Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 320186, 15 pages
http://dx.doi.org/10.1155/2015/320186

Research Article

PMSVM: An Optimized Support Vector Machine Classification Algorithm Based on PCA and Multilevel Grid Search Methods

Yukai Yao,[1] Hongmei Cui,[2] Yang Liu,[3] Longjie Li,[1] Long Zhang,[1] and Xiaoyun Chen[1]

[1] *School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China*
[2] *Gansu Provincial Maternity and Child-Care Hospital, Lanzhou 730050, China*
[3] *China Mobile Communications Group, Henan Co., Ltd., Zhengzhou 450000, China*

Correspondence should be addressed to Xiaoyun Chen; chenxy@lzu.edu.cn

Received 25 July 2014; Revised 27 December 2014; Accepted 8 January 2015

Academic Editor: Joaquim Joao Judice

We propose an optimized Support Vector Machine classifier, named PMSVM, in which System Normalization, PCA, and Multilevel Grid Search methods are comprehensively considered for data preprocessing and parameters optimization, respectively. The main goals of this study are to improve the classification efficiency and accuracy of SVM. Sensitivity, Specificity, Precision, and ROC curve, and so forth, are adopted to appraise the performances of PMSVM. Experimental results show that PMSVM has relatively better accuracy and remarkable higher efficiency compared with traditional SVM algorithms.

1. Introduction

The swift development of machine learning technologies gives us a good chance to process and analyse data in a brand-new perspective. Machine learning, also known as knowledge discovery, is one of the most important branches of computer science, which aims to find useful patterns from data and is quite different from those traditional statistical methods. As a comparatively new machine learning algorithm, Support Vector Machine (SVM) has attracted much attentions recently and has been successfully used in various application vocations [1–6]. In this study, we focus on constructing an optimized SVM model, so as to use it on heart disease data classification, aiming to improve the classification efficiency and accuracy of SVM.

Many literatures have involved contents of using Support Vector Machine to deal with data. Muthu Rama Krishnan et al. designed a SVM based classifier, which was used on two UCI mammogram datasets for breast cancer detection and reached the accuracy of 99.385% and 93.726%, respectively [7]. Xie and Wang integrated a hybrid feature selection method with SVM for erythematosquamous disease diagnosis, which reached the accuracy of 98.61% [8].

Feature selection is the basis of machine learning algorithms; appropriate feature selection strategy can obviously improve the performances of machine learning methods. Deisy et al. proposed a novel information theory based feature selection algorithm to improve the classification accuracy for SVM classifiers [9–12]. Other feature selection methods such as mutual information measurement [13], kernel F-score feature selection, and explicit margin-based feature elimination method are often adopted to get better classification results for SVM or other machine learning algorithms [14–17].

Most of the machine learning algorithms have their parameters; proper measures should be taken to decide the optimized values of them. Genetic Algorithm, Particle Swarm Optimization Algorithm, Artificial Immune System Algorithm and Grid Search Method are those most often used parameter optimization algorithms [18–23]. Generally, data feature selection methods and parameter optimization strategies are comprehensively considered. Lin et al. developed a Simulated Annealing approach for parameter determination and feature selection in SVM, and experiments showed the good performance of it [24]. Tan et al. proposed a new hybrid approach, in which Genetic Algorithm and Support

Vector Machine are integrated effectively based on a wrapper strategy [25], which performed well on the UCI chromosome dataset. Literature [26] presented a hybrid approach based on feature selection, fuzzy weighted preprocessing, and artificial immune recognition system, which was used on the UCI heart disease and hepatitis disease datasets, and the obtained accuracies are 92.39% and 81.82%, respectively.

Besides feature selection and parameter optimization, kernel function is another factor that should be considered for kernel based machine learning algorithms like SVM. Khemchandani et al. adopted an optimal kernel selection technique in Twin Support Vector Machines; the efficiency of it was testified with some UCI machine learning benchmark datasets [12, 27]. Abibullaev et al. introduced a linear programming SVM with multikernel function for brain signal data classification and got a good performance [15, 28–30].

Artificial Neural Network [23], Extreme Learning Machine [31], K-Nearest Neighbor analysis, Fuzzy Logic based methods [32–34], Ensemble Learning algorithms [34–39], and so forth, are often used or hybridly used to finish data classification tasks and usually can get good classification results. As a new version of Support Vector Machine, Least Square SVM involves equality constraints instead of inequality constraints and works with a least squares cost function. An obvious drawback of the Least Square SVM is that the sparseness is lost [22, 40, 41]. Yang et al. developed an adaptive pruning algorithm based on the bottom-to-top strategy, in which the incremental and decremental learning procedures were used and solved the drawback of traditional Least Square SVM [42–45].

Through the investigation of existing literatures, we noticed that the main points of studies using Support Vector Machine for classification are modifying and utilizing typical classification algorithms in combination and trying to acquire better classification performance. In general, the main process includes three procedures: (1) Data Preprocessing (feature selection, normalization, dimension reduction, etc.); (2) Constructing Optimized Classification Models (including parameter optimization); (3) Classification Accuracy and Efficiency Demonstration.

Although lots of efforts have been made on SVM and its applications, the performance of it is undesirable and still needs to be optimized.

The remaining parts of this paper are arranged as follows: The Mathematical Derivation of Support Vector Machine part shows the mathematical nature of SVM; the part Process of Principal Component Analysis gives the detail of Principal Component Analysis; the proposed System Normalization, Stratified Cross Validation, and Multilevel Grid Search based SVM algorithm are described in the Proposed Methods part; corresponding experiments are shown in the Experimental Results part; the following part gives the conclusions of this study.

2. Mathematical Derivation of Support Vector Machine

Support Vector Machine (SVM), as one of the most effective machine learning algorithms used for classification or regression problems, was firstly proposed by Vapnik and his Colleagues in 1995 and its history can be backtracked to the basic works of the Statistical Learning Theory since the 1960s [1–4]. SVM is good at processing nonlinear, high dimension, and little sample machine learning problems.

SVM is built on the basis of the VC Dimension (Vapnik Chervonenkis Dimension) Theory and the Structural Risk Minimum Theory, which are the core contents of the Statistical Learning Theory [2]. SVM has both solid theoretical foundation and ideal generation ability [6]. Presently, SVM has been used in many domains and occasions, such as handwriting recognition, biological character recognition (e.g., face recognition), credit card cheat checking, image segmentation, bioinformatics, function fitting, and medical data analysis [6].

As has been mentioned, SVM can be used to solve classification and regression problems, and SVM in these two different occasions are called SVC and SVR, respectively. In this paper, only SVC is involved, and it is uniformly referred to as "SVM." Intuitively (e.g., in 2-dimensional space), the classification problems can be divided into Linearly Separable tasks (corresponding data is made of linearly separable samples) and Linearly Inseparable tasks (data is formed with linearly inseparable samples, also called nonlinearly separable data, or nonlinear data, for short.). Figure 1 shows these two cases. Such situations can be extended into high dimensional space.

2.1. Linearly Separable Case. For the sake of simplicity, we only consider two-class classification situations.

Given a dataset D, $D = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in R^d$, $y_i \in \{-1, 1\}$. Essentially, dataset D is a set of binary group (x_i, y_i); here $x_i = \{x_{i1}, x_{i2}, \ldots, x_{id}\}$ stands for a data sample, and $y_i = \pm 1$ is the corresponding class label of x_i. Particularly, we use X to represent arbitrary data sample, and its corresponding class label is represented with Y. In such situations, SVM searches for the hyperplane with the largest separation "Margin" [1–4], that is, the Maximum Marginal Hyperplane (MMH). A separating hyperplane can be written as

$$W \cdot X + b = 0, \tag{1}$$

where $W = \{w_1, w_2, \ldots, w_d\}$ is weight vector; b is called as bias. Let $q(X) = W \cdot X + b$, the geometrical distance from a sample X to the optimal hyperplane can be expressed as

$$r = \frac{q(X)}{\|W\|}; \tag{2}$$

here, $q(X)$ is referred to as discriminant function. The purpose of SVM is to find the parameters W and b, so as to maximize the margin of separation (δ in (5)). Without loss of generality, the function margin can be fixed to be equal to 1. Thus, given a training set $\{(x_i, y_i)\}_{i=1}^{n} \in R^d \times \{\pm 1\}$, we get

$$\begin{aligned} W \cdot x_i &\geq +1, \quad \forall y_i = +1, \\ W \cdot x_i &\leq -1, \quad \forall y_i = -1. \end{aligned} \tag{3}$$

The particular samples $(x_i)_s$ satisfy the equalities of (3) which are referred to as Support Vectors; that is, they are exactly

