \cdot x_i) + b) \le \varepsilon + \xi_i^*, & i = 1, \ldots, l \\ \xi_i^{(*)} \ge 0, & i = 1, \ldots, l \end{cases} \tag{3}$$

where $\xi^{(*)}$ is slack variable, C is punishment coefficient, ε is insensitive loss function. $\xi^{(*)}$ guarantees the satisfaction of constraint condition; C controls the equilibrium between the complexity of model and training error; ε is a preset constant that for controlling tube size. If ε set too small, then it is easy to lead to over learning, otherwise, it is easy to lead to the owe learning.

For nonlinear regression, assume that there is such a transform: $\phi : R^n \to H, x \mapsto \phi(x)$, making $K(x, x') = \phi(x) \cdot \phi(x')$, where $(\cdot)$ denotes inner product operation. When a kernel function $K(x, x')$ satisfies the Mercer condition, it corresponds to the inner product of a transform space according to the functional theory. Therefore, the nonlinear regression function can be determined:

$$y = f(x) = \sum_{i=1}^{l}(\bar{\alpha}_i^* - \bar{\alpha}_i)K(x_i, x) + \bar{b} \tag{4}$$

$$s.t. \begin{cases} \sum_{i=1}^{l}\left(\bar{\alpha}_i^* - \bar{\alpha}_i\right) = 0 \\ 0 \le \alpha_i^{(*)} \le C, i = 1, \ldots, l \end{cases} \tag{5}$$

where $\bar{\alpha}^{(*)} = (\bar{\alpha}_1, \bar{\alpha}_1^*, \ldots, \bar{\alpha}_l, \bar{\alpha}_l^*)^T$ is the solution.In regression, selecting different kernel functions will construct different regression models. The commonly used kernel functions are:

(1) Linear kernel function $K(x, x') = x \cdot x'$.
(2) Polynomial kernel function $K(x, x') = ((x \cdot x') + 1)^d$.
(3) Gaussian kernel function $K(x, x') = \exp(-\left\|x - x'\right\|^2 / \sigma^2)$.

In this study, we use Gaussian kernel function, where σ is the kernel parameter. The parameter σ precisely defines the structure of high dimensional feature space, thus controlling the complex nature of ultimate solution. In conclusion, it is found that the choice of C, ε and σ has a great influence on the performance of SVM regression estimation. Therefore, how to reasonably and quickly select the above parameters has become a key factor that affects the practical application effects and scope of SVM.

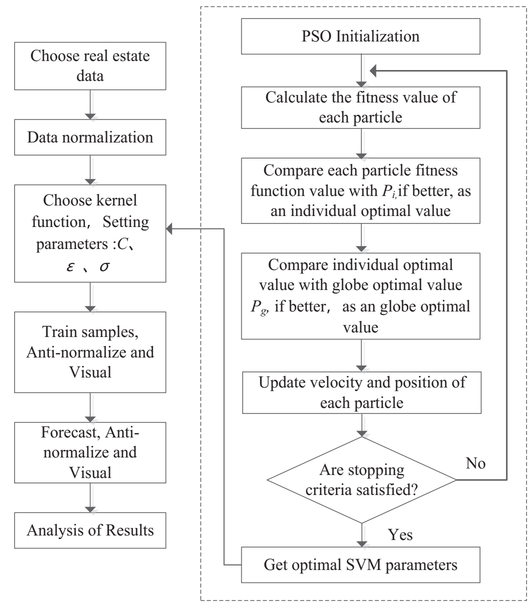

Fig. 1. Real estate price forecasting model based on PSO–SVM.

3. Real estate price forecasting model based on PSO–SVM

3.1. The principle of real estate price forecasting

The real estate price prediction can be seen as a mathematical mapping problem on pattern matching. The independent variables (here called "attribute" or "feature") are the factors that influence the change of price; the corresponding dependent variable is forecast price. Let the factors that influence the price change is a set of $\{x_1, \ldots, x_n\}$, the corresponding price is y. Then there is a nonlinear mapping f satisfies $y = f(x_1, \ldots, x_n)$ between the independent variables and the dependent variable.

