

Research Article

Sales Growth Rate Forecasting Using Improved PSO and SVM

Xibin Wang,¹ Junhao Wen,^{1,2} Shafiq Alam,³ Xiang Gao,¹ Zhuo Jiang,¹ and Jun Zeng²

¹ College of Computer Science, Chongqing University, Chongqing 400030, China

² School of Software Engineering, Chongqing University, Chongqing 400030, China

³ Department of Computer Science, University of Auckland, Auckland 1010, New Zealand

Correspondence should be addressed to Junhao Wen; wjhcqu@cqu.edu.cn

Received 6 March 2014; Revised 31 May 2014; Accepted 3 June 2014; Published 24 June 2014

Academic Editor: Marcelo J. Colaço

Copyright © 2014 Xibin Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate forecast of the sales growth rate plays a decisive role in determining the amount of advertising investment. In this study, we present a preclassification and later regression based method optimized by improved particle swarm optimization (IPSO) for sales growth rate forecasting. We use support vector machine (SVM) as a classification model. The nonlinear relationship in sales growth rate forecasting is efficiently represented by SVM, while IPSO is optimizing the training parameters of SVM. IPSO addresses issues of traditional PSO, such as relapsing into local optimum, slow convergence speed, and low convergence precision in the later evolution. We performed two experiments; firstly, three classic benchmark functions are used to verify the validity of the IPSO algorithm against PSO. Having shown IPSO outperform PSO in convergence speed, precision, and escaping local optima, in our second experiment, we apply IPSO to the proposed model. The sales growth rate forecasting cases are used to testify the forecasting performance of proposed model. According to the requirements and industry knowledge, the sample data was first classified to obtain types of the test samples. Next, the values of the test samples were forecast using the SVM regression algorithm. The experimental results demonstrate that the proposed model has good forecasting performance.

1. Introduction

Advertising investment and sales growth rate are interrelated. Understanding the relationship between these two, and forecasting the “sales growth rate” correctly, is very important for efficient and effective advertising investment under the market economy. Developing sales growth rate forecasting model is nontrivial due to its uncertain, nonlinear, dynamic, and complicated characteristics. Some recent and most commonly used forecasting models are neural network based prediction model [1], multiple linear regression analysis model [2], and grey forecasting model [3]. However, these models have their own weaknesses. For example, the neural network based model converges to the locally optimal solutions, which has a negative influence on forecasting results. Multiple linear regression analysis requires correct premises and assumptions and simultaneous examination of multiple dependent variables, which is not trivial. Although grey forecasting model can be constructed by taking into consideration only a few samples, yet it only depicts a monotonously increasing or decreasing process [4], which is

not what the sales growth behavior looks like. In order to overcome the above problems, it is important to look for a new forecasting method to forecast sales growth rate.

Support vector machine (SVM) is a novel machine learning method based on statistical learning theory, which has a good generalization capability for small training samples and yields higher accuracy [5, 6]. SVM has been successfully applied in different fields such as real estate price forecasting [7], face recognition [8], the business failure prediction [9], face detection [10], EMG signals classification for diagnosis of neuromuscular disorders [11], residential house's damage effect near Open-pit against blasting vibration prediction [12], detecting top management fraud [13], microarray data classification [14], default prediction of the small and medium enterprises [15], and traffic flow prediction [16].

In this paper, the forecasting technique of preclassification and later regression is presented, which is effective and feasible for small samples regression and short-term prediction of time series. As the choice of the parameters heavily influences the forecasting accuracy, hence, to obtain an optimal SVM forecasting model, it is important to choose a good

kernel function, tune the kernel parameters, and determine a soft margin constant C and ε -insensitive loss parameter [17, 18]. Currently, techniques such as grid search (GS), genetic algorithms (GA), and particle swarm optimization (PSO) have been used for the parameters optimization [19, 20]. Compared with GA, particle swarm optimization was found to have the capability of global optimization, simplicity and ease of implementation [21]; however, the standard PSO has some demerits, such as relapsing into local optimum, slow convergence speed, and low convergence precision in the later evolution. In order to overcome the above shortcomings, we proposed and improved PSO (IPSO) technique, where evolution speed factor and aggregation degree factor of the swarm are introduced to improve the convergence speed, and the position-extreme strategy is used to avoid plunging into local optimum. In each iteration process, the inertia weight is changed dynamically based on the current evolution speed factor and aggregation degree factor, which makes the algorithm attain effective dynamic adaptability. Considering the above advantages, this study introduces the IPSO as an optimization technique to simultaneously optimize the SVM parameters.

Furthermore, to achieve a better forecast performance for sales growth rate, combining the IPSO with the “forecasting technique of preclassification and later regression,” we proposed the “regression model based on SVM classification” optimized by IPSO. In summary, the main contributions of this paper are as follows.

- (1) We design and implement a growth rate forecast model running on the sales growth rate forecasting cases. The model is extensible, being able to combine the knowledge discovered by SVM and the industry knowledge.
- (2) We proposed the IPSO to optimize the kernel function parameters for classification and regression. During each iteration, the inertia weight is changed dynamically based on the current evolution speed factor and aggregation degree factor, which provides the algorithm with effective dynamic adaptability. It has a better performance than the standard PSO in searching global optimum while resolving conflict between convergence and the global search for improved forecasting accuracy.
- (3) We first classified the sample data to decide the types of the testing samples, and then the values of the testing samples were predicted using the SVM regression algorithm. This limits the forecast samples in the same type range, reducing the forecasting range, and enhancing the forecast accuracy.
- (4) After the classification, the range of the samples is narrowed down and the trend of forecast is obtained based on these narrowed samples. Samples of the same types resulting from classification contain similar trends and help the model make full use of the data trend. Such capability is not present in the regression without preclassification.

The rest of this paper is organized as follows. Section 2 introduces the regression and classification theory of SVM and PSO algorithm while emphasizing on the analysis of PSO in detail. IPSO is introduced in Section 3 to overcome the premature convergence and the local optimum of PSO. The regression method based on SVM classification optimized by IPSO is presented in Section 4. In Section 5, firstly, we verify the validity of the proposed IPSO algorithm based on three classic benchmark functions; secondly, the IPSO is used to the regression model based on SVM classification and compared with the other three models. The results indicate that the IPSO has far superior performance to PSO in global optimization and convergence speed, and the proposed model has better performance than the other three models. Finally, the conclusions and future research suggestions are highlighted in Section 6.

2. Literature Review

2.1. The Regression and Classification Theory of Support Vector Machine (SVM). Support vector machine was originally used for classification but its principle was extended to the task of regression and forecast as well. The SVM classification model can be described as follows.

Let the training data set be $\{(x_i, y_i)\}$, where $x_i \in R^n$ are the input values and $y_i \in \{-1, +1\}$, $i = 1, 2, \dots, l$, are the forecasting values; the generalized linear SVM finds an optimal separating hyperplane:

$$f(x_i) = (\omega \cdot x_i) + b \quad (1)$$

by solving the following optimization problem:

$$\begin{aligned} \min_{\omega, b, \xi} \quad & \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^l \xi_i \\ \text{s.t.} \quad & y_i ((\omega \cdot x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0. \end{aligned} \quad (2)$$

This optimization model can be solved by introducing the Lagrange multipliers α_i for its dual optimization model. After the optimal solution α_i^* is obtained, the optimal hyperplane parameters ω^* and b^* can be determined, and the separating hyperplane can be described as follows:

$$f(x) = \text{sgn} [(\omega^* \cdot x + b^*)] = \text{sgn} \left(\sum_{i=1}^l \alpha_i^* y_i (x_i, x) + b^* \right). \quad (3)$$

For nonlinear classification, assume that there is a transform: $\phi : R^n \rightarrow H$, $x \mapsto \phi(x)$, making $K(x, x') = \phi(x) \cdot \phi(x')$, where (\cdot) denotes inner product operation. According to the functional theory, as long as a kernel function meets the Mercer condition, then it corresponds to the inner product of a transform space [22]. Therefore, the nonlinear classification function can be determined as follows:

$$f(x) = \text{sgn} \left(\sum_{i=1}^l \alpha_i^* y_i K(x_i, x) + b^* \right). \quad (4)$$

The SVM regression model can be described as follows.