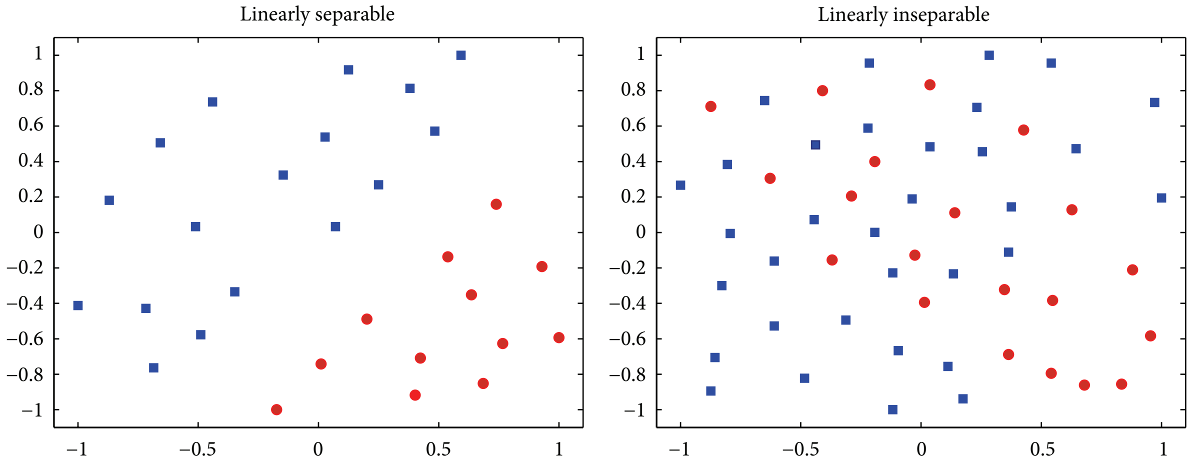


Figure 1: Linear separable data and nonlinear separable data.

the closest samples to the optimal separating hyperplane. Accordingly, the geometrical distance from the Support Vector x^* to the optimal separating hyperplane is

$$r^* = \frac{q(x^*)}{\|W\|} = \begin{cases} \frac{1}{\|W\|}, & \forall y^* = +1, \\ -\frac{1}{\|W\|}, & \forall y^* = -1. \end{cases} \tag{4}$$

Obviously, the margin of separation δ is

$$\delta = 2r^* = \frac{2}{\|W\|}. \tag{5}$$

To get the maximum margin hyperplane is to maximize δ with respect to W and b.

$$\begin{aligned} &\max_{W,b} \quad \frac{2}{\|W\|} \\ &\text{s.t.} \quad y_i(Wx_i + b) \geq 1, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{6}$$

Equivalently,

$$\begin{aligned} &\min_{W,b} \quad \frac{1}{2}\|W\|^2 \\ &\text{s.t.} \quad y_i(Wx_i + b) \geq 1, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{7}$$

The constrained optimization problem in (7) is known as the Primal Problem. Through the method of Lagrange Multiplier, we construct the following Lagrange Function:

$$L(W, b, \alpha) = \frac{1}{2}W^TW - \sum_{i=1}^{n}\alpha_i\left[y_i(Wx_i + b) - 1\right], \tag{8}$$

where α_i is the Lagrange Multiplier with respect to the i_{th} inequality. We can get the following conditions of optimality, through differentiating $L(W, b, \alpha)$ with respect to W and b and setting the results equal to 0:

$$\begin{aligned} \frac{\partial L(W, b, \alpha)}{\partial W} &= 0, \\ \frac{\partial L(W, b, \alpha)}{\partial b} &= 0; \end{aligned} \tag{9}$$

thus, we obtain

$$\begin{aligned} W &= \sum_{i=1}^{n}\alpha_i y_i x_i, \\ \sum_{i=1}^{n}\alpha_i y_i &= 0. \end{aligned} \tag{10}$$

The corresponding Dual Problem can be inferred by means of substituting (10) into (8):

$$\begin{aligned} &\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j \\ &\text{s.t.} \quad \sum_{i=1}^{n}\alpha_i y_i = 0 \\ &\qquad \alpha_i \geq 0, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{11}$$

The following equation (12) gives the Karush-Kuhn-Tucker (KKT) complementary condition:

$$\alpha_i\left[y_i(Wx_i + b) - 1\right] = 0, \quad i = 1, 2, \ldots, n. \tag{12}$$

As a result of it, only the support vectors (x_i, y_i) (the closest samples to the optimal separating hyperplane, which determine the maximal margin) correspond to the nonzero $\alpha_i s$ (all the other $\alpha_i s$ equal to zero); (11), which describes the Dual Problem, is a typical Convex Quadratic Programming Optimization Problem. In most cases, the Convex Quadratic Programming Optimization Problem function can efficiently converge to the global optimum, by means of adopting some appropriate optimization techniques. We can acquire the optimal weight vector W^* with (13) after determining the optimal Lagrange multipliers α_i^*:

$$W^* = \sum_{i=1}^{n}\alpha^* y_i x_i; \tag{13}$$

therefore, the corresponding optimal bias b^* can be expressed as follows:

$$b^* = 1 - W^* x_s, \quad \text{for } y_s = +1. \tag{14}$$

2.2. Linearly Inseparable Cases

2.2.1. Soft Margin SVM. Soft margin SVM aims to extend the Maximal Separating Margin SVM (so-called, hard margin SVM), so that the hyperplane allows a few of noisy data to exist. On this occasion, a variable ξ_i, named Slack factor, is introduced to account for the amount of a violation of classification of the classifier. Classification problem in such cases can be described as

$$\begin{aligned} &Wx_i + b \geq 1 - \xi_i, \quad \forall y_i = +1, \\ &Wx_i + b \leq -1 + \xi_i, \quad \forall y_i = -1; \end{aligned} \tag{15}$$

that is,

$$y_i (Wx_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, n. \tag{16}$$

The Primal Problem can be described as

$$\begin{aligned} \min_{W,b} \quad & \frac{1}{2} \|W\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i (Wx_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{17}$$

The corresponding Dual Problem Function (Dual Function) of the soft margin is formulated as

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{18}$$

KKT complementary condition in such inseparable case is

$$\alpha_i [y_i (Wx_i + b) - 1 + \xi_i] = 0, \quad i = 1, 2, \ldots, n, \tag{19}$$

$$\beta_i \xi_i = 0, \quad i = 1, 2, \ldots, n, \tag{20}$$

where β_i's are the Lagrange Multipliers corresponding to ξ_i that have been introduced to enforce the nonnegativity of ξ_i. At the point (saddle point), at which the derivative of the Lagrange function for the primal problem with respect to ξ_i is zero, the evaluation of the derivative yields

$$\alpha_i + \beta_i = C. \tag{21}$$

Simultaneously considering (20) and (21), we acquire

$$\xi_i = 0 \quad \text{if } \alpha_i < C. \tag{22}$$

Consequently, the optimal weight W^* can be expressed as follows:

$$W^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i. \tag{23}$$

The optimal bias b^* can be obtained by means of taking any sample (x_i, y_i) in the training set, for which we have $0 < \alpha_i^* < C$ and the corresponding $\xi_i = 0$, and using the sample in (19)

$$b^* = 1 - \xi_i - \sum_{sv} \alpha_i^* y_i x_i x_s, \quad \forall y_s = +1. \tag{24}$$

2.2.2. Kernel Trick Based SVM. For linearly inseparable cases, kernel trick is another commonly used technique. An appropriate kernel function, which is based on the inner product between the given samples, is to be defined as a nonlinear transformation of samples from the original space to a feature space with higher or even infinite dimension so as to make the problems linearly separable. That is, a complicated classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space. Actually, we can adopt a nonlinear mapping function

$$\Phi : X \longrightarrow \phi(X), \qquad R^d \longrightarrow F \tag{25}$$

to map data X in original (or primal) space into a higher (ever infinite) dimension space F, such that the mapped data in new feature space is more likely linearly separable. Thus we can extend the separating hyperplane function into the following form:

$$\begin{aligned} &W \cdot \phi(x_i) + b \geq +1 \quad \forall y_i = +1, \\ &W \cdot \phi(x_i) + b \leq -1 \quad \forall y_i = -1; \end{aligned} \tag{26}$$

that is,

$$y_i [W \cdot \phi(x_i)] + b \geq +1, \quad i = 1, 2, \ldots, n. \tag{27}$$

The separating hyperplane is

$$W \cdot \phi(X) + b = 0. \tag{28}$$

The primal problem in such case is

$$\begin{aligned} \min_{w,b} \quad & \frac{\|W\|^2}{2} \\ \text{s.t.} \quad & y_i [W \cdot \phi(x_i) + b] \geq +1, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{29}$$

Using the same mathematical trick in linearly separable SVM, we get the corresponding Dual Function

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^{n} \alpha_i \\ & \qquad - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & \alpha_i \geq 0, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{30}$$

The KKT complementary condition is

$$\alpha_i y_i \left[W\phi\left(x_i\right) + b\right] - 1 = 0, \quad i = 1, 2, \ldots, n. \tag{31}$$

After getting the optimal Lagrange Multipliers α_i^*, we acquire the optimal weight vector and the corresponding bias that are described with function (32) and function (33), respectively:

$$W^* = \sum_{i=1}^{n} \alpha_i^* y_i \phi\left(x_i\right), \tag{32}$$

$$b^* = 1 - \sum_{sv} \alpha_i^* y_i \phi\left(x_i\right) \phi\left(x_s\right), \quad \forall y_s = +1. \tag{33}$$

Fortunately, the inner product like the form of $\phi(x_i)\phi(x_j)$ can be instituted with $K(x_i, x_j)$, that is,

$$K\left(x_i, x_j\right) = \phi\left(x_i\right) \phi\left(x_j\right); \tag{34}$$

thus, we only need to calculate inner product in the original space not in the feature space, which reduces the calculation complexity obviously; meanwhile, we are refrained from searching for the proper nonlinear mapping function $\phi(X)$.