Besides the supply and demand, the real estate price is influenced by many different factors like land price, development costs, the profits of developers, political aspects, and so on, which have led to the price nonlinear change. At the same time, the real estate price also changes with the economic development trend, and has a strong time-varying. So, using traditional linear forecasting model is difficult to obtain a better forecast results. Moreover, the nonlinear BP neural network is good at nonlinear fitting, but it is easy to get a local minimum without convergence, influencing the accuracy of prediction. However, the SVM has a strong nonlinear processing capability, which can be applied to modeling and forecasting based on the internal relationship of the historical data on real estate price.

In SVM regression model, eight features are extracted as input data of real estate, and the corresponding average price is used as regression output, thus the correlation coefficients of the regression function $y = f(x) = \sum_{i=1}^{l}(\bar{\alpha}_i^* - \bar{\alpha}_i)K(x_i, x) + \bar{b}$ can be determined. When forecasting the prices of other years, inputting the year feature vector, SVM will output the corresponding real estate price value according to the determined regression function. The process of PSO–SVM for forecasting the real estate price is shown in Fig. 1.

3.2. Preprocessing of real estate data

3.2.1. Real estate data normalization

Using SVM for real estate price forecasting, the training data and testing data need to be normalized. The main purpose of normalization is to avoid attributes in greater numerical ranges dominating those in smaller numerical ranges. Additionally, the normalization could avoid numerical difficulties during the calculation [20]. For a group of real estate data $\{x_{1k}, x_{2k}, \ldots, x_{ik}\}$, it is normalized as $\{X_{1k}, X_{2k}, \ldots, X_{ik}\}$ by the formula (6):

$$X_{ik} = \frac{x_{ik} - x_k^{\min}}{x_k^{\max} - x_k^{\min}} \tag{6}$$

where X_{ik} is the scaled value, x_{ik} is original value, $x_k^{\max}$ is the maximum value of feature k in the data set, $x_k^{\min}$ is the minimum value of feature k in the data set.

3.2.2. The construction of the training sample sets

For a group of real estate training data, the training sample sets are constructed, which is expressed as followings:

$$X = \begin{bmatrix} b_1 & b_2 & \ldots & b_m \\ b_2 & b_3 & \ldots & b_{m+1} \\ \vdots & \vdots & & \vdots \\ b_{n-m} & b_{n-m+1} & \ldots & b_{n-1} \end{bmatrix}, \quad Y = \begin{bmatrix} b_{m+1} \\ b_{m+2} \\ \vdots \\ b_n \end{bmatrix} \tag{7}$$

where X is the input vector, Y is the output vector, m is the dimension of the input vector.

3.3. The optimal SVM model by PSO

3.3.1. Particle swarm optimization

Particle swarm optimization is an evolutionary computation, which was developed by Kennedy and Eberhart [21]. As described by Eberhart and Kennedy, the PSO algorithm is an adaptive algorithm based on a social–psychological metaphor: a population of individuals (referred to as particles) adapts by returning stochastically toward previously successful regions [15].

The swarm consists of n particles; each particle has a position vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, and a velocity vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, where $i = 1,2,\ldots,n$. Each particle is represented as a potential solution to a problem in a D-dimensional search space. During each generation each particle is accelerated toward the particles previous best position and the global best position. Where, the best previously visited position of the ith particle denotes $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$; the best previously visited position of the swarm denotes $P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$. The new velocity value is then used to calculate the next position of the particle in the search space. This process will keep the iteration until setting the maximum number of iteration or a minimum error is achieved. The updating of velocity and particle position can be obtained by using the following formula:

$$v_{id}^{l+1} = w \times v_{id}^l + c_1 \times rd_1^l \times (p_{id}^l - x_{id}^l) + c_2 \times rd_2^l \times (p_{gd}^l - x_{id}^l) \tag{8}$$