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

The function $f(x_i)$ is represented by using a linear function in the feature space:

$$y = f(x_i) = (\omega \cdot x_i) + b, \quad (5)$$

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

$$\begin{aligned} \min_{\omega, b, \xi_i^*} \quad & \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ \text{s.t.} \quad & \begin{cases} ((\omega \cdot x_i) + b) - y_i \leq \varepsilon + \xi_i, & i = 1, 2, \dots, l \\ y_i - ((\omega \cdot x_i) + b) \leq \varepsilon + \xi_i^*, & i = 1, 2, \dots, l \\ \xi_i^*, \xi_i \geq 0, & i = 1, 2, \dots, l, \end{cases} \end{aligned} \quad (6)$$

where ξ_i and ξ_i^* are nonnegative slack variables which measure the deviation ($y_i - f(x_i)$), C is punishment coefficient, and ε is insensitive loss function. ξ_i^* guarantees the satisfaction of constraint condition; C controls the equilibrium between the complexity of model and training error; ε is a preset constant that controls tube size.

For nonlinear regression, introduction of kernel function and then nonlinear regression function can be determined:

$$\begin{aligned} y = f(x) = \sum_{i=1}^l (\bar{\alpha}_i^* - \bar{\alpha}_i) K(x_i, x) + \bar{b} \\ \text{s.t.} \quad \begin{cases} \sum_{i=1}^l (\bar{\alpha}_i^* - \bar{\alpha}_i) = 0 \\ 0 \leq \alpha_i^{(*)} \leq C, \quad i = 1, 2, \dots, l. \end{cases} \end{aligned} \quad (7)$$

The kernel function $K(x, x')$ typically has the following alternatives.

(1) Linear kernel function:

$$K(x, x') = x \cdot x'. \quad (8)$$

(2) Polynomial kernel of degree:

$$K(x, x') = ((x \cdot x') + 1)^d. \quad (9)$$

(3) Gauss kernel:

$$K(x, x') = \exp\left(\frac{-\|x - x'\|^2}{2\sigma^2}\right). \quad (10)$$

(4) Sigmoid kernel function:

$$K(x, x') = \tanh(kx \cdot x' - \sigma), \quad (11)$$

where σ is the kernel parameter, which denotes the width of Gauss kernel function and affects the complexity of the sample data distribution in the high space.

In this paper, we choose the Gauss kernel function as the kernel function of the classification model and the regression model. The objective functions are formulas (4) and (7), and the corresponding restrictions are described as above. In the classification model, the samples x_i are dependent variables and y_i are decision variables, representing the forecast type label. In regression model, the samples x_i are dependent variables and y_i are decision variables, representing the forecast values. In order to improve the performance of the classification model and the regression model, IPSO is used to optimize the parameters (C, σ) . For convenience, we use (c, g) to denote (C, σ) .

2.2. Particle Swarm Optimization (PSO). PSO is a meta-heuristic based on evolutionary computation, which was developed by Kennedy and Eberhart [23]. As described by Eberhart and Kennedy, the PSO algorithm is an adaptive algorithm based on a social-psychological metaphor; a population of individuals (referred as particles) adapts by returning stochastically toward previously successful regions [24, 25]. Below we provide a brief outline on the working of PSO.

In PSO, the swarm consists of n particles; $n \in N^*$ represents the number of particles; each particle has a position vector $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ and a velocity vector $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ [26, 27], where $i = 1, 2, \dots, n$. Particles representing a potential problem solution move through 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 particle i is denoted by $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, the best previously visited position of the swarm is denoted by $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$. 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 iterations or a minimum error is achieved. The updating of velocity and particle position can be obtained by using the following formulas:

$$\begin{aligned} v_{id}^{l+1} = w \times v_{id}^l + c_1 \times rd_1^l \times (p_{id}^l - x_{id}^l) \\ + c_2 \times rd_2^l \times (p_{gd}^l - x_{id}^l), \end{aligned} \quad (12)$$

$$x_{id}^{l+1} = v_{id}^{l+1} + x_{id}^l, \quad (13)$$

where $i = 1, 2, \dots, n$, $d = 1, 2, \dots, D$; w denotes the inertial weight coefficient; c_1, c_2 are learning factors; rd_1^l and rd_2^l are positive random number in the range under normal distribution; l denotes the l th iteration; x_{id}^l is the position of the particle i in d -dimensional space; $v_{id} \in [v_{\max}, v_{\min}]$ denotes the velocity of a particle i in d -dimensional space; x_{id}^{l+1} represents the position of the particle i at the $l + 1$ th iteration; and v_{id}^{l+1} denotes the movement vector of particle i at the $l + 1$ th iteration. Moreover, in formula (12), the first term $w \times v_{id}^l$ denotes the particle's inertia; the second term $c_1 \times rd_1^l \times (p_{id}^l - x_{id}^l)$ indicates the particle's cognition-only

model, and the third term $c_2 \times rd_2^l \times (p_{gd}^l - x_{id}^l)$ stands for the particle's social-only model.

More specifically, the training procedure for the PSO algorithm is briefly described as follows.

Step 1. Initialize all particles; initialize parameters of the PSO algorithm including the velocity $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$ and position $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ of each particle. Set the acceleration coefficient c_1 and c_2 , particle dimension, and the fitness threshold Acc . rd_1^l and rd_2^l are the two random numbers with the range from 0 to 1.

Step 2. Calculate the fitness values of all particles and store the P_i and P_g at the current iteration.

Step 3. If the number of iterations is terminated or the accuracy is satisfied, then output P_i and P_g positions, and the algorithm terminates. Otherwise, go to Step 4.

Step 4. For each particle, compare the current position and individual optimum, if better, then update. For each particle, compare the current position and the global optimum, if better, then update.

Step 5. Calculate the velocity vectors in formula (12) for all particles.

Step 6. Modify the positions of all particles utilizing formula (13) and then go to Step 2.

In PSO, inertia weight w is employed to control the impact of the previous history of velocities on the current velocity. A larger scale contributes to searching for the global optimal solution in an expansive area, fast convergence, but its precision is not good because of the rough search. The smaller scale improves the precision of the optimal solution, but the algorithm may be trapped in a local optimization. So, the balance between exploration and exploitation in PSO is dictated by w . Thus, proper control of the w is very important to search the optimal solution accurately and efficiently. To balance the global exploration and local exploration capability, some researchers adopt linearly decreasing inertia weight [28, 29]. Typically, $w_{\text{line}}(k)$ is reduced linearly with each iteration, from w_{start} to w_{end} . It can be described as follows:

$$w_{\text{line}}(k) = \frac{w_{\text{start}} - k \cdot (w_{\text{start}} - w_{\text{end}})}{T_{\text{max}}}, \quad (14)$$

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

3. Proposed Improved PSO (IPSO) Algorithm

3.1. Evolution Speed-Aggregation Degree Strategy. Analyzing the linearly decreasing inertia weight, it can be found that there are some problems in this method. Firstly, if better solution is detected during early evolution, the chances increase for quick convergence to the optimal solution; however,

the linearly decreasing inertia weight makes the algorithm converge slowly. Secondly, in the later evolution, with w decrease, it results in decline in global search capability of the algorithm, weakened diversity, and easily trapping into local optimum. In order to overcome the deficiencies of the linear weight, this paper adopts a nonlinearly descending inertia weight of PSO to balance the global and local exploration capability.