In fact, the kernel trick cannot always guarantee the mapped problems to be absolutely linearly separable, so soft margin SVM and kernel based SVM are integrated to exert the different advantages of them two and thus can solve the linearly inseparable problems more efficiently. The corresponding Dual form for the constrained optimization problem in the kernel soft margin SVM is listed as follows:

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K\left(x_i, x_j\right) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & 0 \le \alpha_i \le C, \quad i = 0, 1, \ldots, n. \end{aligned} \tag{35}$$

Accordingly, the optimal classifier is

$$f(x) = \sum_{i=1}^{n} \alpha_i^* y_i K\left(x_i, x\right) + b^*, \tag{36}$$

where

$$b^* = 1 - \sum_{i=1}^{n} \alpha_i^* y_i K\left(x_i, x_s\right), \quad \forall y_s = +1. \tag{37}$$

The most frequently used kernel functions are

(i) $K(x, y) = x \cdot y$;

(ii) $K(x, y) = (x \cdot y + c)^h$, $c \ge 0$;

(iii) $K(x, y) = \exp\{-\|x - y\|^2 / 2\sigma^2\}$;

(iv) $K(x, y) = \tanh(kx \cdot y - \delta)$, $k > 0$.

The third one is the RBF kernel function, which is used in our experiments.

3. Process of Principal Component Analysis

Principal Components Analysis (PCA), as was firstly introduced by Pearson in 1901, is a methodology that can be used to reduce the number of explicative variables of a dataset. In this paper, PCA is used for dimension reduction.

The main procedure of PCA is introduced as follows.

Given a dataset D', where $D' = \{x_1, x_2, \ldots, x_n\}$, $x_i \in R^d$, $i = 1, 2, \ldots, n$. Essentially, dataset D' is a $n \times d$ matrix. Firstly, D' should be normalized to get matrix with d normalized n dimensional column vectors, in case that attributes with large domains dominate with smaller ones (the following procedures will be exerted on these column vectors). Then, K orthonormal vectors will be calculated to act as a basis for the normalized input data, which are referred to as the principal components. Let $Y_1, Y_2, \ldots, Y_K$ denote these K mutually perpendicular unit vectors; they should satisfy the following requirements:

$$\operatorname{COV}\left(Y_i, Y_j\right) = 0, \quad \text{where } i, j \in \{1, 2, \ldots, K\}, \ i \ne j. \tag{38}$$

Thirdly, these principal components are sorted descendingly according to their "significance." Finally, those first several components will be chosen to reconstruct a good approximation of the original data. Thus the dimension size of the data is reduced; that is, PCA can be used to reduce data dimension.

4. The Proposed Methods

In this paper, we select the Stalog Heart Disease Dataset, which is from the UCI (University of California at Irvine) Machine Learning Repository, as our experimental data. The dataset contains 270 tuples; each tuple includes thirteen data attributes and one class attribute. The detail of the dataset is described with Table 1.

Because the value ranges of different attributes vary greatly, the tuples in Stalog Heart Disease Dataset are preprocessed as follows. Firstly, to facilitate the following experiments, we change the values of the attribute "class" into 1 and −1, respectively. That is, let 1 represent "presence," and −1 represent "absence." Then tuples in the dataset are normalized by column to [−1, 1]. In the normalized process, only the front thirteen data attributes are considered. We name such attribute data normalization process as System Normalization. The actual normalization procedure for each column vector C_j, $j = 1, 2, \ldots, d$, in Dataset D' is illustrated with (39).

Let $D_{(i,j)}$ represent the i_{th} element of C_j.

$$\begin{gathered} D_{(i,j)} = \frac{D_{(i,j)} - \left(\max\left(C_j\right) + \min\left(C_j\right)\right)/2}{\left(\max\left(C_j\right) - \min\left(C_j\right)\right)/2} \\ \forall i, j, \quad i = 1, 2, \ldots, n; \quad j = 1, 2, \ldots, d. \end{gathered} \tag{39}$$

To demonstrate the performance of SVM classifier, we often adopt cross validation method to get the accuracy of the class model. Traditionally, when using K-fold cross validation, the input dataset is randomly divided into K subsets, so that all the subsets have almost the same number

Table 1: Description of the UCI Stalog Heart Disease Dataset.

Number	Attributes	Full name of attributes	Value types
1	age	Age in years	Real
2	sex	Sex (1 = male; 0 = female)	Binary
3	cp	cp: chest pain type (1 = typical angina; 2 = atypical angina; 3 = nonanginal pain; 4 = asymptomatic)	Nominal
4	trestbps	Resting blood pressure (in mm Hg on admission to the hospital)	Real
5	chol	Serum cholesterol in mg/dL	Real
6	fbs	(Fasting blood sugar >120 mg/dL) (1 = true; 0 = false)	Binary
7	restecg	Resting electrocardiographic results (values 0, 1, 2)	Nominal
8	thalach	Maximum heart rate achieved	Real
9	exang	Exercise induced angina (1 = yes; 0 = no)	Binary
10	oldpeak	ST depression induced by exercise relative to rest	Real
11	slope	The slope of the peak exercise ST segment	Ordered
12	ca	Number of major vessels (0–3) colored by fluoroscopy	Real
13	thal	3 = normal; 6 = fixed defect; 7 = reversible defect	Nominal
14	num	Diagnosis of heart disease (angiographic disease status: 1 = absence; 2 = presence)	Binary

of samples. In the training phase of the classification model, each subset will act as testing subset only once and will act as training subset $K - 1$ times. In other words, when using K-fold classification validation, K classification models will be founded, each of them using $K - 1$ subsets as training set, and the remaining one subset as testing set. The final classification performance is to be appraised by using the average result of the K classification models.

In this paper, the stratified sample technology is adopted to generate the K folds (subsets); thus each subset will have the same number of positive samples and the same number of negative samples; the ratio between positive samples and negative samples of each subset is just the same as that of the whole dataset. That is, the acquired subsets will keep the statistical distribution of the original dataset. Algorithm 1 describes the main process of the Stratified Cross Validation.

4.1. Multilevel Grid Search. In this paper, we select C-SVC as our SVM classifier, and the RBF kernel as the kernel function. Thus we need to decide the values of the punishment parameter C of C-SVC and the parameter g of the RBF kernerl function. Generally, Grid Search technique is the most frequently used skill for deciding the optimal value of these two parameters. Given initial value ranges and search steps for the two parameters, respectively, Grid Search algorithm will iteratively check every value pair of C and g to try to obtain the optimal values of the value pair. Different settings of the initial value ranges and the steps for these two parameters will influence greatly the optimum results. Meanwhile, because the Grid Search method is of typical exhaustive search technology, the search process is time consuming.

Here, we present an adapted grid search method named Multilevel Grid Search (MGS), which effectively spares the search time and gets the same search result as the traditional grid search process.

Algorithm 2 reveals the details of our Multilevel Grid Search method.

Figures 2 and 3 show the parameters optimization results of the MGS method. Figure 2(a) gives the Grid Search results on S-DATA3; Figure 2(b) demonstrates the Grid Search results on PCA and System Normalization process based S-DATA3; Figures 2(c)-2(d) demonstrate the Multilevel Grid Search process on PCA and System Normalization process based S-DATA3; the situation of Figure 3 is about the same as Figure 2, but its experimental data is R-DATA6.

Figures 2(a) and 3(a) show that PCA has little influence on the classification accuracy of SVM, and averagely, experimental results in our study show that when adopting proper PCA threshold, PCA based SVM has similar classification accuracy as traditional SVM algorithms.