$$x_{id}^{l+1} = v_{id}^{l+1} + x_{id}^l \tag{9}$$

where $i = 1,2,\ldots,n$, $d = 1,2,\ldots,D$; w denotes the inertial weight coefficient, whose value is nonnegative, and affect the overall optimization ability; c_1 and c_2 are learning factors (also called the acceleration constant); rd_1^l and rd_2^l are positive random number in the range [0,1] under normal distribution; l denotes the lth iteration; x_{id}^l is the position of the particle i in d-dimensional space, which denotes the current value of SVM parameters C, σ, and ε; $v_{id} \in [v_{\max}, v_{\min}]$ denotes the velocity of a particle i in d-dimensional space, which decides to update the direction and distance of the next generation of C, σ and ε.

Inertia weight w is employed to control the impact of the previous history of velocities on the current velocity. A larger inertia weight value facilitates the global exploration, while a small value tends to facilitate local exploration. In order to balance the global exploration and local exploration capability, we adopt linear decreasing inertia weight. Typically, $w(k)$ is reduced linearly with each iteration, from w_{start} to w_{end}. It can be described as the formula (10):

$$w(k) = w_{start} - k \times \frac{w_{start} - w_{end}}{T_{\max}} \tag{10}$$

where k is the current iteration number, T_{max} is the maximum number of iteration, w_{start} is the maximum value of inertia weight, w_{end} is the minimum value of inertia weight.

Meanwhile, the speed evolution equation with constriction factor χ is adopted to improve the convergence speed of the particle swarm algorithm that is formula (11):

$$v_{id}^{l+1} = \chi\{w \times v_{id}^l + c_1 \times rd_1^l \times (p_{id}^l - x_{id}^l) + c_2 \times rd_2^l \times (p_{gd}^l - x_{id}^l)\} \tag{11}$$

3.3.2. Optimize the parameters of SVM by PSO

The Gaussian kernel function is used to construct SVM regression model in this paper, then the width σ of the Gaussian kernel, the penalization parameter C, and the ε-insensitive parameter need to be determined. As particle swarm optimization algorithm not only has strong global search ability and but also helps to search for the optimum parameters quickly. Therefore, the PSO method is applied to determine the parameters of SVM, which is shown in Fig. 1. The process of optimizing the parameters of SVM by PSO is represented as follows:

Step 1: Read the real estate sample data, and the sample data is divided into two subsets: one subset is training set, the other subset is testing set. Then, preprocess the sample data.
Step 2: Initialize PSO: Initialize all particles. Initialize parameters of PSO algorithm including the velocity $V_i = (v_{i1}, v_{i2}, \ldots, v_{id})$ and position $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ of each particle. Set acceleration coefficient c_1 and c_2. Set particle dimension, max number of iteration $T_{\max}$, and fitness threshold *Acc*. rd_1 and rd_2 are the two random numbers with the range from 0 to 1.
Step 3: Set the values of P_i and P_g. Set the current optimal position of the particle i is $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, that is $P_i = X_i (i = 1, \ldots, n)$ and the optimal individual in group as the current P_g.
Step 4: Define and evaluate fitness function. The k-fold cross validation (where $k = 1$) method is used to evaluate fitness in this study, and mean absolute percent error (MAPE) is adopted as the fitness function. The fitness function is expressed as the following formula:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \left| \frac{y_i - f(x_i)}{y_i} \right| \tag{12}$$

where y_i is the actual value, $f(x_i)$ is the output value, n is the number of the selected sample data.
Calculate the current fitness function value MAPE_i of each particle. According to the fitness function value, particle history optimal value and the global optimal value to determine the best individual adapt position P_i and P_g. If the current position is better than the current best position P_i according to the fitness, the position will take the place of the previous P_i. Otherwise, P_i remains unchanged. Choose the minimum value P_i and compare it with the previous

Table 1
Forecasting results by PSO–SVM, SVM and BP/Yuan.