Let $f(P_g^i)$ be the i th generation best global position corresponding to the fitness function value and $f(P_g^{i-1})$ the $i-1$ th generation best global position corresponding to the fitness function value.

Definition 1 (evolution speed β). Consider

$$\beta = \frac{\min(f(P_g^i), f(P_g^{i-1}))}{\max(f(P_g^i), f(P_g^{i-1}))}, \quad (15)$$

where $\min()$ represents the minimum value function and $\max()$ represents the maximum value function.

According to the above assumptions and definition, it can be found that $0 < \beta \leq 1$. The parameter not only considers the algorithm iteration history, but also reflects the evolution of the particle swarm speed; that is, if the value of β is smaller, then the evolution speed is faster. After a certain number of iterations, the value of β remains 1, which determines that the algorithm has found the optimal solution.

Whether PSO algorithm is premature convergence or global convergence, the particles of the swarm will appear to be in "gather" phenomenon. This means that either all particles gathered at a particular position or gathered in a few specific positions. Therefore, another factor affecting the performance of the algorithm is the aggregation degree of the particles.

In iteration process, the best global position corresponding to the fitness function value $f(P_g^i)$ is always better than the current best position corresponding to the fitness function value $f(P_i^t)$ of each particle, because the current best position corresponding to the fitness function value will compare with the best global position corresponding to the fitness function value in each iteration and update the best global position if the current position is better. Particularly, if the current best position corresponding to the fitness function value is equal to the best global position corresponding to the fitness function value, then we consider the $f(P_g^i)$ is better than $f(P_i^t)$, and the best global position does not need to be updated. Let $f(P_i^t)$ be the t th generation best global position corresponding to the fitness function value, and the t th generation average fitness function value is described as follows:

$$f_{\text{average}}(t) = \frac{1}{N} \sum_{i=1}^N f(P_i^t). \quad (16)$$

Definition 2 (aggregation degree α). In our case the smaller the value of the fitness function is the better it gets. The aggregation degree is defined as follows:

$$\alpha = \frac{\min(f(P_g^i), f_{\text{average}}(t))}{\max(f(P_g^i), f_{\text{average}}(t))}. \quad (17)$$

Obviously, $0 < \alpha \leq 1$, which reflects the current level of aggregation of all particles and, to some extent, also reflects the diversity of the particles. Compared to the smaller value of α , for larger value of α , the aggregation degree of the swarm is higher, and the particle variability is lower. In particular, when $\alpha = 1$, all particles of the swarm are identical in properties. But if the algorithm falls into the local optimum, in this case, it will not be easy for the swarm to escape the local extreme point.

Based on the above discussion, we can obtain the nonlinearly inertia weight expression:

$$w_{\text{nonline}}(k) = w_{\text{start}} - A\alpha + B\beta, \quad (18)$$

where A is the weight of evolution speed and B is the weight of aggregation degree.

3.2. Position-Extreme Strategy. To make the algorithm escape the local optimum, we set the judgment condition, which is to change the global optimal values in the evolution process. If the global optimal value does not improve in k consecutive iterations, that is, $k > \text{limit}$, then we consider the algorithm trapped into local optima. In such a case, the search strategy of the particles will change so that the particles escape from local optimum and start exploring new positions. When the particles get into a new local optimum, the algorithm will choose the smaller local optimum from before and after two local optimum values, based on the smaller fitness function value priority principle, and then enter into the next update. This means that the particle's current local optimum value will be compared with the previously obtained local optimum value during each iteration, to obtain a new local optimum value. The corresponding update equations are described as follows:

$$\begin{aligned} x_{id}^{l+1} &= \text{rand}(0, 1) \cdot x_{id}^{l+1}, \\ P_i^{l+1} &= \text{rand}(0, 1) \cdot P_i^{l+1}, \end{aligned} \quad (19)$$

where $\text{rand}()$ is a random function and $\text{rand}(0, 1)$ is a random number between 0 and 1.

3.3. The Improved PSO Algorithm. According to the strategy mentioned above, the improved PSO algorithm can be summarized as follows.

Step 1 (initialize IPSO). Initialize all particles; initialize parameters of IPSO algorithm including the velocity $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$ and position $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ of each particle. Set the acceleration coefficient c_1 and c_2 , particle dimension, the maximum number of iterations T_{max} , the maximum number of consecutive times limit, the weight of

evolution speed A , the weight of aggregation degree B , the maximum value of inertia weight w_{start} , the minimum value of inertia weight w_{end} , and the fitness threshold Acc . rd_1^l and rd_2^l are the two random numbers ranging between 0 and 1. T is the current number of iterations.

Step 2 (set the values of P_i and P_g). Set the current optimal position of the particle i as $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$, that is, $P_i = X_i$ ($i = 1, 2, \dots, n$), and the optimal individual in group as the current P_g .

Step 3 (define and evaluate fitness function). For classification problems, Acc is defined as classification accuracy; that is,

$$\text{Acc} = \frac{\text{The number of correctly classified samples}}{\text{The total number of samples}}. \quad (20)$$

For the regression problems, Acc is defined as regression error (RMSE); that is,

$$\text{Acc} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (21)$$

where n is the number of the samples, y_i are the original values, and \hat{y}_i are the forecasting values.

Step 4 (update velocity and position of each particle). Search for the better kernel parameters according to formulas (12) and (13). And the inertia weight is changed dynamically based on the current evolution speed factor and aggregation degree factor, which is formulated as formula (18).

Step 5 (change the number of iterations). Let $t = t + 1$.

Step 6 (check stop condition). If $t > T_{\text{max}}$ or Fitness function value $< \text{Acc}$, then stop the iteration and P_g is the optimal solution which represents the best parameters for SVM. Otherwise, go to Step 7.

Step 7 (judge the global optimum vale unchanged in consecutive k times). If $k > \text{limit}$, then go to Step 8; otherwise, go to Step 3.

Step 8 (updated the position according to the new position formulas (19)). In this paper, the parameter combination (c, g) is the optimized object, which is taken as the input of IPSO. When the stop condition is met, IPSO will output the optimal parameter combination.