5. Experimental Results

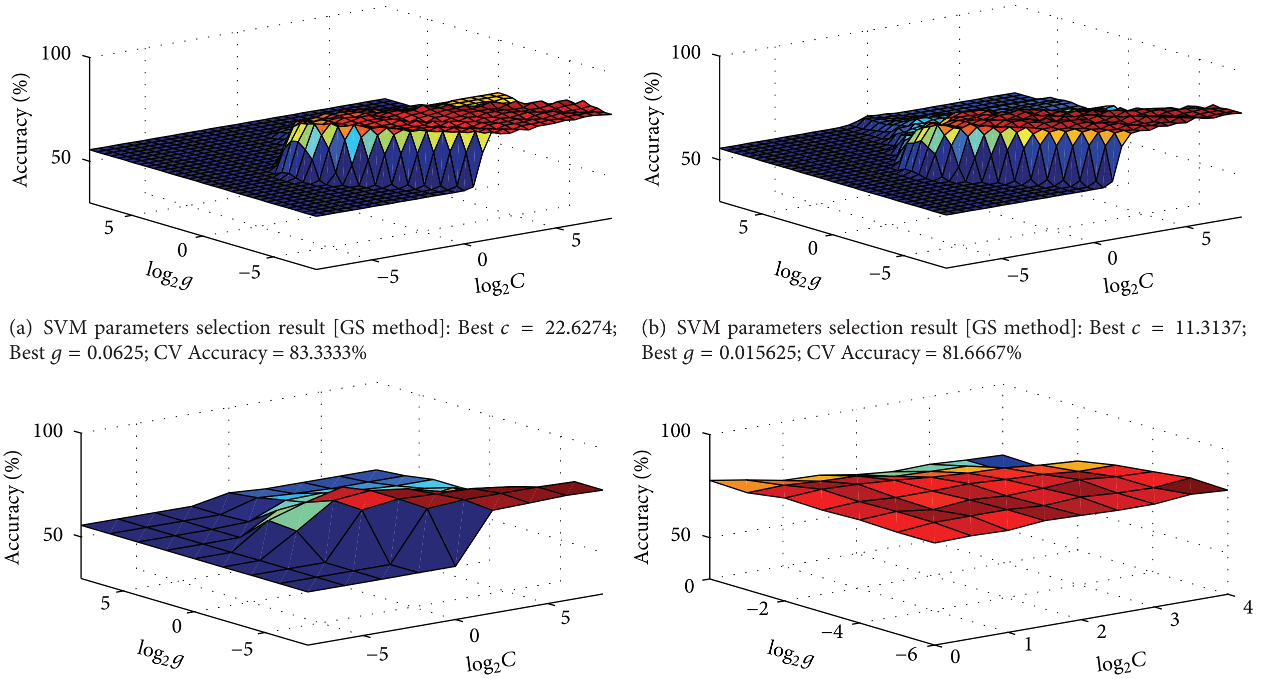
In this paper, the UCI Stalog heart-disease dataset is used to test our method. The main purpose of our study is to create a more efficient and available SVM model. The original dataset is processed based on the holdout method; that is, the given data are partitioned into two independent sets, one as training set, the other as test set. Two-thirds of the original data are allocated to the training set; the remaining one-third is allocated to the test data. Firstly, we repeat stratified subsampling based holdout method 10 times to generate 10 stratified datasets S-DATA1, S-DATA2, . . . , S-DATA10; then, random subsampling based holdout method is repeated 10 times to generate R-DATA1, R-DATA2, . . . , R-DATA10; in this paper, the experiments are mainly based on S-DATA1, S-DATA3, . . . , S-DATA9, and R-DATA2, R-DATA4, . . . , R-DATA10.

```
**Require:** $D, K$
**Ensure:** $D_1, D_2, \ldots, D_K, ACC_{average}$
Calculate the row size ($S_r$) of dataset $D$;
Count the number of positive samples ($N^+$) and negative samples ($N^-$), respectively;
$T_1 = \left\lceil \frac{N^+}{K} \right\rceil$;
$T_2 = \left\lceil \frac{N^-}{K} \right\rceil$;
$i = 1$;
$U = D$;
for ($i < K$) do
    $D_i = \phi$;
    Sample $T_1$ positive samples from $U$;
    Sample $T_2$ negative samples from $U$;
    Add those sampled samples into subset $D_i$;
    $U = U - D_i$;
    $i = i + 1$;
end for
$D_i = U$;
$i = 1$;
for ($i \leq K$) do
    using subset $D_i$ as testing set, the remaining $K - 1$ subsets as training set;
    using the training set to create classification model of SVM;
    using the testing set to generate the accuracy rate $ACC_i$;
    $i = i + 1$;
end for
$ACC_{average} = \frac{1}{k} \sum_{i=1}^{K} ACC_i$
return the average accuracy rate $ACC_{average}$;
```

Algorithm 1: Stratified Cross Validation.

```
Require: $K, X, threshold\_ACC$
Ensure: $Best_C, Best_g, ACC$
INPUT: $C_{\min}, C_{\max}, bigC_{step}$;
INPUT: $g_{\min}, g_{\max}, bigg_{step}$;
GridSearch($Train_{label}, Train_{data}, C_{\min}, C_{\max}, g_{\min}, g_{\max}, bigC_{step}, bigg_{step}, K$);
$C_{step} = 0.5, g_{step} = 0.5$
while $ACC \neq threshold\_ACC$ do
    Record the best ACC and the corresponding $C$, $g$;
    $C_{\text{mid}} = \frac{C_{\max} + C_{\min}}{2}$
    $g_{\text{mid}} = \frac{g_{\max} + g_{\min}}{2}$
    if $C \geq C_{\text{mid}}$ then
        $C_{\min} = C_{\text{mid}}$
    else
        $C_{\max} = C_{\text{mid}}$
    end if
    if $g \geq g_{\text{mid}}$ then
        $g_{\min} = g_{\text{mid}}$
    else
        $g_{\max} = g_{\text{mid}}$
    end if
    GridSearch($Train_{label}, Train_{data}, C_{\min}, C_{\max}, g_{\min}, g_{\max}, C_{step}, g_{step}, K$);
end while
return $Best_C, Best_g$.
```

Algorithm 2: Multilevel Grid Search (MGS).



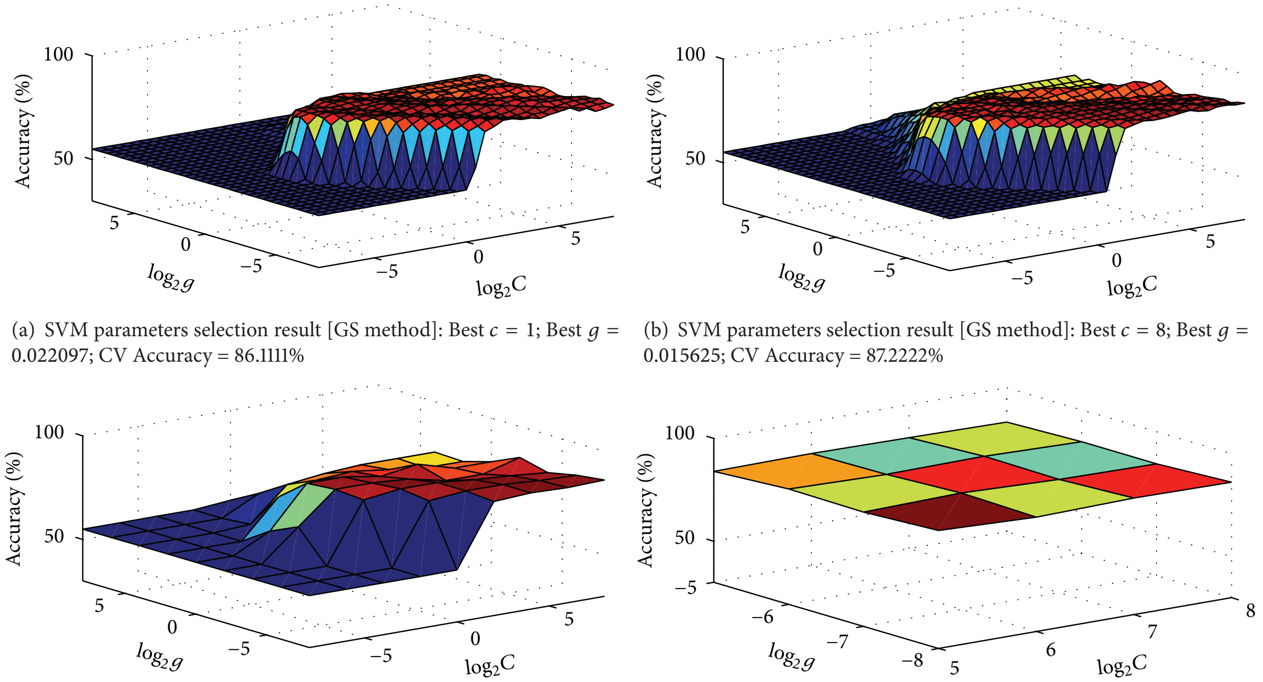
(a) SVM parameters selection result [GS method]: Best c = 22.6274; Best g = 0.0625; CV Accuracy = 83.3333%

(b) SVM parameters selection result [GS method]: Best c = 11.3137; Best g = 0.015625; CV Accuracy = 81.6667%

(c) SVM parameters selection result [MGS method]: Best c = 4; Best g = 0.015625; CV Accuracy = 80.5556%

(d) SVM parameters selection result [MGS method]: Best c = 11.3137; Best g = 0.015625; CV Accuracy = 81.6667%

FIGURE 2: MGS on S-DATA3.



(a) SVM parameters selection result [GS method]: Best c = 1; Best g = 0.022097; CV Accuracy = 86.1111%

(b) SVM parameters selection result [GS method]: Best c = 8; Best g = 0.015625; CV Accuracy = 87.2222%

(c) SVM parameters selection result [MGS method] Best c = 1; Best g = 0.0625; CV Accuracy = 86.1111%

(d) SVM parameters selection result [MGS method] Best c = 32; Best g = 0.0039063; CV Accuracy = 87.2222%

FIGURE 3: MGS on R-DATA6.

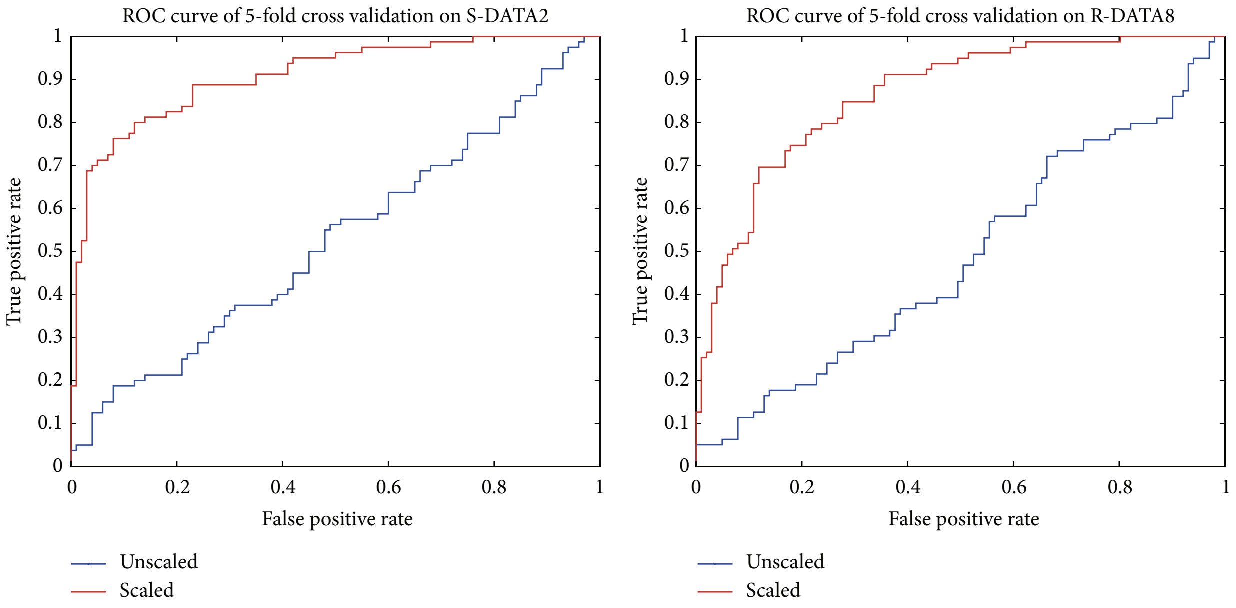


Figure 4: Roc curve for scaled and unscaled data.