Case	Year	Actual value	PSO–SVM		SVM		BP	
			Forecasting value	Relative error (%)	Forecasting value	Relative error (%)	Forecasting value	Relative error (%)
Case1: Average selling price from 2007 to 2010	2007	2627	2731	−3.96	2780	−5.82	2820	−7.35
	2008	2785	2694	3.27	2621	5.89	2960	−6.28
	2009	3442	3413	0.84	3389	1.54	3656	−6.22
	2010	4281	4311	−0.70	4356	−1.75	4154	3.00
	MAPE (%)		2.19		3.75		5.71	
Case2: Average selling price of 2010(Q1, Q3)	2010(Q1)	3676	3795	−3.24	3826	−4.08	3864	−5.11
	2010(Q3)	4020	3963	1.42	4130	−2.74	4172	−3.78
	MAPE (%)		2.33		3.94		4.45	

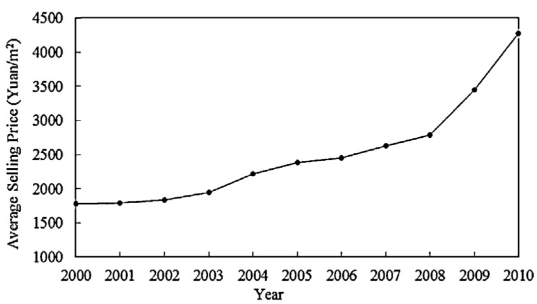

Fig. 2. The average selling price from 2000 to 2010.

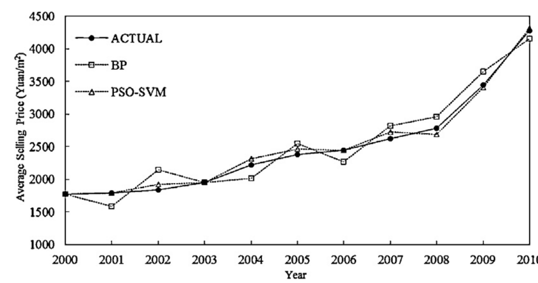

Fig. 4. Forecasting results by BP and PSO–SVM in case 1.

P_g value, if less than the previous value, then takes the place of the previous P_g value. Otherwise, P_g remains unchanged.
Step 5: Update velocity and position of each particle. Search for the better c, σ and ε according to the formula (9) and (11).
Step 6: Change the number of iteration. Let $t = t + 1$.
Step 7: Check stop condition. If $t > T_{max}$ or $MAPE_j < ACC$, then stop the iteration and P_g is the optimal solution which represents the best parameters for SVM. Otherwise, go to step 4.

Decoding the obtained optimal solution, get the optimized parameters.

4. Comparative experimental analysis

The average selling price of real estate of Chongqing (a city in China) from 2000 to 2010 and the average selling price of the first and the third quarter of 2008, 2009, and 2010 are used to study the real estate price forecasting performance of the proposed PSO–SVM model compared with BP. The real estate sample data are shown in Figs. 2 and 3, respectively. In order to experiment, the sample data should be divided into the two subsets: the training data and the testing data. In Fig. 2, the data from 2000 to 2006 are adopted as training data; the data from 2007 to 2010 are adopted as testing data (recorded as case 1). In Fig. 3, the data of the first quarter and the third quarter of 2008 and 2009 are adopted as training data; the data of the first quarter and the third quarter of 2010 are adopted as testing data (recorded as case 2). The forecasting accuracy is measured by relative error and MAPE.

In order to verify the importance of parameter optimization, as well as the feasibility and effectiveness of our proposed algorithm in parameter optimization, series of comparative experiments with other forecasting models are also given.