4. Sales Growth Rate Forecasting Model Based on SVM Classification Optimized by IPSO

Sales growth rate forecasting is a time series forecasting problem. The future sales growth rates are predicted based on the historical sales data. Firstly, the experimental data should be preprocessed to improve the efficiency and the precision of forecasting model. These include the selection of attributes and data normalization. Secondly, establish the classification

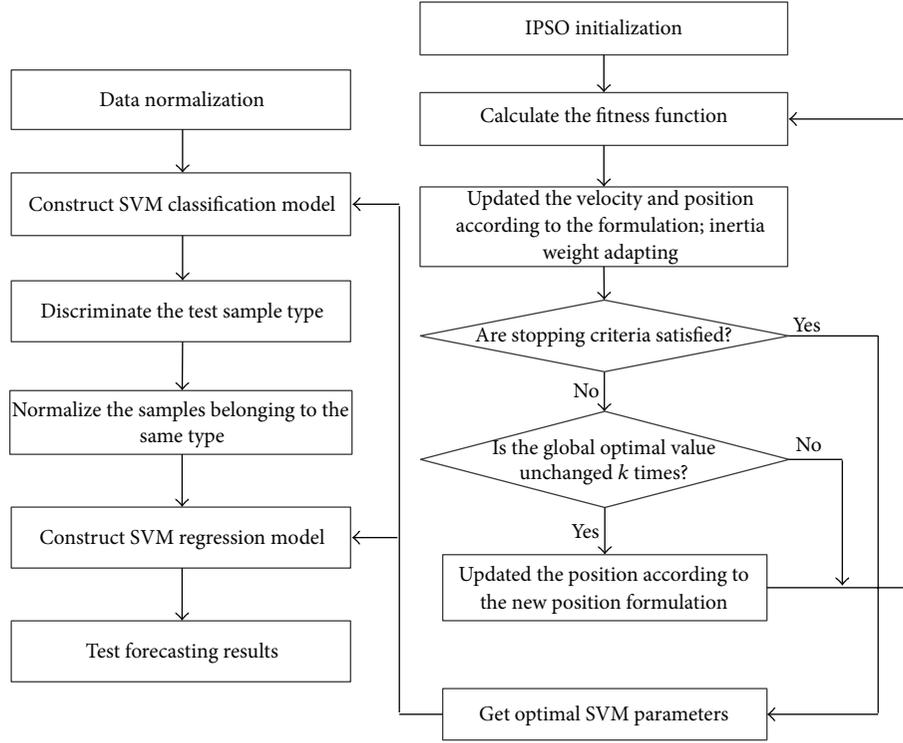


FIGURE 1: Regression model based on SVM classification optimized by IPSO.

model by training the sample data and obtain type label of the sample data. Then, construct the sales growth rate forecasting model, and train it on the sample data with the same type label. Thirdly, evaluate the model using root mean square error (RMSE) to validate the forecasting performance of the “regression method based on SVM classification” by using the sample data, which is shown in Figure 1.

4.1. The Preprocessing of Sales Data. In the preprocessing phase, first, the data is normalized. The main purpose of normalization is to avoid attributes in greater numerical ranges dominate those in smaller numerical ranges. In addition, the normalization could avoid numerical difficulties during the later calculation stages. The data is normalized according to the following formula:

$$X_{ik} = \frac{x_{ik} - x_k^{\min}}{x_k^{\max} - x_k^{\min}}, \quad (22)$$

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

Then, the training sample sets are constructed, which is expressed as follows:

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \\ x_2 & x_3 & \cdots & x_{m+1} \\ \vdots & \vdots & & \vdots \\ x_{n-m} & x_{n-m+1} & \cdots & x_{n-1} \end{bmatrix}, \quad Y = \begin{bmatrix} y_{m+1} \\ y_{m+2} \\ \vdots \\ y_n \end{bmatrix}, \quad (23)$$

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

4.2. Regression Method Based on SVM Classification Optimized by IPSO. In solving the nonlinear regression problem, to make full use of the advantages of SVM classification, we adopt the “regression method based on SVM classification.” As IPSO has far superior performance on global optimization and convergence speed, so IPSO is applied to determine the parameters of SVM, which is shown in Figure 1.

Let the training data set $K = \{(x_i, y_i) \mid i = 1, 2, \dots, k\}$ and the testing data set $S = \{(x_j, y_j) \mid j = 1, 2, \dots, s\}$, where $x_i \in R^n$ are input attributes and y_i are decision attributes. The basic steps of the “regression model based on SVM classification” optimized by IPSO are represented as follows.

Step 1. Divide K into M_p ($p = 1, 2, \dots, k$) types according to the practical application, where $K = \cup_{i=1}^k M_i$, $M_n \cap M_i = \emptyset$ ($i = 1, 2, \dots, k; n \neq i$), where M_p is a subset of data set K ; \cup represents the set operator “Union”; \cap represents the set operator “intersection”; and \emptyset represents the “Null set.”

Step 2. As a training set of K , the SVM classifier is generated and trained by training data.

Step 2.1. Normalize the sample data.

Step 2.2. Select the kernel function, and adopt IPSO to optimize the parameters.

Step 2.3 Train the normalized data and then get the SVM classification model.

Step 3. Using this classification model we can obtain the type label of the testing samples. Classify the testing samples and get the type label i of each sample (x_j, y_j) .

Step 4. For $M_p \in \text{type } i \wedge (x_j, y_j) \in \text{type } i$, M_p is training set; adopt SVM regression algorithm to forecast y_i value of each testing samples.

Step 4.1. Normalize M_p and (x_j, y_j) , which belong to the same type i .

Step 4.2. Select the kernel function, and adopt IPSO algorithm to optimize the parameters.

Step 4.3. Train the normalized training set, and establish SVM regression model.

Step 4.4. Utilize the training model to forecast y_i value of each testing samples.

5. Experimental Analysis

To evaluate the performance of the proposed IPSO algorithm and the regression method based on SVM classification, we conduct two numerical experiments. Firstly, experiment 1 validates the proposed IPSO algorithm based on three classic benchmark functions. Secondly, we verify the effectiveness and feasibility of the regression method based on SVM classification in experiment 2. We use the sales growth rate data set for this purpose. In experiment 2, Gaussian kernel function was selected as kernel function of SVM.

5.1. Experiment 1 (IPSO versus PSO)

5.1.1. The Classic Benchmark Functions. In order to compare the performance of IPSO and standard PSO, three classic benchmark functions are considered in our experiment, namely, Sphere function, Rosenbrock function, and Rastrigin function. The selection of these functions is based on the need of having slightly diverse functions to avoid any bias in selection. Sphere function is a unimodal quadratic function; Rosenbrock function is a unimodal function, which is difficult to minimize; and Rastrigin function is a multimodal function having a large number of local optimum. Three classic benchmark functions are detailed as follows.

(1) Sphere function:

$$f_1(x) = \sum_{i=1}^N x_i^2, \quad (24)$$

where $x_i \in [-100, 100]$ and the theoretical optimal is zero;

(2) Rosenbrock function:

$$f_2(x) = \sum_{i=1}^N \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right], \quad (25)$$

where $x_i \in [-10, 10]$ and the theoretical optimal is zero.

(3) Rastrigin function:

$$f_3(x) = \sum_{i=1}^N \left[x_i^2 - 10 \cos 2\pi x_i + 10 \right], \quad (26)$$

where $x_i \in [-5.12, 5.12]$ and the theoretical optimal is zero.

According to the characteristics of the functions, we know that $f_1(x)$ and $f_2(x)$ are unimodal functions; there is only one optimum in their domain, which is used to test the optimization precision and execution performance of the IPSO algorithm; $f_3(x)$ is a multimodal function, there are many local optima in its domain, which is used to test the global search ability and the ability to avoid premature of the IPSO algorithm.