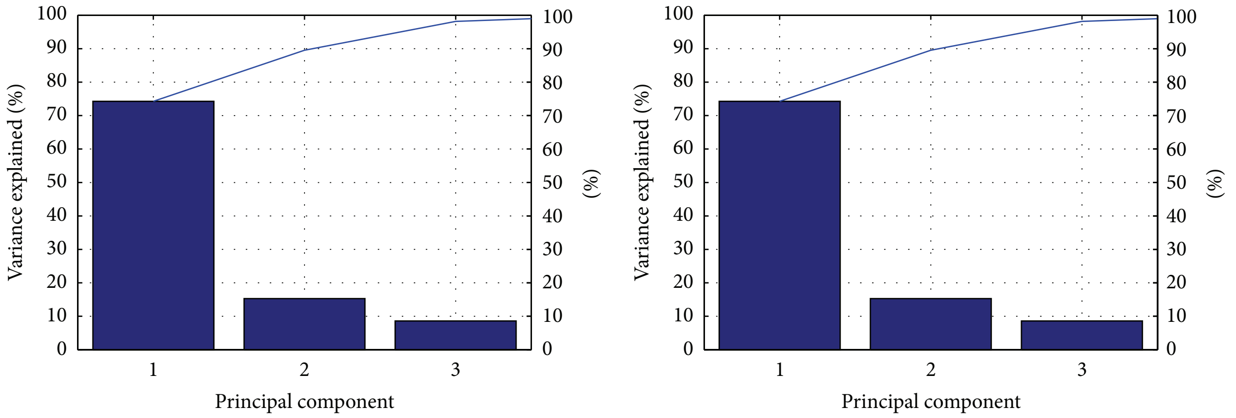


Figure 5: PCA results on two different datasets.

To study the necessity of our System Normalization process, lots of experiments are done on our datasets, and the results show that System Normalization exerts a great influence on the classification result of SVM; that is, it can effectively advance the classification accuracy of SVM. For simplicity, only part of the results are demonstrated. Figure 4 shows the influence of normalization on the classification result using traditional SVM classifier.

SVM is a promising machine learning algorithm, which has solid theory foundation and good generalization ability. But the training time of even the fastest SVMs can be extremely slow. In this paper, PCA is adopted to reduce the dimension of our data. Here, the threshold of PCA is 85%.

Figure 5 shows the PCA results on the Stalog Herat Disease Dataset. As shown in Figure 5, the System Normalization process has no obvious effect on PCA. Further study shows that, when adopting proper threshold (in this study, it is 85%), PCA has no direct influence on the classification accuracy of SVM. Figure 6 shows the ROC curve of SVM classifier on two different sampled Heart Disease Datasets: one is random sampled and the other is stratified sampled. It is obvious that PCA based SVM has similar classification accuracy as traditional SVM, but it can reduce the time complexity in some extent (about a 25% reduction).

We name the classifier based on PCA and MGS methods as PMSVM, and we call the classifier based on PCA strategy as PLSVM. All experimental results are compared with that of the famous LIBSVM algorithm. To demonstrate the performances of PMSVM, PLSVM with that of LIBSVM, Confusion Matrix, Sensitivity, Specialty, Precision, ROC curve, and AUC are used as the main evaluative criteria for classification accuracy; meanwhile, the time overhead is

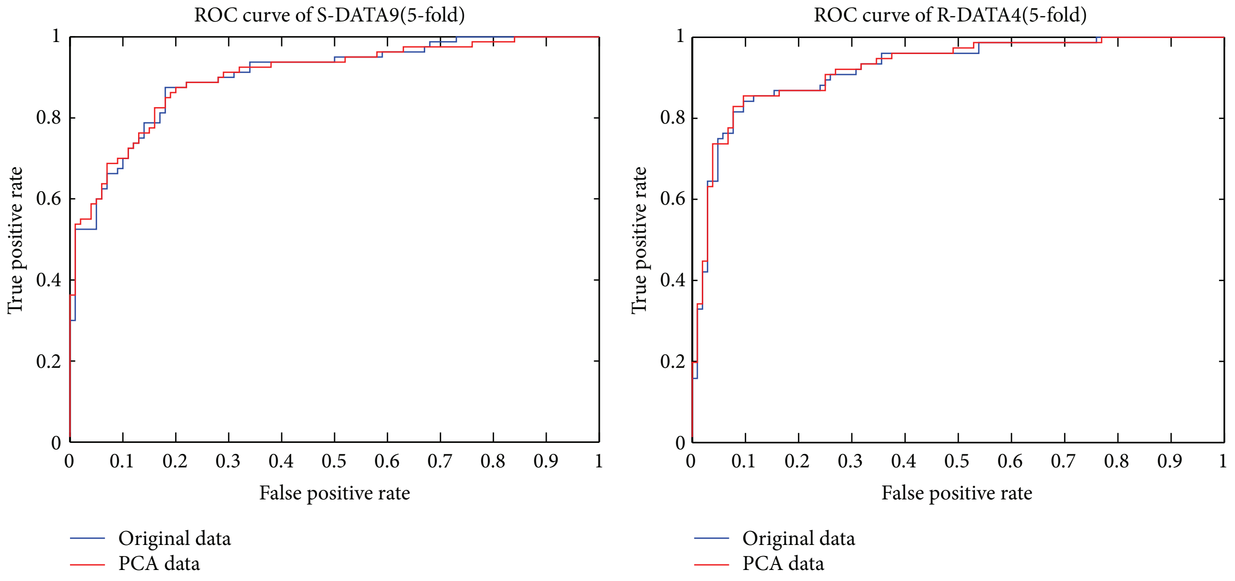


Figure 6: ROC curve on original data and PCA processed data (5-fold CV).

checked for measuring the efficiency of our method. All SVM algorithms are based on C-SVC classification model and RBF kernel function.

Confusion Matrix is a useful tool to a classifier for describing the classification results for different classes. Given Q classes, the Confusion Matrix is a $Q\times Q$ matrix, the element $CM_{i,j}$ describes the number of samples that a classifier classifies class i into class j. For an ideal classifier that has 100% classification accuracy, all the samples should be described with the elements on the main diagonal of the Confusion Matrix. That is, let $\|D\|$ represent the number of total samples of dataset D; we can get the following:

$$\sum_{i=1}^{Q} CM_{i,j} = \|D\|, \quad i = j. \tag{40}$$

Table 2 gives an example for a 2-class classifiers, where, $\text{True}_{\text{Positive}}$ represents the number of positive samples that are correctly classified as positive samples; $\text{Fause}_{\text{Negative}}$ represents the number of positive samples that are wrongly classified as negative samples; $\text{True}_{\text{Negative}}$ represents the number of negative samples that are correctly classified as negative samples; $\text{False}_{\text{Positive}}$ represents the number of negative samples that are wrongly classified as positive samples.

The following results of Sensitivity, Specialty, Precision, and so forth, closely rely on the contents of Confusion Matrix. Let Pos represent the total number of positive samples, and let Neg represent the total number of negative samples; it is easy to get the following relations, which are described as

$$\text{Pos} = \text{True}_{\text{Positive}} + \text{False}_{\text{Negative}},$$
$$\text{Neg} = \text{True}_{\text{Negative}} + \text{Fase}_{\text{Positive}}. \tag{41}$$

Table 2: Confision matrix for 2-class classifiers.