In PSO–SVM model, PSO is adopted to search for the optimal parameters of SVM; in SVM model, parameters are set to the default values. The forecasting values, relative error among PSO–SVM, SVM and BP are shown in Table 1. Table 1 displays in detail the forecasting accuracy comparison of our proposed forecasting model and the SVM model. From Table 1, it turns out that the forecasting accuracy of our proposed model is higher than SVM model, apparently showing that the PSO is an effective method for parameter optimization. It also shows that model parameters influence the forecasting performance of SVM evidently. So, the

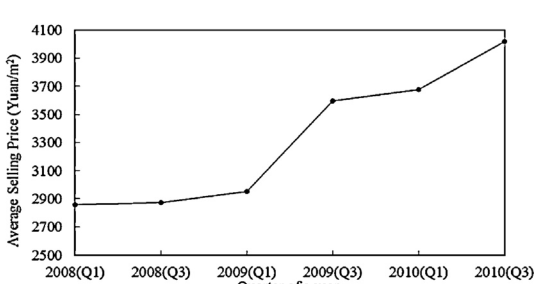

Fig. 3. The average selling price of the first quarter and the third quarter of 2008, 2009 and 2010.

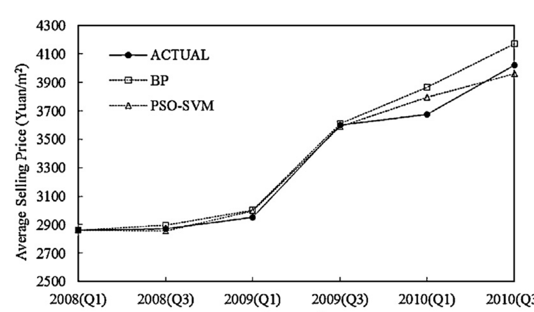

Fig. 5. Forecasting results by BP and PSO–SVM in case 2.

better algorithm will give the better combinational parameters of SVM.

In order to compare our proposed method with other approach, two experiments were conducted by using PSO–SVM model, SVM model, and BP neural network model. The comparative results are shown in Figs. 4 and 5 and Table 1.

Fig. 4 presents the comparison of the forecasting for the real estate data in case 1 by PSO–SVM and BP. Fig. 5 presents the comparison of the forecasting for the real estate data in case 2 by PSO–SVM and BP. In Figs. 4 and 5, compared with BP neural network, the simulation results of forecast show that the proposed method based on PSO–SVM is more accurate. From Table 1, it can be seen clearly that the forecast accuracy of SVM is better than the BP neural network, and the PSO–SVM is better than SVM. In fact, in the process of BP neural network forecast, the size of the training set has a greater impact on the training effect. When the training set is too small, it is difficult to efficiently fit the practical problem; otherwise, it will cause long training time and may occur over fitting. However, SVM obtains higher forecasting accuracy with the small samples. It can conclude that compared with BP neural network, SVM can better solve practical problems such as small samples, nonlinearity. In all, the experimental results indicate that PSO–SVM has higher forecasting accuracy than BP neural network in real estate price forecasting.

5. Conclusions and future work

In this paper, a novel real estate price forecasting model based on PSO and SVM is presented. Compared to ant colony algorithm, grid algorithm, and genetic algorithm, PSO is powerful and easy to implement. Thus, PSO is used to optimize parameters of SVM. The real estate sample data of Chongqing are adopted to study the forecasting performance of PSO–SVM model compared with BP neural network. The experimental results show that PSO–SVM has higher forecasting accuracy than BP neural network. As future work we will apply the proposed method to the larger data set, as well as carry out promotion and practical application. In addition, we will also focus on the research that with the expansion of the data scale, how PSO can maintain its good global search and convergence, and make the necessary improvements to improve the local search ability, so as to ensure the accuracy of the forecast within the acceptable range. Furthermore, the proposed algorithm will be applied to other data sets of our project.

Acknowledgements

This work was supported by National Science Foundations of China under Grant Nos. 61075053 and 71102065, the Ph.D. Programs Foundation of Ministry of Education of China No. 20120191110028, and the Fundamental Research Funds for the Central Universities Project No. CDJZR10090001.

References

[1] P. Anglin, Local dynamics and contagion in real estate markets, in: The International Conference on Real Estates and Macro Economy, Beijing, China, 2006, pp. 1–31.
[2] K.E. Case, R.J. Shiller, Forecasting prices and excess returns in the housing market, Real Estate Economics 18 (3) (2003) 253–273.
[3] J.C. Du, S.A. Cross, Cold in-place recycling pavement rutting prediction model using grey modeling method, Construction and Building Materials 21 (5) (2007) 921–927.
[4] C.C. Hsu, C.Y. Chen, Applications of improved grey prediction model for power demand forecasting, Energy Conversion and Management 44 (14) (2003) 2241–2249.
[5] W.Y. Wu, S.P. Chen, A prediction method using the grey model GMC(1,n) combined with the grey relational analysis: a case study on Internet access population forecast, Applied Mathematics and Computation 169 (1) (2005) 198–217.
[6] C. Gao, E. Bompard, R. Napoli, H. Cheng, Price forecast in the competitive electricity market by support vector machine, Physica A: Statistical Mechanics and its Applications 382 (1) (2007) 98–113.
[7] W.C. Hong, Hybrid evolutionary algorithms in a SVR-based electric load forecasting model, International Journal of Electrical Power & Energy Systems 31 (7–8) (2009) 409–417.
[8] C.J. Lu, T.S. Lee, C.C. Chiu, Financial time series forecasting using independent component analysis and support vector regression, Decision Support Systems 47 (2) (2009) 115–125.
[9] X. Liang, H. Zhang, J. Xiao, Y. Chen, Improving option price forecasts with neural networks and support vector regressions, Neurocomputing 72 (13–15) (2009) 3055–3065.
[10] C.C. Liu, K.W. Chuang, An outdoor time scenes simulation scheme based on support vector regression with radial basis function on DCT domain, Image and Vision Computing 27 (10) (2009) 1626–1636.
[11] Y. Li, J. Xia, S. Zhang, J. Yan, et al., An efficient intrusion detection system based on support vector machines and gradually feature removal method, Expert Systems with Applications 39 (1) (2012) 424–430.
[12] X. Shi, J. Zhou, Prediction residential house's damage effect near openpit against blasting vibration based on svm with grid searching method/genetic algorithm, Advanced Science Letters 11 (1) (2012) 238–243.
[13] E. Avci, Selecting of the optimal feature subset and kernel parameters in digital modulation classification by using hybrid genetic algorithm–support vector machines: HGASVM, Expert Systems with Applications 36 (2) (2009) 1391–1402.
[14] Y.H. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, 1999 CEC 99, in: Proceedings of the 1999 Congress on Evolutionary Computation, Indianapolis, USA, 1999, pp. 1945–1950.
[15] D.W. Boeringer, D.H. Werner, Particle swarm optimization versus genetic algorithms for phased array synthesis, IEEE Transactions on Antennas and Propagation 52 (3) (2004) 771–779.
[16] A. Subasi, Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders, Computers in Biology and Medicine 43 (5) (2012) 576–586.
[17] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing 14 (3) (2004) 199–222.
[18] U. Thissen, R. van Brakel, A.P. de Weijer, et al., Using support vector machines for time series prediction, Chemometrics and Intelligent Laboratory Systems 69 (1–2) (2003) 35–49.
[19] P.Y. Hao, New support vector algorithms with parametric insensitive/margin model, Neural networks: The Official Journal of the International Neural Network Society 23 (1) (2010) 60–73.
[20] H.L. Chen, D.Y. Liu, B. Yang, et al., A new hybrid method based on local fisher discriminant analysis and support vector machines for hepatitis disease diagnosis, Expert Systems with Applications 38 (9) (2011) 11796–11803.
[21] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942–1948.
[22] S. Malpezzi, A simple error correction model of housing price, Journal of Housing Economics 8 (1) (1999) 27–62.