5.1.2. Experiment 1 (Comparative Analysis of Algorithm Performance). In our first experiment, the classic benchmark functions are set to the fitness function of the particles. To eliminate causal factors, each function optimization experiments are run 10 times, and finally calculate the average. The algorithm parameters are set as follows: size of particle $N = 20$, the maximum number of iterations $T_{\max} = 300$, the initial inertia weight $w_{\text{start}} = 0.9$, the learning factor c_1 and $c_2 = 1.5$, the maximum number of the optimal value consecutive same times limit = 145, the evolution speed factor $A = 0.6$, and the aggregation degree factor $B = 0.05$. In iteration process, the inertia weight is adaptively adjusted depending on the fitness function values. The termination condition of iterations is when the fitness function value achieves convergence condition or reaches the maximum number of iteration.

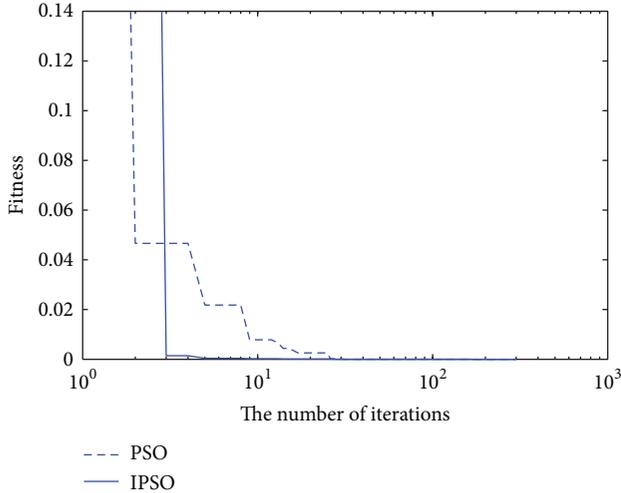
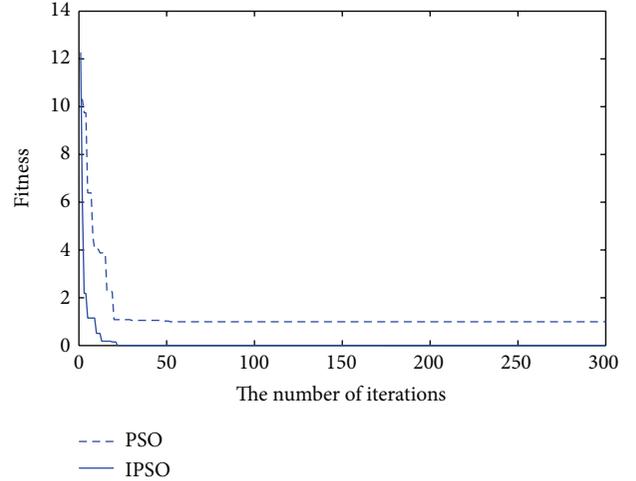
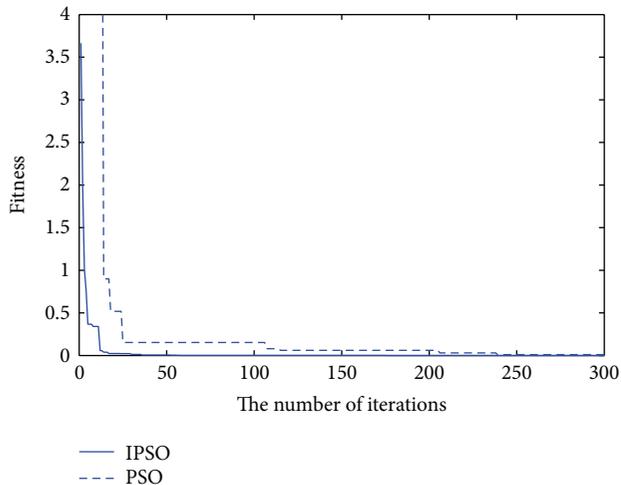
Table 1 shows that, for all functions, IPSO algorithm optimization results significantly better than standard PSO algorithm, and the average iteration time is significantly reduced; that is, IPSO algorithm can significantly improve the convergence speed of the particles. We observed that for the unimodal functions, the standard PSO algorithm can also get the theoretical optimum, but as a whole, the robustness of the algorithm is poor.

Figures 2, 3, and 4 demonstrate the above experiments. For Figure 2, in order for better comparison between IPSO and PSO, the horizontal axis uses the log-scale, and the maximum number of iterations is also set at 300. From the figures we can know that when solving $f_1(x)$, the performance of the two algorithms is similar, and both of them can converge to the global optimum, but IPSO algorithm converges faster, requires less iteration, and has higher efficiency; when solving $f_2(x)$, two algorithms can converge to the global optimum, but IPSO algorithm has an obvious advantage in convergence rate; when solving $f_3(x)$, PSO algorithm traps into local optimum and is difficult to find the global minimum point. But IPSO algorithm can converge to the global optimum in a short time and has the strong optimization capability. Overall, IPSO outperformed the traditional PSO algorithm on the selected functions, providing basis to be used for the optimization of SVM parameters.

We further analyze the solution of the function $f_3(x)$ due to its function characteristics. The PSO algorithm easily falls

TABLE 1: Comparison of IPSO and PSO on the three classic benchmark functions.

Test function	IPSO		PSO	
	Mean best fitness	Average time/ms	Mean best fitness	Average time/ms
$f_1(x)$	2.16×10^{-4}	150.4	2.81×10^{-4}	312.5
$f_2(x)$	0.81×10^{-1}	421.6	11.3×10^{-1}	539.2
$f_3(x)$	0.48	327.2	25.3	565.1

FIGURE 2: The evolution curve of function $f_1(x)$.FIGURE 4: The evolution curve of function $f_3(x)$.FIGURE 3: The evolution curve of function $f_2(x)$.

into local optimum, resulting in slow convergence speed and even stagnation. From Figure 4, we can find that the PSO algorithm traps into the local optimum when the number of the function evolution generations approximately equals to 25 and appears stagnation phenomenon. However, IPSO algorithm can jump out of the local extreme point and quickly find the optimal solution. Because of the adoption of the evolution speed factor and the aggregation degree factor, the inertia weight w can adaptively adjust according to the actual situation of the PSO iteration, resulting in

improved search capability and convergence speed of the algorithm. In addition, the position-extreme strategy can avoid the algorithm plunging into local optimum. Therefore, the evolution speed factor, the aggregation degree factor, and the position-extreme strategy can effectively improve the performance of PSO algorithm.

5.2. Experiment 2 (Validate the Effectiveness and Feasibility of the “Regression Method Based on SVM Classification” Optimized by IPSO)

5.2.1. Data Set.

To study the relationship between the advertising investment and the sales growth rate, we chose the advertising investment and the sales growth rate historical data. We then applied the regression model to forecast the trend of the sales growth rate. The forecasting dates are from 2012(Q1) to 2012(Q4), with 4 groups of data. As we provide short term forecasting, the data far from forecasting date provide less useful information to forecasting value; therefore, instead, we select 16 groups of data from 2008(Q1) to 2011(Q4) as input to construct and train forecast model. Then, we carry out forecasting and compare the results with actual data.

The sales growth rate is subject to many influences, such as TV advertising, Loushu, radio advertising (RA), folding, poster, leaflets, direct mail (DM), newspaper advertisements, and panels, and there may exist correlation between these attributes. So, based on needs and attribute correlation analysis, we select the input attributes as follows: Loushu, folding, poster, leaflets, DM, newspaper advertisements, panels,

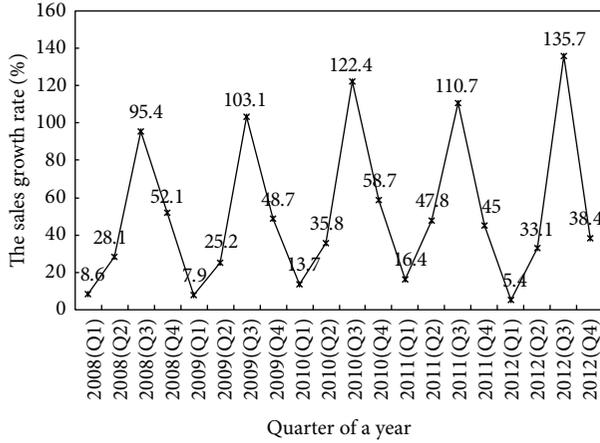


FIGURE 5: The sales growth rate (%) in the quarter of a year from 2008 to 2012.

walled packaging, sales offices packaging, slogan, SMS, and TV advertising. The sales growth rate is decision attribute. The data of the sales growth rate for modeling is shown in Figure 5.

As Figure 5 illustrates, the sales growth rate changes with the seasons and its change extent is large. So, the preclassification according to the sales growth rate and later regression can improve the forecasting accuracy, because the classification narrowing the range of the sample data and the regression is based on similar homogenous samples. In the following sections, we will build the “regression method based on SVM classification” to forecast the unknown trends.

5.2.2. Establishing the Classification Model. According to the requirement analysis, the data from 2008(Q1) to 2011(Q4) are adopted as the training data; and the data from 2012(Q1) to 2012(Q4) are adopted as the testing data. The test samples are divided into three types in accordance with the sales growth rate and the industry knowledge: type I, low growth: $0 \leq Y < 30.0\%$; type II, normal growth: $30.0\% \leq Y < 60.0\%$; type III, high growth: $Y \geq 60.0\%$, which are described in Table 2.

In Table 2, the first column represents the time when the statistics were collected; for example, 2008(Q1) represents the first quarter of 2008; the second column represents the original types corresponding to statistical time; the third column represents the forecast types, which are obtained by using the SVM classification model. The normal type indicates the results of correct classification. The bold type indicates the results of misclassification.

From Table 2, it can be seen clearly that the overall forecasting accuracy of the sample is 90%, that is, 2 misjudgments out of 20, where the accuracy of the training samples is 93.75% and the accuracy of the testing samples is 75%.

For forecasting the sales growth rate, Gaussian kernel function is chosen as the kernel function in classification and regression stages. First of all, we construct the classification model and train the model on the training set and build the classification model with the minimum classification error of cross-validation (CV). Secondly, we evaluate the accomplished performance on classification model by testing

TABLE 2: Classification results of the sample data.

Quarter of a year	Original type	Forecast type
2008(Q1)	I	I
2008(Q2)	I	II
2008(Q3)	III	III
2008(Q4)	II	II
2009(Q1)	I	I
2009(Q2)	I	I
2009(Q3)	III	III
2009(Q4)	II	II
2010(Q1)	I	I
2010(Q2)	II	II
2010(Q3)	III	III
2010(Q4)	II	II
2011(Q1)	I	I
2011(Q2)	II	II
2011(Q3)	III	III
2011(Q4)	II	II
2012(Q1)	I	I
2012(Q2)	II	II
2012(Q3)	III	III
2012(Q4)	II	I

the trained model over the test data set. The classification results of the testing data set are shown in Table 2 and Figure 6. Thirdly, we build the regression model based on preclassification and analyzed it. Finally, we assess the validity of the regression results.

Figure 6 shows the adoption of the SVM classification model to forecast the type labels from 2012(Q1) to 2012(Q4). The “red box” represents the forecasting classification label, and the “blue star” represents the actual classification label. If the forecasting labels of the classification agree with the actual classification labels, then we obtain the correct results of the classification. From Figure 6, it can be seen that for the four data classifications, one is misclassification, which corresponds to the “red box,” depicting the absence of overlap between accurate and misclassified data.

In training stage, we adopt classification accuracy rate as the fitness function. Besides, as cross validation (CV) is the preferred procedure in testing the out-of-sample classification capability when the dataset size is small [30], so CV is adopted in this paper to avoid experimental bias.

5.2.3. Comparison with Other Models. On the basis of the classification, we set up the SVM regression model (i.e., the “regression model based on classification”) and perform regression analysis using the sample data. In regression based on preclassification, IPSO is adopted to optimize the kernel parameters during two phases, initially during the classification and later during regression analysis. The parameters optimization results by IPSO are shown in Figure 7. From Figure 7, it can be seen clearly that after 100 iterations, IPSO obtains the optimal parameter combination ($c = 2.9439$,

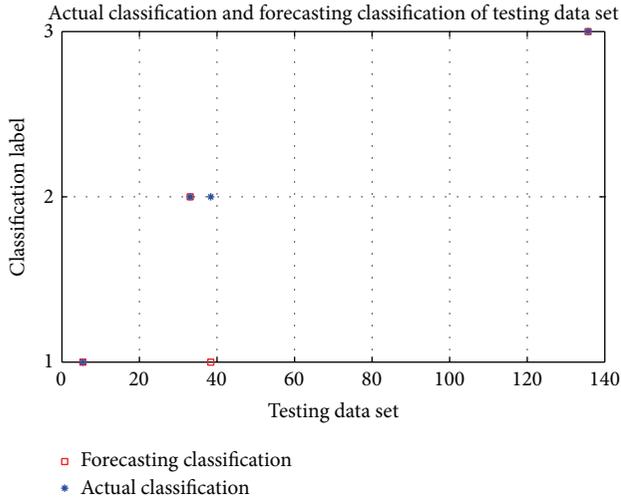


FIGURE 6: The classification results of testing data set.

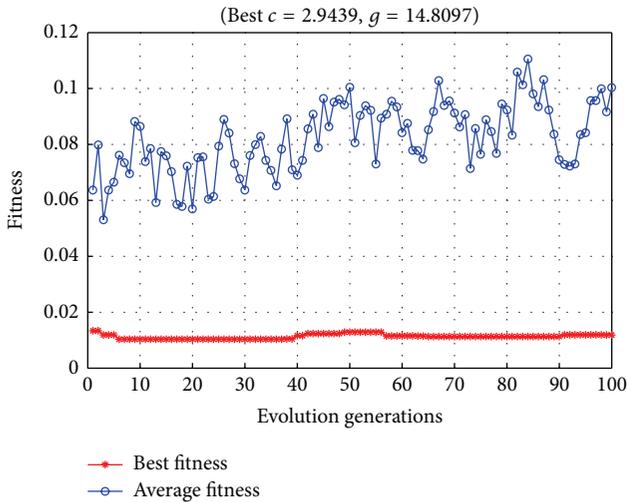


FIGURE 7: The fitness of selecting the optimal parameters by IPSO.

$g = 14.8097$), and also seen the best fitness value and the average fitness value of particles in each iteration.

Additionally, we also use genetic algorithms (GA) and grid search (GS) for comparison purpose on the same sample data against the results obtained by IPSO in selecting the optimal SVM parameters. The optimal parameters optimization result by GA is shown in Figure 8. The figure shows that after the iterations count reaches 100, GA obtains the optimal parameter combination ($c = 7.1426, g = 2.1765$). At the same time, we also see the best fitness value and the average fitness value of particles in each iteration. The optimal parameters optimization result by GS is shown in Figure 9. Figure 9 shows that after 100 iterations GS obtains the optimal parameter combination ($c = 64, g = 0.17678$). Meanwhile, we can observe that the fitness value of the particles keeps on changing under different parameter combinations.

In these experiments, we adopt the n -CV in PSO, GA, and GS as tuning parameters. Different vales of n correspond

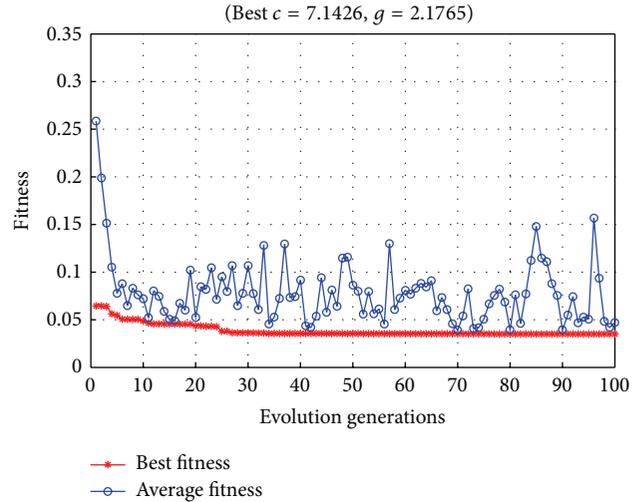


FIGURE 8: The fitness of selecting the optimal parameters by GA.

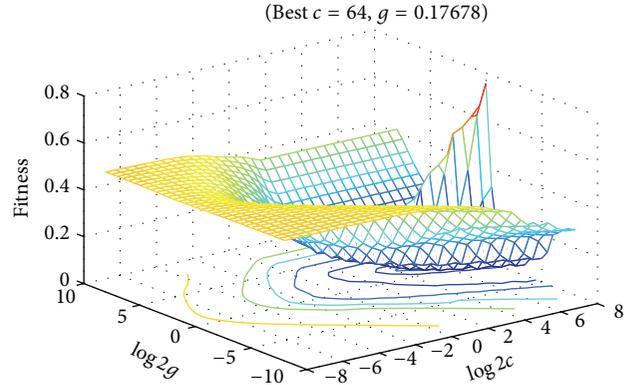


FIGURE 9: The fitness of selecting the optimal parameters by GS.

to the different training models. In order to obtain a better training model, we used the different values of n to train the models. The obtained fitness values that correspond to the different values of n are shown in Table 3. In Table 3, the bold values indicate the best fitness compared to the other two methods with the same value of n . For example, when $n = 2$, that is, 2-CV, the RMSE of IPSO is 1.25, the RMSE of GA is 3.92, and the RMSE of GS is 4.63. In particular, for PSO algorithm, when $n = 4$, the algorithm gets the minimum error in parameter optimization; for GA algorithm, when $n = 5$, the algorithm obtains the minimum error in parameter optimization; for GS algorithm, on the whole, the error is larger than the former two methods, because there is no attribute selection. Last row of Table 3 reports the average values of the fitness for $n = 2, 3, 4, 5$. It can be seen that IPSO has the minimum average fitness value, that is, generally, using the IPSO to get the best fitness value as compared to the other two methods for the given data set.

Table 3 and Figures 7–9 show the performance of three approaches in determining the optimal parameters. It is concluded that IPSO is superior to the other two approaches in terms of overall fitness.

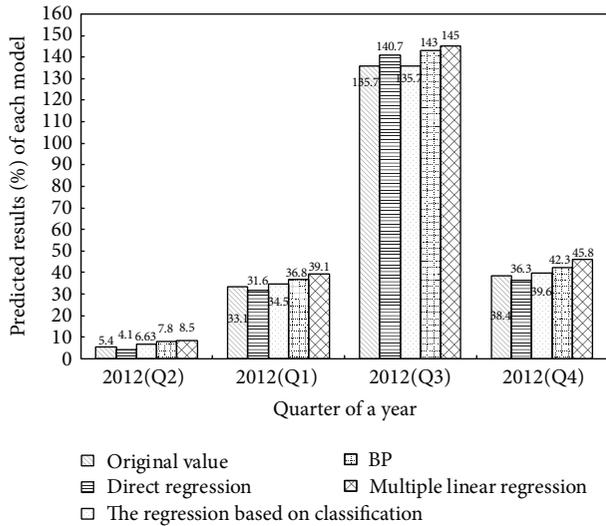


FIGURE 10: The forecasting results of the growth rate by four kinds of models.

TABLE 3: The RMSE (%) of different optimization algorithms.

	IPSO	GA	GS (without attribute selection)
CV-2	1.25	3.92	4.63
CV-3	1.26	3.89	4.83
CV-4	1.20	4.37	4.42
CV-5	1.23	3.51	4.59
Average	1.24	3.92	4.62

To validate our proposed model and provide a comparison, in this study, we have also used three other forecasting models, namely, (SVM) direct regression model, multiple linear regression model, and BP neural network model. The forecasting values and RMSE among the regression model based on classification, (SVM) direct regression model, BP neural network, and multiple linear regression are shown in Figures 10 and 11. Figure 10 shows the results of using four models forecasting the sales growth rate from 2012(Q1) to 2012(Q4). Figure 11 illustrates the error between predicted and actual values and the average error. The figures show that the regression model based on classification has outperformed its competitors in forecasting sales growth rate.

5.2.4. *Analysis and Discussion.* We proposed the forecasting technique of preclassification and later regression based on SVM model. The above experimental results demonstrate that the proposed method is effective and feasible generally and outstanding in small samples regression and short-term prediction of the time series. The general findings are as follows.

- (1) SVM maps input vectors nonlinearly into a high-dimensional feature space and construct the optimum separating hyperplane to realize the classification. Exploiting this characteristic, the “regression model based on SVM classification” can obtain higher forecast accuracy, even though the classification exist

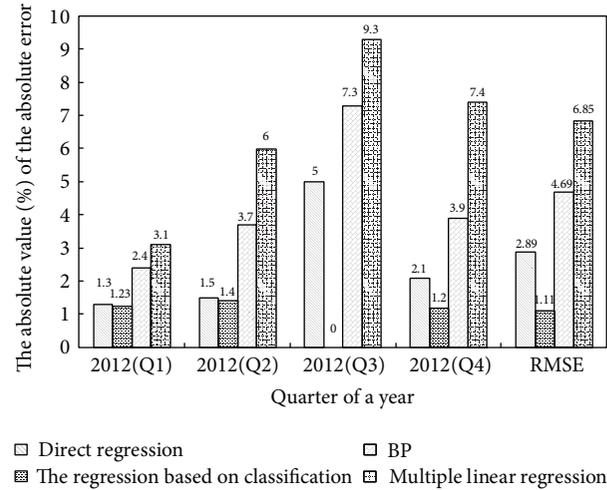


FIGURE 11: Comparison of absolute error forecasting among four kinds of models.

error, because SVM has higher classification accuracy, and the forecast is defined on the samples with the same class label, which have the same or similar change trend.

- (2) According to the change characteristics of the sample data, the actual requirements and industry knowledge, the sample cluster was classified first to decide the types of the test samples, which are divided into three types: type I: low growth; type II: normal growth; type III: high growth. This method limits the forecast samples in the same type range and makes full use of the samples’ change trend, reducing the forecasting range (or number), which makes the forecast accuracy higher.
- (3) After the classification, the range (or number) of the samples is narrowed, and the overall trend of forecast is weaker than the overall regression. So, in relation to the time series medium-term or long-term prediction, this method has some limitations. In order to compensate for this shortcoming, add new training samples according to the time series, to improve the gradient and the time-effectiveness of training samples.

6. Conclusions

In this study, the forecasting technique of preclassification and later regression is proposed, based on the classification of SVM for sales growth rate forecasting. We propose IPSO to optimize the parameters of SVM. We observed that the inertia weight w has a great impact on the convergence speed and accuracy of the PSO algorithm. As the linearly weight decreasing strategy does not make good use of the particles’ iteration and cause disadvantages, so we propose IPSO algorithm for performance improvement. We introduce two factors in this algorithm; the evolution speed factor and the aggregation degree factor to balance the global search and

the local search capability. This results in better convergence of the algorithm. Meanwhile, to avoid the algorithm plunging into local optimum, the position-extreme strategy is adopted.

To assess the performance of the IPSO, we used three classic benchmark functions. The results indicate that the IPSO algorithm converges faster and has better searching capability. We then applied IPSO to the regression model based on SVM classification. We used the sales growth rate data set for our experimental study and categorized them into three categories of low growth, normal growth, and high growth to assess the forecasting ability of the regression model based on SVM classification.

For comparison, we selected three relevant models, that is, the direct regression model, BP neural network, and multiple linear regression model. The experimental results indicate that forecasting accuracy of our proposed model for sales growth rate is much better than the other three forecasting models in sales growth rate forecasting.

In the future, we intend to generalize our model and extend its application to more real-world data. Furthermore, we aim to study the effects of different advertising investment strategies on the sales growth rate from microscopic perspective, so that the advertising investment can be optimized for optimal allocation of resources and maximize growth rate.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Basic Research Program of China (973) (2013CB328903), the National Science Foundations of China under Grant nos. 61075053, 61379158, 71301177, 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. CDJZR 10090001.

References

- [1] Z. Xiao, S.-J. Ye, B. Zhong, and C.-X. Sun, "BP neural network with rough set for short term load forecasting," *Expert Systems with Applications*, vol. 36, no. 1, pp. 273–279, 2009.
- [2] S. Yang, Y. Huang, and Q. Luan, "Research on aesthetic education in school physical education based on multiple linear regression method," in *Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012*, vol. 220 of *Lecture Notes in Electrical Engineering*, pp. 547–553, Springer, London, UK, 2013.
- [3] L. Wu, S. Liu, L. Yao, and S. Yan, "The effect of sample size on the grey system model," *Applied Mathematical Modelling*, vol. 37, no. 9, pp. 6577–6583, 2013.
- [4] J.-C. Du and S. A. Cross, "Cold in-place recycling pavement rutting prediction model using grey modeling method," *Construction and Building Materials*, vol. 21, no. 5, pp. 921–927, 2007.
- [5] T. Kavzoglu and I. Colkesen, "A kernel functions analysis for support vector machines for land cover classification," *International Journal of Applied Earth Observation and Geoinformation*, vol. 11, no. 5, pp. 352–359, 2009.
- [6] Y. B. Dibike, S. Velickov, D. Solomatine, and M. B. Abbott, "Model induction with support vector machines: introduction and applications," *Journal of Computing in Civil Engineering*, vol. 15, no. 3, pp. 208–216, 2001.
- [7] X. B. Wang, J. H. Wen, Y. H. Zhang, and Y. B. Wang, "Real estate price forecasting based on SVM optimized by PSO," *International Journal for Light and Electron Optics*, vol. 125, no. 3, pp. 1439–1443, 2014.
- [8] X. Zhou, W. Jiang, Y. Tian, and Y. Shi, "Kernel subclass convex hull sample selection method for SVM on face recognition," *Neurocomputing*, vol. 73, no. 10–12, pp. 2234–2246, 2010.
- [9] F. Lin, C.-C. Yeh, and M.-Y. Lee, "The use of hybrid manifold learning and support vector machines in the prediction of business failure," *Knowledge-Based Systems*, vol. 24, no. 1, pp. 95–101, 2011.
- [10] L. Wang, T. Zhang, Z. Jia, and L. Ding, "Face detection algorithm based on hybrid Monte Carlo method and Bayesian support vector machine," *Concurrency Computation: Practice and Experience*, vol. 25, no. 9, pp. 1064–1072, 2013.
- [11] A. Subasi, "Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders," *Computers in Biology and Medicine*, vol. 43, no. 5, pp. 576–586, 2013.
- [12] X. Shi and 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*, vol. 11, no. 1, pp. 238–243, 2012.
- [13] P.-F. Pai, M.-F. Hsu, and M.-C. Wang, "A support vector machine-based model for detecting top management fraud," *Knowledge-Based Systems*, vol. 24, no. 2, pp. 314–321, 2011.
- [14] S. Yoon and S. Kim, "K-Top Scoring Pair Algorithm for feature selection in SVM with applications to microarray data classification," *Soft Computing*, vol. 14, no. 2, pp. 151–159, 2010.
- [15] H. S. Kim and S. Y. Sohn, "Support vector machines for default prediction of SMEs based on technology credit," *European Journal of Operational Research*, vol. 201, no. 3, pp. 838–846, 2010.
- [16] M. H. Zhang, Y. B. Zhen, G. L. Hui, and G. Chen, "Accurate multisteps traffic flow prediction based on SVM," *Mathematical Problems in Engineering*, vol. 2013, Article ID 418303, 8 pages, 2013.
- [17] C.-C. Liu and 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*, vol. 27, no. 10, pp. 1626–1636, 2009.
- [18] X. Liang, H. Zhang, J. Xiao, and Y. Chen, "Improving option price forecasts with neural networks and support vector regressions," *Neurocomputing*, vol. 72, no. 13–15, pp. 3055–3065, 2009.
- [19] G. Berti, "GrAL—the grid algorithms library," *Future Generation Computer Systems*, vol. 22, no. 1–2, pp. 110–122, 2006.
- [20] J. Gu, M. Zhu, and L. Jiang, "Housing price forecasting based on genetic algorithm and support vector machine," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3383–3386, 2011.
- [21] R. Hassan, B. Cohanaim, O. de Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, pp. 1138–1150, April 2005.

- [22] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [23] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.
- [24] S. Alam, G. Dobbie, P. Riddle, and M. A. Naeem, "Particle swarm optimization based hierarchical agglomerative clustering," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT '10)*, pp. 64–68, September 2010.
- [25] D. W. Boeringer and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 3, pp. 771–779, 2004.
- [26] S. Alam, G. Dobbie, and P. Riddle, "Exploiting swarm behaviour of simple agents for clustering web users' session data," in *Data Mining and Multi-Agent Integration*, Springer, New York, NY, USA, 2009.
- [27] A. Noorul Haq, K. Karthikeyan, K. Sivakumar, and R. Saravanan, "Particle swarm optimization (PSO) algorithm for optimal machining allocation of clutch assembly," *The International Journal of Advanced Manufacturing Technology*, vol. 27, no. 9-10, pp. 865–869, 2006.
- [28] W. Jin, J. Zhang, and X. Zhang, "Face recognition method based on support vector machine and particle swarm optimization," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4390–4393, 2011.
- [29] M. J. Abdi and D. Giveki, "Automatic detection of erythematous diseases using PSO-SVM based on association rules," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 603–608, 2013.
- [30] R. Johnson and D. Wichern, *Applied Multivariate Statistical Analysis*, Qatar University Press, Doha, Qatar, 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