Class	Positive	Negative
Positive	$\text{True}_{\text{Positive}}$	$\text{Fause}_{\text{Negative}}$
Negative	$\text{False}_{\text{Positive}}$	$\text{True}_{\text{Negative}}$

"Sensitivity" is the ratio between $\text{True}_{\text{Positive}}$ and Pos. That is, it is the ratio of truly classified positive samples. Sensitivity can be described as

$$\text{Sensitivity} = \frac{\text{True}_{\text{Positive}}}{\text{Pos}}. \tag{42}$$

"Specificity" is the ratio between $\text{True}_{\text{Negative}}$ and Neg. In another word, it is the ratio of truly sampled negative samples. Specificity can be defined as

$$\text{Specificity} = \frac{\text{True}_{\text{Negative}}}{\text{Neg}}. \tag{43}$$

Precision gives the ratio between the number of truly classified positive samples and the total number of samples that are classified as true samples. Precision can be described as

$$\text{Precision} = \frac{\text{True}_{\text{Positive}}}{\text{True}_{\text{Positive}} + \text{False}_{\text{Positive}}}. \tag{44}$$

ROC curve is a useful visual tool for comparing the performances of different classification models. ROC is the abbreviation of Receiver Operating Characteristic. ROC curve displays the comparative performance evaluation between the true positive rate and the false positive rate of a given classifier. Here, AUC means the area under the curve (ROC), and it can express the accuracy of a given classification model. Figure 7 demonstrates the ROC curve of

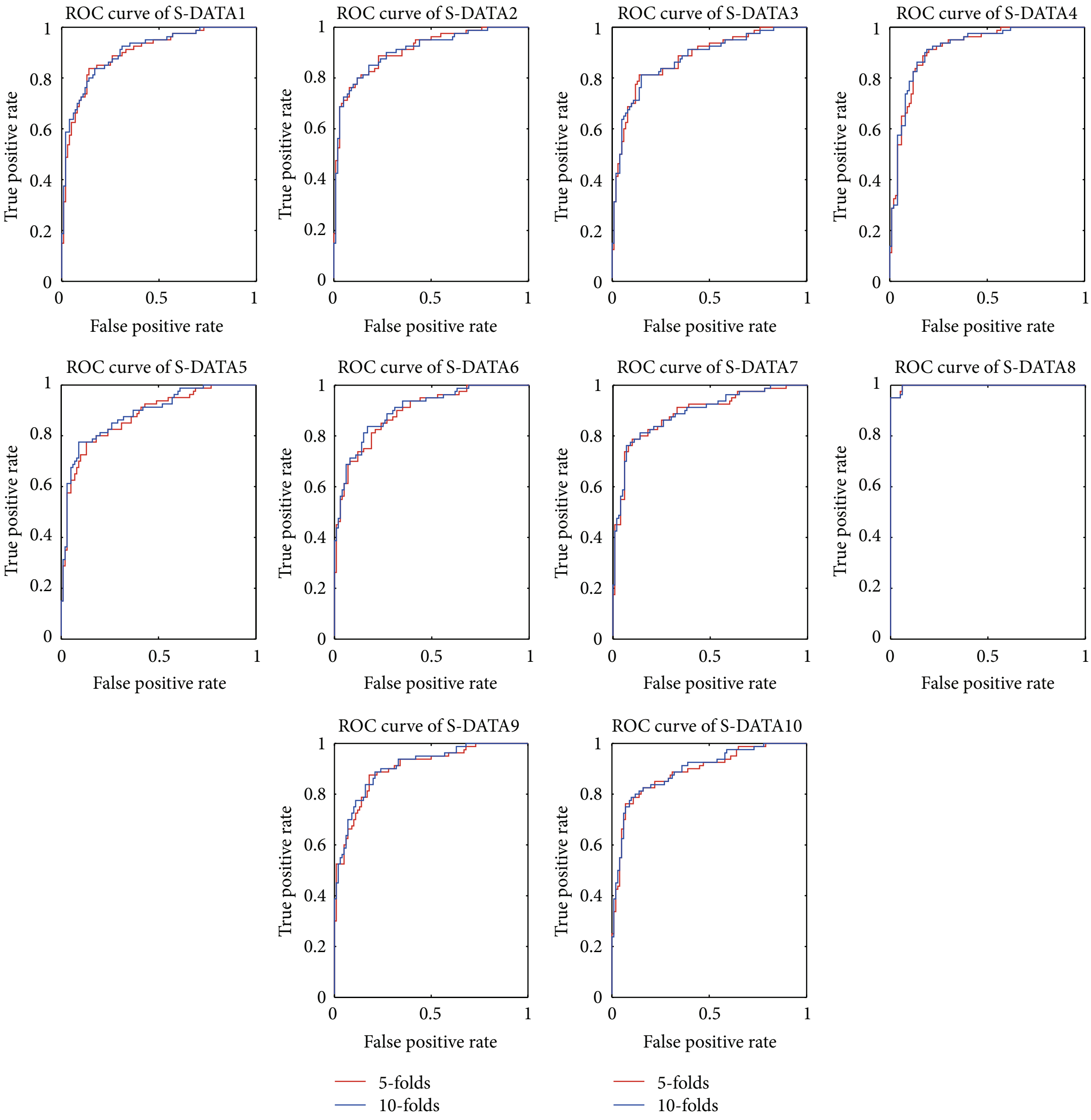


Figure 7: 5-fold and 10-fold ROC curve on 10 stratified sampled datasets.

of standard LIBSVM on 10 stratified sampled datasets using 5-fold cross validation and 10-fold cross validation methods, respectively. As we can see from the classification results, shown with Figure 7, that there are no obvious differences between the results of 5-fold cross validation method and that of 10-fold cross validation method. So, we will adopt the 5-fold cross validation technology as our default cross validation way in the following experiments, and the cross validation technology we used is referred as Stratified Cross Validation (SCV), as is mentioned in Algorithm 1.

Figure 8 demonstrates the ROC curve of PMSVM, PLSVM, and LIBSVM. Here, both parameters, C and g, of PLSVM and LIBSVM are set to the constant experience value 0.1. As is shown in Figure 8, the classification results of PLSVM and LIBSVM are quite similar; the difference between them is that PLSVM can effectively progress the classification efficiency by 25% or so, because of the PCA process adopted. Generally, the classification performances of the PMSVM, which is based on our PCA and MGS strategies, have obvious advantages; the classification accuracy is more than that of PLSVM and LIBSVM; meanwhile, the running time of PMSVM is significantly shorter than that of them; that is, the time consuming of our PMSVM has about a reduction of 75% or so than that of traditional LIBSVM.

Tables 3 and 4 show the overall situations between PMSVM and LIBSVM (here, parameters C and g of PMSVM

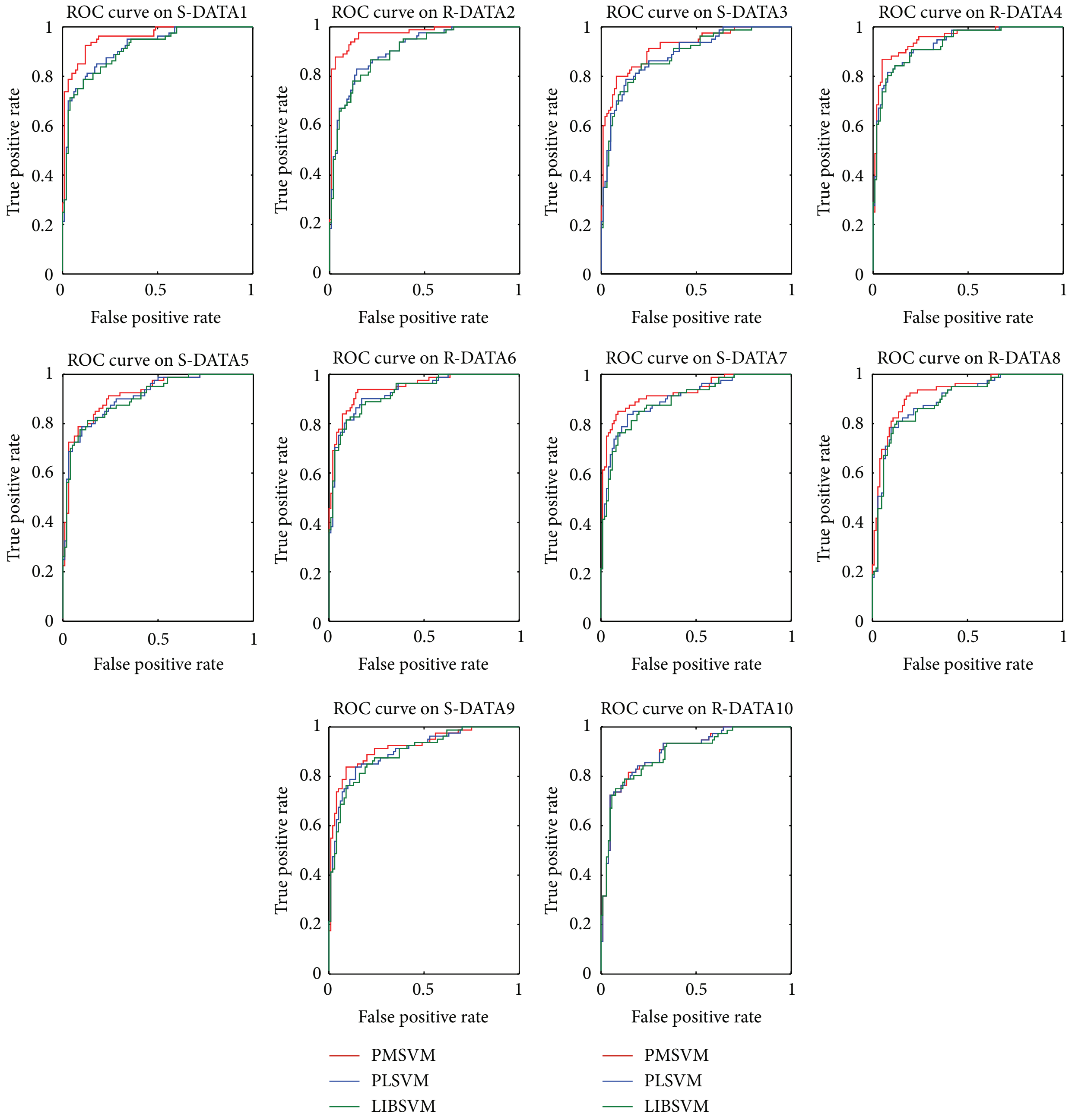


Figure 8: ROC curve of LIBSVM, PLSVM, and PMSVM.

and LIBSVM are acquired through Multilevel Grid Search and Grid Search, resp.). Figure 9 demonstrates the time overhead of PMSVM, PLSVM, and LIBSVM.

In the whole experiment, Stratified Sampling and Random Sampling methods show no direct influences on classification results of PMSVM, PLSVM, and LIBSVM, but Stratified Cross Validation provides more reliable results. And the influence of Stratified Sample technology on the classification results of imbalanced data is what we will study further in our future work.

6. Conclusions

In this paper, an optimized Support Vector Machine classifier, named PMSVM, is proposed, in which System Normalization, PCA, and MGS methods are used to try to average up the performances of SVM. Experimental results show that System Normalization can effectively assure the classification accuracy (Figure 4). PCA (Principal Component Analysis) can play a role of dimensionality reduction. As shown in our experiments, chosing proper threshold value, PCA can both economize classification time and assure classification

Table 3: Comparison of the accuracy and efficiency of algorithms on stratified sampled data.

Dataset	Algorithm	Parameters		Time (S)	Accuracy				
		C	g		Sensitivity	Specificity	Precision	AUC	CVA (%)
S-DATA1	LIBSVM	90.5097	0.011049	19.740543	0.813	0.890	0.856	0.94463	84.4444
	PMSVM	2.82840	0.176780	4.6617020	0.838	0.910	0.882	0.97575	84.4444
S-DATA3	LIBSVM	22.6274	0.062500	20.844411	0.813	0.910	0.878	0.93363	83.8889
	PMSVM	11.3137	0.015625	4.9883590	0.763	0.870	0.824	0.91788	81.6667
S-DATA5	LIBSVM	0.35355	0.125000	20.070109	0.788	0.880	0.840	0.91075	84.4444
	PMSVM	0.50000	0.125000	4.8111860	0.788	0.880	0.840	0.91910	84.4444
S-DATA7	LIBSVM	1.41420	0.353550	20.134157	0.838	0.920	0.893	0.91213	85.0000
	PMSVM	8.00000	0.044194	4.7952370	0.880	0.910	0.877	0.92675	85.0000
S-DATA9	LIBSVM	0.17678	0.083880	19.628079	0.800	0.880	0.842	0.92212	85.5556
	PMSVM	2.82840	0.250000	4.8249560	0.863	0.920	0.896	0.94350	86.6667

Table 4: Comparison of the accuracy and efficiency of algorithms on random sampled datasets.

Dataset	Algorithm	Parameters		Time (S)	Accuracy				
		C	g		Sensitivity	Specificity	Precision	AUC	CVA (%)
R-DATA2	LIBSVM	0.50000	0.1767800	19.610260	0.805	0.857	0.825	0.90640	82.2222
	PMSVM	256.000	0.0039063	5.5637770	0.780	0.857	0.825	0.96925	83.8889
R-DATA4	LIBSVM	1.00000	0.0888388	19.566909	0.842	0.913	0.877	0.94345	87.7778
	PMSVM	2.82840	0.0883880	4.1918490	0.842	0.923	0.889	0.95078	87.7778
R-DATA6	LIBSVM	1.00000	0.0220970	19.573738	0.802	0.909	0.878	0.93628	86.1111
	PMSVM	8.00000	0.0156250	4.5530740	0.802	0.929	0.903	0.94762	87.2222
R-DATA8	LIBSVM	0.17678	1.0000000	19.871964	0.785	0.911	0.873	0.89034	83.8889
	PMSVM	4.00000	0.0312500	5.1134920	0.797	0.861	0.818	0.91580	83.3333
R-DATA10	LIBSVM	0.12500	0.1767800	19.638424	0.750	0.885	0.826	0.89020	84.4444
	PMSVM	0.25000	0.0883880	4.8284440	0.763	0.885	0.829	0.89890	83.8889

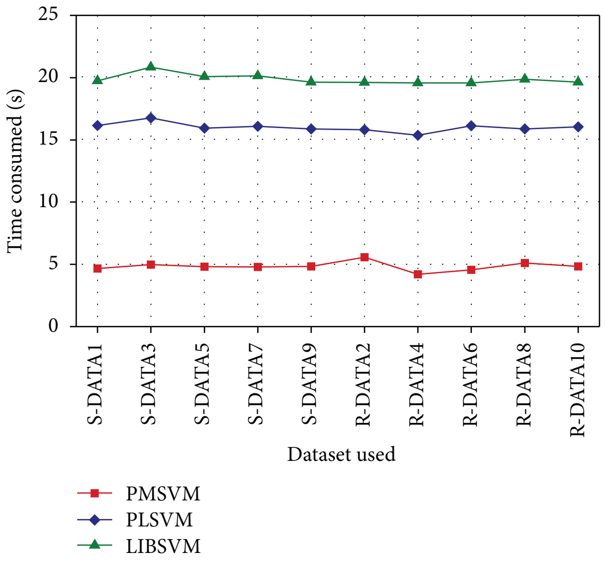


Figure 9: Comparison of time consuming.

accuracy. The most prominent character in our study is whenever adopting our PCA and Multilevel Grid Search (MGS) methods, the time consuming of our PMSVM algorithm can be reduced by 75% or so, averagely, and meanwhile can get similar or better classification accuracy than the classical LIBSVM algorithm.

Conflict of Interests

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

Acknowledgments

The authors would like to thank Professor Xiaoyun Chen for her scrupulous guiding and Dr. C.-J. Lin for supplying the LIBSVM software that is used in this research. This study is supported by the Fundamental Research Funds for the Central Universities (lzujbky-2013-229, lzujbky-2014-47).

References

[1] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[2] V. N. Vapnik, *The Nature of Statistical Learningtheory*, Springer, New York, NY, USA, 1995.

[3] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[4] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[5] H.-Y. Kao, T.-K. Chang, and Y.-C. Chang, "Classification of hospital web security efficiency using data envelopment analysis and support vector machine," *Mathematical Problems in Engineering*, vol. 2013, Article ID 542314, 8 pages, 2013.

[6] X. Wu and V. Kumar, *The Top Ten Algorithms in Data Mining*, Chapman & Hall, CRC Press, 2009.

[7] M. Muthu Rama Krishnan, S. Banerjee, C. Chakraborty, C. Chakraborty, and A. K. Ray, "Statistical analysis of mammographic features and its classification using support vector machine," *Expert Systems with Applications*, vol. 37, no. 1, pp. 470–478, 2010.

[8] J. Y. Xie and C. X. Wang, "Using support vector machines with a novel hybrid feature selection method for diagnosis of erythemato-squamous diseases," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5809–5815, 2011.

[9] C. Deisy, S. Baskar, N. Ramraj, J. S. Koori, and P. Jeevanandam, "A novel information theoretic-interact algorithm (IT-IN) for feature selection using three machine learning algorithms," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7589–7597, 2010.

[10] D. Çalişir and E. Dogantekin, "A new intelligent hepatitis diagnosis system: PCA-LSSVM," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10705–10708, 2011.

[11] E. Avci, "A new expert system for diagnosis of lung cancer: GDA-LS_SVM," *Journal of Medical Systems*, vol. 36, no. 3, pp. 2005–2009, 2012.

[12] R. Burbidge, M. Trotter, B. Buxton, and S. Holden, "Drug design by machine learning: support vector machines for pharmaceutical data analysis," *Computers & Chemistry*, vol. 26, no. 1, pp. 5–14, 2001.

[13] L. T. Vinh, S. Lee, Y. T. Park, and B. J. D'Auriol, "A novel feature selection method based on normalized mutual information," *Applied Intelligence*, vol. 37, no. 1, pp. 100–120, 2012.

[14] T. Deselaers, G. Heigold, and H. Ney, "Object classification by fusing SVMs and Gaussian mixtures," *Pattern Recognition*, vol. 43, no. 7, pp. 2476–2484, 2010.

[15] R. T. Peres and C. E. Pedreira, "A new local-global approach for classification," *Neural Networks*, vol. 23, no. 7, pp. 887–891, 2010.

[16] K. Polat and S. Güneş, "A new feature selection method on classification of medical datasets: kernel F-score feature selection," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10367–10373, 2009.

[17] Y. Aksu, D. J. Miller, G. Kesidis, and Q. X. Yang, "Margin-maximizing feature elimination methods for linear and nonlinear kernel-based discriminant functions," *IEEE Transactions on Neural Networks*, vol. 21, no. 5, pp. 701–717, 2010.

[18] L.-F. Chen, C.-T. Su, K.-H. Chen, and P.-C. Wang, "Particle swarm optimization for feature selection with application in obstructive sleep apnea diagnosis," *Neural Computing and Applications*, vol. 21, no. 8, pp. 2087–2096, 2012.

[19] J. S. Sartakhti, M. H. Zangooei, and K. Mozafari, "Hepatitis disease diagnosis using a novel hybrid method based on support vector machine and simulated annealing (SVM-SA)," *Computer Methods and Programs in Biomedicine*, vol. 108, no. 2, pp. 570–579, 2012.

[20] H. Kahramanli and N. Allahverdi, "Extracting rules for classification problems: AIS based approach," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10494–10502, 2009.

[21] S. Özşen and S. Güneş, "Effect of feature-type in selecting distance measure for an artificial immune system as a pattern recognizer," *Digital Signal Processing*, vol. 18, no. 4, pp. 635–645, 2008.

[22] H. Kahramanli and N. Allahverdi, "Design of a hybrid system for the diabetes and heart diseases," *Expert Systems with Applications*, vol. 35, no. 1-2, pp. 82–89, 2008.

[23] N. García-Pedrajas and D. Ortiz-Boyer, "A cooperative constructive method for neural networks for pattern recognition," *Pattern Recognition*, vol. 40, no. 1, pp. 80–98, 2007.

[24] S.-W. Lin, Z.-J. Lee, S.-C. Chen, and T.-Y. Tseng, "Parameter determination of support vector machine and feature selection using simulated annealing approach," *Applied Soft Computing Journal*, vol. 8, no. 4, pp. 1505–1512, 2008.

[25] K. C. Tan, E. J. Teoh, Q. Yu, and K. C. Goh, "A hybrid evolutionary algorithm for attribute selection in data mining," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8616–8630, 2009.

[26] K. Polat and S. Güneş, "A hybrid approach to medical decision support systems: combining feature selection, fuzzy weighted pre-processing and AIRS," *Computer Methods and Programs in Biomedicine*, vol. 88, no. 2, pp. 164–174, 2007.

[27] R. Khemchandani, Jayadeva, and S. Chandra, "Optimal kernel selection in twin support vector machines," *Optimization Letters*, vol. 3, no. 1, pp. 77–88, 2009.

[28] H.-L. Chen, D.-Y. Liu, B. Yang, J. Liu, and G. Wang, "A new hybrid method based on local fisher discriminant analysis and support vector machines for hepatitis disease diagnosis," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11796–11803, 2011.

[29] B. Liu, Z. Hao, and E. C. C. Tsang, "Nesting one-against-one algorithm based on SVMs for pattern classification," *IEEE Transactions on Neural Networks*, vol. 19, no. 12, pp. 2044–2052, 2008.

[30] B. Abibullaev, J. An, S.-H. Jin, S. H. Lee, and J. I. Moon, "Minimizing inter-subject variability in fNIRS-based brain-computer interfaces via multiple-kernel support vector learning," *Medical Engineering & Physics*, vol. 35, no. 12, pp. 1811–1818, 2013.

[31] L.-N. Li, J.-H. Ouyang, H.-L. Chen, and D.-Y. Liu, "A computer aided diagnosis system for thyroid disease using extreme learning machine," *Journal of Medical Systems*, vol. 36, no. 5, pp. 3327–3337, 2012.

[32] D. S. Yeung, W. W. Y. Ng, D. Wang, E. C. C. Tsang, and X. Z. Wang, "Localized generalization error model and its application to architecture selection for radial basis function neural network," *IEEE Transactions on Neural Networks*, vol. 18, no. 5, pp. 1294–1305, 2007.

[33] Q. L. Ye, C. X. Zhao, N. Ye, and Y. N. Chen, "Iterative support vector machine with guaranteed accuracy and run time," *Expert Systems*, vol. 27, no. 5, pp. 338–348, 2010.

[34] J. Hamidzadeh, R. Monsefi, and H. Sadoghi Yazdi, "DDC: distance-based decision classifier," *Neural Computing and Applications*, vol. 21, no. 7, pp. 1697–1707, 2012.

[35] C.-M. Chen, Y.-F. Lu, and C.-M. Hong, "Minimal structure of self-organizing HCMAC neural network classifier," *Neural Processing Letters*, vol. 23, no. 2, pp. 201–228, 2006.

[36] K. Polat and S. Güneş, "An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease," *Digital Signal Processing*, vol. 17, no. 4, pp. 702–710, 2007.

[37] S.-J. Wang, A. Mathew, Y. Chen, L.-F. Xi, L. Ma, and J. Lee, "Empirical analysis of support vector machine ensemble

classifiers," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6466–6476, 2009.

[38] K. Polat and S. Gunes, "Detection of ECG Arrhythmia using a differential expert system approach based on principal component analysis and least square support vector machine," *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 898–906, 2007.

[39] C.-T. Lin, C.-M. Yeh, S.-F. Liang, J.-F. Chung, and N. Kumar, "Support-vector-based fuzzy neural network for pattern classification," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 1, pp. 31–41, 2006.

[40] T. van Gestel, J. A. K. Suykens, B. Baesens et al., "Benchmarking least squares support vector machineclassifiers," *Machine Learning*, vol. 54, no. 1, pp. 5–32, 2004.

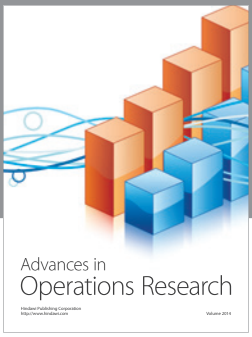
[41] K. Polat, S. Şahan, H. Kodaz, and S. Güneş, "Breast cancer and liver disorders classification using artificial immune recognition system (AIRS) with performance evaluation by fuzzy resource allocation mechanism," *Expert Systems with Applications*, vol. 32, no. 1, pp. 172–183, 2007.

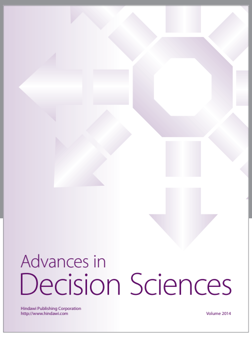
[42] K. Polat, S. Şahan, and S. Güneş, "Automatic detection of heart disease using an artificial immune recognition system (AIRS) with fuzzy resource allocation mechanism and k-nn (nearest neighbour) based weighting preprocessing," *Expert Systems with Applications*, vol. 32, no. 2, pp. 625–631, 2007.

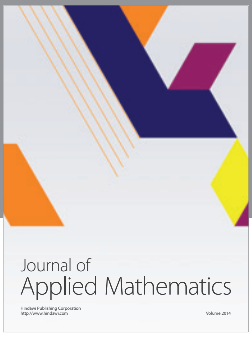
[43] M. J. Abdi and D. Giveki, "Automatic detection of erythemato-squamous diseases using PSO-SVM based on association rules," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 603–608, 2013.

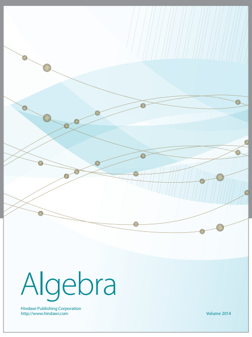
[44] K. Polat and S. Güneş, "Automated identification of diseases related to lymph system from lymphography data using artificial immune recognition system with fuzzy resource allocation mechanism (fuzzy-AIRS)," *Biomedical Signal Processing and Control*, vol. 1, no. 4, pp. 253–260, 2006.

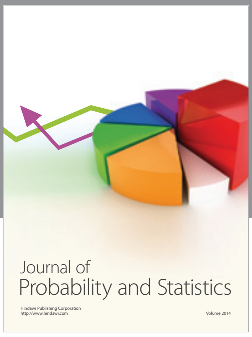
[45] X. Yang, J. Lu, and G. Zhang, "Adaptive pruning algorithm for least squares support vector machine classifier," *Soft Computing*, vol. 14, no. 7, pp. 667–680, 2010.


Advances in
Operations Research

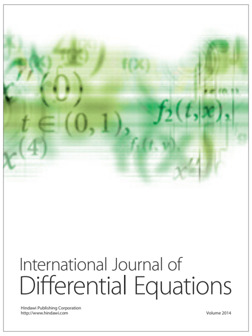

Advances in
Decision Sciences


Journal of
Applied Mathematics

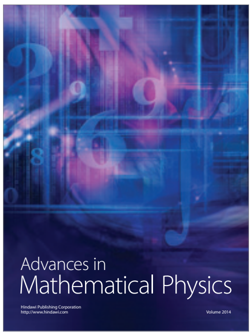

Algebra

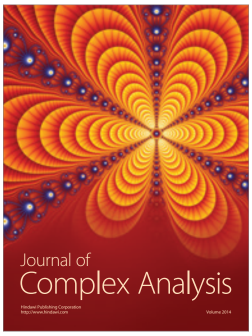

Journal of
Probability and Statistics

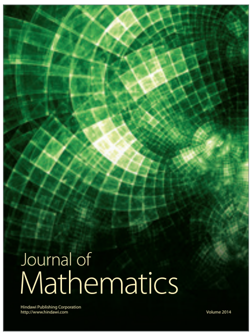
The Scientific
World Journal

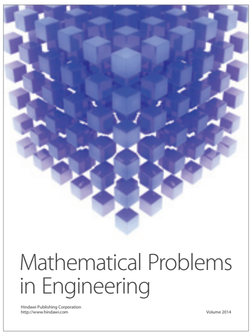

International Journal of
Differential Equations

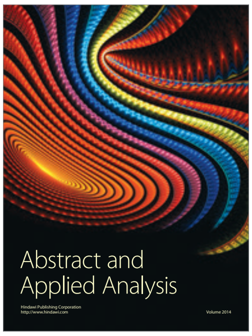
International Journal of
Combinatorics

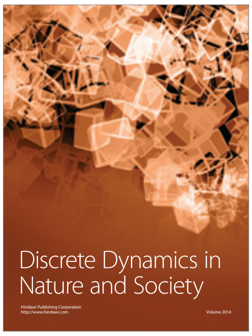

Advances in
Mathematical Physics

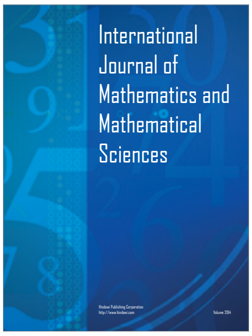

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

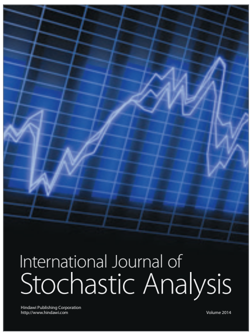

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization