Hindawi Publishing Corporation
Journal of Optimization
Volume 2016, Article ID 6305043, 8 pages
http://dx.doi.org/10.1155/2016/6305043

Research Article

An Optimal SVM with Feature Selection Using Multiobjective PSO

Iman Behravan,[1] Oveis Dehghantanha,[1] Seyed Hamid Zahiri,[2] and Nasser Mehrshad[2]

[1] *Department of Electrical Engineering, University of Birjand, No. 21, Sadaf 1.1 Street, Naranj 2 Alley, Shahid Avini Boulevard, Ghaffari Avenue, Birjand, South Khorasan 97176-33533, Iran*
[2] *Department of Electrical Engineering, Faculty of Engineering, University of Birjand, Birjand, Iran*

Correspondence should be addressed to Iman Behravan; i.behravan@gmail.com

Received 5 December 2015; Revised 22 May 2016; Accepted 7 June 2016

Academic Editor: Joe Imae

Support vector machine is a classifier, based on the structured risk minimization principle. The performance of the SVM depends on different parameters such as penalty factor, C, and the kernel factor, σ. Also choosing an appropriate kernel function can improve the recognition score and lower the amount of computation. Furthermore, selecting the useful features among several features in dataset not only increases the performance of the SVM, but also reduces the computational time and complexity. So this is an optimization problem which can be solved by heuristic algorithm. In some cases besides the recognition score, the reliability of the classifier's output is important. So in such cases a multiobjective optimization algorithm is needed. In this paper we have got the MOPSO algorithm to optimize the parameters of the SVM, choose appropriate kernel function, and select the best feature subset simultaneously in order to optimize the recognition score and the reliability of the SVM concurrently. Nine different datasets, from UCI machine learning repository, are used to evaluate the power and the effectiveness of the proposed method (MOPSO-SVM). The results of the proposed method are compared to those which are achieved by single SVM, RBF, and MLP neural networks.

1. Introduction

A pattern recognition system consists of different parts. One of the most important parts of such a system is classifying, which is done by different classifiers at the end of the process. Obviously, having a powerful classifier with high accuracy is critical in a pattern recognition system, since the output accuracy of the system is highly affected by the accuracy of the classifier. So an accurate pattern recognition system which can be used in different applications strongly needs a high performance classifier. One of the powerful classification techniques is support vector machine, briefly called SVM [1]. SVM is a supervised learning method that constructs a classification model using training data. SVM minimizes the generalization error and maximizes the geometric margin between two classes. This classifier uses a kernel function to map the input data into a high-dimensional feature space in order to find an optimal hyperplane to separate the two-class data. The performance of the SVM depends on the amount of kernel parameter, σ, and the amount of penalty factor, C. Also choosing an appropriate kernel function is important. Furthermore, selecting the useful features among several features in the training dataset to train SVM plays an important role in improving the performance of the SVM. So, before training the SVM, the user should select a suitable kernel function and also optimal amounts for kernel parameter and penalty factor. Besides that, as mentioned before, feature selection is important for improving the performance and reducing the complexity. To solve this problem different methods based on heuristic algorithms have been proposed. For example, Huang and Wang have used GA to optimize the SVM's parameters and also performing feature selection simultaneously in order to increase the classification accuracy [2]. They used RBF kernel in all experiments. Samanta et al. have proposed a GA-SVM method for bearing fault detection in rotating machines [3]. They had genetic algorithm, optimize the parameters of SVM, and also perform feature selection to improve the SVM ability in recognizing the vibration signals. Wu et al. proposed a method, based on GA and SVM, for predicting bankruptcy [4]. They have used GA only to optimize the classifier's parameters without feature selection. Like GA, other optimization algorithms such as PSO and SA have been used to promote the SVM's performance in different practical fields like Biomedical [5–7] and Face Recognition [8]. Another important point that is not considered in the mentioned researches is the reliability of the classifier, which means the validation of the classifier's output. This is a very critical point that should be considered in selecting a classifier for different applications such as military and medicine. In all mentioned researches, the researchers have used only one fitness function to evaluate their methods. But, in addition to recognition score, calculating the reliability of the classifier's output is a good way to evaluate the performance of the classifier. Reliability means the validation of the classifier's output, for an unknown sample. In some problems, although the recognition score of a class is high, the corresponding reliability of that class may be low, and vice versa. Figure 1 shows this concept. According to Figure 1 the recognition score of the hollow circles is 100% but the corresponding reliability is (5/6) 83%. These numbers for dark circles are 80% and 100%, respectively.

In this study multiobjective form of PSO has been used to find optimal hyperplanes for two objective functions: recognition score and reliability. The remainder of this paper is organized as follows. In Section 2, SVM is briefly introduced. In Section 3, PSO and MOPSO algorithms are reviewed. In Section 4, two forms of artificial neural networks are reviewed as powerful methods in classification. In Section 5, the proposed method has been introduced. Section 6 shows the experimental results and the final section is devoted to conclusion.

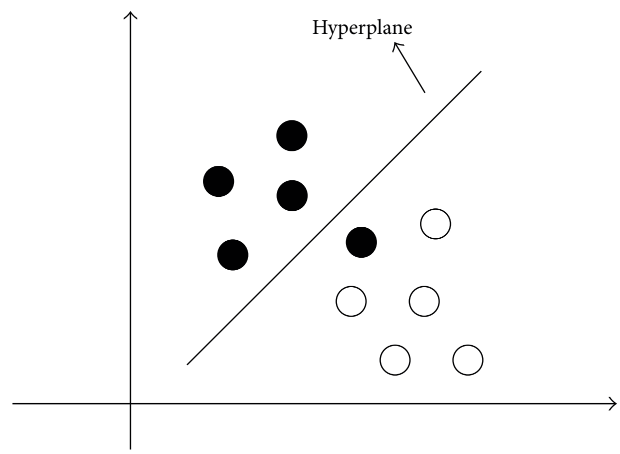


Figure 1: The recognition scores for the hollow circles and dark circles are 100% and 80%, respectively. The corresponding reliabilities are 83% and 100%, respectively.

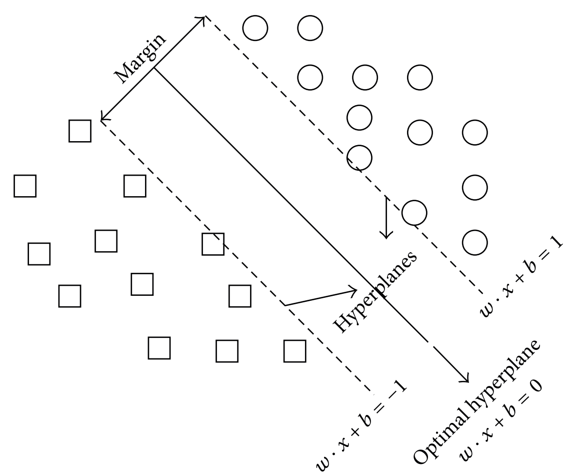


Figure 2: Optimal hyperplane.

2. Support Vector Machine

SVM is a two-class classifier described as follows [9]. Let (x_i, y_i), $1 < i < N$, indicate a set of data containing N training samples. Each sample must conform to the criteria $x_i \in R^d$. y_i demonstrates the class of corresponding sample, x_i. So $y_i \in \{-1, 1\}$ and d indicates the number of dimensions of input data. The separating hyperplane can be derived as in

$$w \cdot x_i + b = 0, \quad 1 \le i \le N. \tag{1}$$

If such a hyperplane exists, then linear separation is obtained. The samples which are nearest ones to the separating hyperplane are called support vectors. In boundaries (support vectors), (1) is reformed as

$$w \cdot x_i + b = \pm 1. \tag{2}$$

According to (2) for each sample (3) is true:

$$y_i \cdot (w \cdot x_i + b) \ge 1. \tag{3}$$

So the problem is finding w and b. There are numerous hyperplanes which can separate the two-class data but SVM produces the optimal hyperplane as indicated in Figure 2. This hyperplane has the maximum distance to support vectors. The margin of a separating hyperplane is $2/\|w\|$. So if we want to find the optimal hyperplane, we should minimize $\|w\|$. For simplicity we can substitute $(1/2)\|w\|^2$ with $\|w\|$. So we are dealing with an optimization problem. It means that we have to minimize $(1/2)\|w\|^2$ subjected to (3).

In Figure 2 the samples are linearly separable, but in most cases they cannot be separated as easy as indicated in Figure 2. For nonlinear problems positive slack variables ζ_i are introduced. So the problem changed into

$$\begin{aligned} \text{Min} \quad & \frac{1}{2}\|w\|^2 + C \cdot \sum_{i=1}^{n} \zeta_i \\ \text{s.t} \quad & y_i \cdot (w \cdot x_i + b) \ge 1 - \zeta_i, \\ & \zeta_i \ge 0, \\ & 1 \le i \le N. \end{aligned} \tag{4}$$

In (4) C is called penalty factor. It is introduced to control the tradeoff between margin maximization and error minimization. This problem can be solved by means of Lagrange multipliers. Thus the classification decision function becomes

$$F(x) = \operatorname{sign}\left(\sum_{i=1}^{N}\alpha_i \cdot y_i \cdot K\left(x_i, x_j\right) + b\right), \tag{5}$$

where α_i is the Lagrange multiplier. $K(x_i, x_j) = \varphi(x_i)\cdot\varphi(x_j)$ is kernel function through some another mapping function, $\varphi(x)$. QP solver is used to find α_i. After that w and b can be achieved by

$$w = \sum_{i=0}^{N}\alpha_i \cdot y_i \cdot \varphi\left(x_i\right), \tag{6}$$

$$b = \frac{1}{N_{\mathrm{SV}}}\sum_i\left(y_i - \sum_j \alpha_j \cdot y_j \cdot K\left(x_j, x\right)\right). \tag{7}$$

In (7) N_{SV} is the number of support vectors and x is the input unknown sample.

Some common kernel functions are

linear: $k(x, y) = x \cdot y + 1$,

polynomial: $k(x, y) = (x \cdot y + 1)^{\sigma}$,

RBF: $k(x, y) = \exp(-\|x - y\|/(2\cdot\sigma^2))$,

quadratic: $1 - \|x - y\|^2/(\|x - y\| + \sigma)$,

in all of these functions σ should be optimally tuned with C.

3. Particle Swarm Optimization Method

3.1. Single-Objective PSO. Particle swarm optimization algorithm is first suggested by Kennedy and Eberhart in 1995 [10]. This algorithm is produced by inspiration of birds flocking and fishes grouping. In fact they used the mechanism of birds flocking to solve optimization problems. It means that a group of particles search the solution space for the best solution. Each particle has a position, velocity, and a memory to save its best position from the beginning of the process. In each iteration the particle which has the best position is regarded as the leader and the other particles tend to reach its position. So their movement is affected by two factors: their best position from the first iteration to current iteration and the leader's position. Equations (8) and (9) describe how particles move through iterations:

$$\begin{aligned} v_{id}^{t+1} &= w \cdot v_{id}^{t} + c_1 \cdot \operatorname{rand} \cdot \left(p_{best}^{d} - x_{id}^{t}\right) + c_2 \\ &\quad \cdot \operatorname{rand}\left(p_{gbest}^{d} - x_{id}^{t}\right), \end{aligned} \tag{8}$$

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

In the above equations, v_{id} is the dth dimension of the velocity of the ith particle, x denotes the position of the particle, t is the number of iterations, c_1 and c_2 are learning factors, rand is a positive random number between 0 and 1 under normal distribution, w is the inertia weight coefficient, p_{best} is the best position of the particle from the beginning to current iteration, and p_{gbest} shows the position of the leader in each iteration.

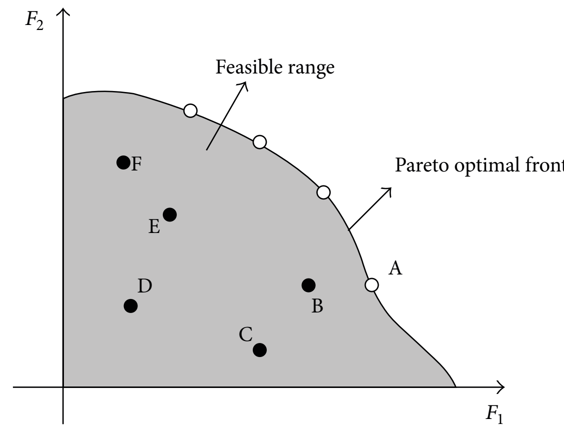


Figure 3: Pareto optimal front.

3.2. Multiobjective PSO. In a multiobjective optimization problem obviously, there is more than one objective function, to be optimized, so a multiobjective optimization problem can be defined as follows [11]:

$$\begin{aligned} \text{Minimize} \quad & F(x) = \left[f_1(x), f_2(x), \ldots, f_k(x)\right] \\ \text{s.t} \quad & g_j(x) < 0, \\ & h_j(x) = 0, \end{aligned} \tag{10}$$

where $x = (x_1, x_2, \ldots, x_n)$ is a solution, f_i, $i = 1, \ldots, k$, are objective functions, and g_j, h_j are constraints of the problem. Contrary to single-objective case, here we cannot find a single solution which is the best for all objective functions. Instead we are looking for a set of solutions. Actually there is a tradeoff between different objective functions. So the definition of the optimality is different in this case. We call x an optimal solution if another solution, like Y, cannot be found which has better fitness in all objective functions. Such a solution is a member of Pareto optimal front [12]. We say x_1 is dominated by x_2, if x_2 is better than x_1 in all objective functions. But if x_1 is better just in one objective function than x_2, it is nondominated. So in multiobjective form we have a set of solutions that contains nondominated particles. It means that the members of this set cannot dominate each other. Figure 3 shows Pareto optimal front for a two-objective function problem. According to this picture the solutions in the Pareto front dominate the other solutions but cannot dominate each other. In MOPSO each particle has a set of leaders and has to select one of them through a mechanism. Usually this set is called External Archive [13, 14]. External Archive contains nondominated particles from the first iteration.

In fact External Archive preserves outputs of the algorithm. Up to now different versions of MOPSO are introduced. In this study we have used the one introduced in [15] because of its speed and rapid convergence. In this form to select a leader for each particle, the solution space is divided into numerous hypercubes and different solutions from the External Archive exist in these hypercubes.

They are placed in hypercubes according to their coordination calculated by objective functions. Each hypercube is evaluated through dividing the number of its solutions into a constant number. After evaluating each hypercube, roulette wheel mechanism will select one of these hypercubes. And finally a solution, placed in the selected hypercube, will be selected randomly as the leader for the particle. MOPSO process is described as follows:

(1) Initializing the position and the velocity of each particle.

(2) Evaluating the particles.

(3) Saving nondominated particles in a repository.

(4) Producing hypercubes to cover the solution space.

(5) Initializing the memory of each particle

$$p_{best}\,[i] = \text{position}\,[i]\,. \tag{11}$$

(6) Main loop

   (a) Calculating the velocity of each particle by (8) (but in this form p^{d}_{gbest} should be replaced by $rep[h]$).

   (b) Updating the position of the particles through (9).

   (c) Evaluating the particles.

   (d) Updating the repository.

   (e) Updating p_{best} for each particle.

(7) End of the main loop.

4. Artificial Neural Networks

Artificial neural network is introduced in 1974 [16]. The aim of this network is to extract logical results from received information by simulating the activity of the brain using a similar structure. In fact, artificial neural networks are organized in such a way that the relationships between inputs and outputs (which can be complex or nonlinear) are saved in a network structure and are therefore capable of assigning the related output to each of the inputs. After determining the structural components of these networks, the components of this structure are modified based on numerous comparisons between the output of the network and the desired output, so that the difference between these two values approaches zero over consecutive comparisons. In this sense, a neural network can be considered as a blind model that is able to perform the mapping (not necessarily linear) from input (vector) space to output (vector) space. In this paper we have used two of the most widely used artificial neural networks, the multilayer perceptron neural network (MLP) and radial basis function neural network (RBF), and totally compared capability of them with optimal support vector machine.

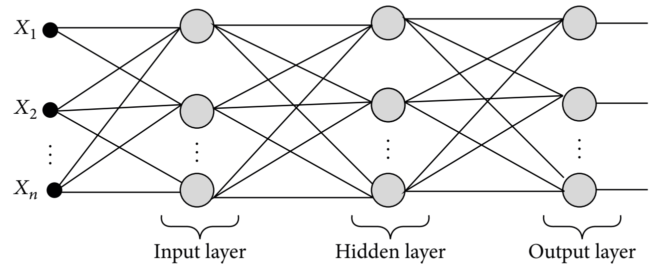


Figure 4: A multilayer perceptron neural network.

4.1. The Multilayer Perceptron Neural Network (MLP). The simplest perceptron neural network consists of three (input, hidden, and output) layers as shown in Figure 4. The numbers of neurons in each layer are determined using the trial and error method. The initial weights of this neural network are determined randomly. The backpropagation error algorithm is used for training the neural network in which the weights of the network change in a supervised manner based on the difference between the neural network output and desired output, so, for the every input, the output can be generated by the neural network. The input and output patterns are first normalized by a normalizing factor in order to equalize the effect of training process in changing the weights of the network in the training process. For the pth input pattern, the squared error in all neurons is calculated using the following equation:

$$E_p = \frac{1}{2}\left(d^p - y^p\right)^2 = \frac{1}{2}\sum_{j=1}^{n_j}\left(d_j^p - y_j^p\right)^2, \tag{12}$$

where d_j^p and y_j^p are, respectively, the values for desired output and calculated output in the jth neuron for pattern p. Total squared error for all patterns can also be calculated using the following equation:

$$E = \sum_{p=1}^{N} E_p = \frac{1}{2}\sum_{p=1}^{n_p}\sum_{j=1}^{n_j}\left(d_j^p - y_j^p\right). \tag{13}$$

In the following equations $w_{ij}(t+1)$ represents current weight, $w_{ij}(t)$ represents previous weight, η represents learning coefficient, and α represents momentary coefficient:

$$\begin{aligned} w_{ij(t+1)} &= w_{ij}\,(t) + \eta d\Delta w_{ij}\,(t) + \alpha\Delta w_{ij}\,(t+1)\,, \\ \Delta w_{ij} &= -\left(\frac{\delta E_p}{\delta w_{ij}\,(t)}\right). \end{aligned} \tag{14}$$

In this method weights are updated repeatedly for all learning patterns. The training process stops when the total error value for all patterns reaches a value lower than the determined critical point or when the whole learning period reaches the final point. It is noteworthy that the training method mentioned here is an error backpropagation method with momentary term, which lowers the possibility of coordination at local minima compared with the error backpropagation method.

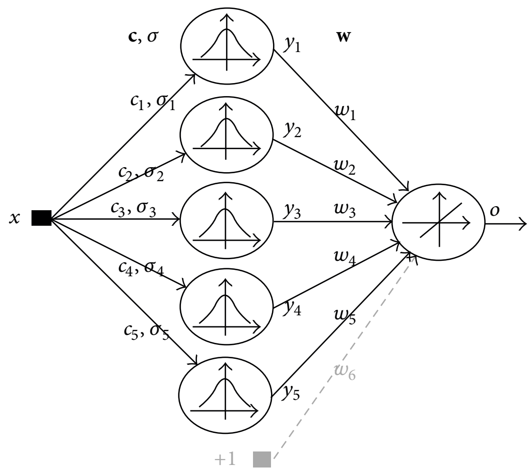


Figure 5: A radial basis function neural network.

4.2. Radial Basis Function (RBF) Neural Network. RBF is a popular supervised neural network learning algorithm. It is a specific kind of MLP network [17]. The RBF network is constituted by only the following three layers as shown in Figure 5:

Input Layer. It broadcasts the inputs without distortion.

RBF Layer. Hidden layer contains the RBF.

Output Layer. Simple layer contains a linear function.

Basis functions normally take the form $\phi = \|\vec{x}_i - \vec{\mu}_i\|$. The function depends on the distance (usually taken to be Euclidean) between the input vector $\vec{x}$ and a vector $\vec{\mu}_i$. The most common form of basis function used is the Gaussian function

$$\phi = \exp \frac{\|\vec{x}_i - \vec{\mu}_i\|}{2\sigma_j^2}, \tag{15}$$

where $\vec{\mu}_i$ determines the center of basis function and σ_j is a width parameter that controls how the curve is spread. Generally, these centers are selected by using some fuzzy or nonfuzzy clustering algorithms. In this work, we have used the K-means algorithm to select the initial cluster centers in the first stage and then these centers are further fine-tuned by using point symmetry distance measure. The number of neurons in the output layer is equal to the number of classes of the classification problem. Each output layer neuron computes a linear weighted sum of the outputs of the hidden layer neurons as follows:

$$y_i(x) = \sum_{i=1}^{N} \phi_i(X) \cdot w_i. \tag{16}$$

The weight vectors are determined by minimizing the mean squared differences between the classifier outputs:

$$y_k = \sum_{j=0}^{m} w_{k,j} s_i. \tag{17}$$

C	K_n	σ	F_1	$\cdots$	F_n

Figure 6: Construction of particles.

And target values t_k are as follows:

$$E = \frac{1}{2} \sum_{k=1}^{M} (y_k - t_k)^2. \tag{18}$$

The parameters $(\Delta w, \Delta\mu, \Delta\sigma)$ are given by (for more explication, see [17])

$$\frac{\partial E}{\partial w_{ki}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial w_{ki}} \tag{19}$$

or

$$\frac{\partial E}{\partial y_k} = (t_k - y_k). \tag{20}$$

Thus

$$\frac{\partial E}{\partial w_{ki}} = -(t_k - y_k) s_i. \tag{21}$$

After computation, we obtain

$$\begin{aligned} \frac{\partial E}{\partial \mu_{ji}} &= \sum_k \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial s_j} \frac{\partial s_j}{\partial \mu_{ji}} \\ &= \frac{s_j}{\sigma_j^2} (x_i - \mu_{ji}) \sum_{i=1} (t_k - y_k) w_{kj}, \\ \frac{\partial E}{\partial \sigma_{ji}} &= \sum_k \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial s_j} \frac{\partial s_j}{\partial \sigma_{ji}} = \frac{2 s_j}{\sigma_j} \log s_j \sum_{i=1} (t_k - y_k) w_{kj}. \end{aligned} \tag{22}$$

5. Proposed Method

In this paper we have used MOPSO to optimize penalty factor, choose adequate kernel function, tune the selected kernel's parameter, and feature selection for two objective functions, recognition score and reliability, and its performance is compared with RBF and MLP neural networks. The construction of particles is indicated in Figure 6.

The first variable, C, is for tuning penalty factor. K_n is for selecting kernel functions. The amount of this variable can be 1, 2, 3, or 4 to choose one kernel among the four kernels introduced in Section 2. σ is for selecting the selected kernel's parameter (except linear). The rest of the particle is for feature selection. For a dataset with n number of features, $F_1, F_2, \ldots, F_n$ are between 0 and 1. If they are less than or equal to 0.5, the corresponding feature is not selected. Conversely if they are bigger than 0.5, the corresponding feature is selected.

If we consider the two classes as "positive" and "negative," then the predicted test samples can be divided into four groups:

(1) Samples which are "positive" and correctly predicted as "positive" (TP).

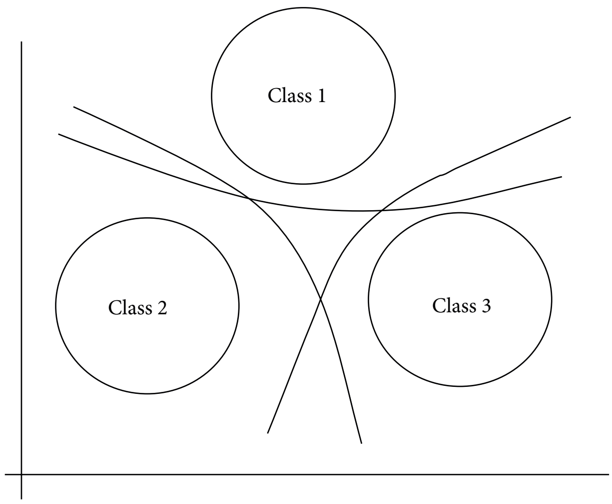


Figure 7: Classifying of a 3-class dataset with one-versus-all method.

(2) Samples which are "positive" but classified as "negative" (FN).

(3) Samples which are "negative" and correctly classified as "negative" (TN).

(4) Samples which are "negative" but predicted as "positive" (FP).

According to this categorization, recognition score is calculated by

$$\text{Recognition Score} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{23}$$

and the reliability for each class equals

$$\begin{aligned} \text{Pos-reliability} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{Neg-reliability} &= \frac{\text{TN}}{\text{TN} + \text{FP}}. \end{aligned} \tag{24}$$

The termination criteria are that the iteration number reaches 200. To calculate the fitness functions, for each particle, SVM should be trained by the determined parameters, kernel function, and selected features and then recognition score and reliability for each class can be achieved by (23) to (24). For multiclass classification we have used one-versus-all method. In this method for each class of the dataset we found the optimal hyperplane, which separates the corresponding class from the others. Thus the input sample is labeled according to the opinion of the obtained hyperplanes about that sample. Figure 7 shows this method for a 3-class dataset.

6. Experimental Results

The suggested method applied to nine different datasets from UCI machine learning repository [18]. In Table 1 the characteristics of these datasets are shown. Table 2 shows the experimental results on these datasets, Table 3 contains the learning time for different methods, and Table 4 shows the results of proposed method in classifying different datasets with and without feature selection.

Table 1: Characteristics of used datasets.

Dataset	Number of classes	Number of samples	Number of features
Glass	6	214	9
Iris	3	150	4
Wine	3	175	13
German	2	1000	20
Ionosphere	2	351	33
Sonar	2	208	60
Hepatitis	2	80	19
Bupa	2	345	6
Vowel	11	990	13
Heart	2	270	13

According to Table 2 it can be seen that MOPSO-SVM gives comparable and also better results than MLP and RBF neural networks for Glass, Iris, Wine, Ionosphere, Hepatitis, and Vowel datasets. The important point demonstrated in Table 2 is the rates of reliabilities given for different datasets. As indicated in Table 2, the proposed method gives high rates of reliabilities for most of the datasets, meaning that the output of the promoted classifier is strongly reliable.

In fact since the hyperplanes obtained by MOPSO have an amount of errors in classifying of the test samples (unknown samples), some samples exist that more than one hyperplane assigns them to their corresponding classes. Also there may be some samples that none of the hyperplanes assign them to their corresponding classes. Such samples are considered as error samples, at which their classes cannot be distinguished. Figure 8 illustrates this concept. Another point that is obviously seen from Table 2 is that MOPSO-SVM outperforms the original SVM in most of the experiments. It means that the proposed method is an expert classifier which automatically finds the optimal SVM parameters and best feature subset for classifying different datasets. It should be noted that in all the experiments different kernel functions were chosen for single SVM and the amounts of the recognition score and reliability reported for single SVM written in Table 2 are the average results of different SVM with different kernel functions.

Analyzing the numbers seen in Table 2, we can conclude that MOPSO-SVM is a powerful and effective classifier, due to rates of reliabilities and recognition scores achieved by this method for different datasets. These numbers show that MOPSO-SVM is a reliable classifier which means that this promoted classifier can act perfectly in special applications such as military and medicine which strongly require a high-reliable classifier. Table 3 contains the learning time for different algorithms. Comparing to single SVM, MOPSO-SVM requires less learning time in most experiments. This is the result of feature selection. In fact removing redundant features from datasets results in reduction of learning time. Also proposed method has less learning time than MLP and RBF neural networks. In Table 4 the results of

Table 2: Percentage of recognition score and reliability.

	Glass	Iris	Wine	German	Ionosphere	Sonar	Hepatitis	Bupa	Vowel	Heart
MOPSO-SVM										
Recognition score	81.31	94.67	97.75	84.20	92.31	90.87	96.25	82.32	97.78	87.41
Reliability	92.94	97.93	100	89.89	93.99	90.85	92.095	82.06	99.89	87.3
SVM										
Recognition score	61.21	82.33	90.45	78.35	92.59	83.41	92.5	73.04	97.17	83.88
Reliability	94.64	98.45	99.65	79.57	92.98	87	89.87	72.77	99.89	85.49
MLP										
Recognition score	82.78	98.54	98.42	89.86	96.44	93.76	92.76	87.94	77.6	92.8
Reliability	73.088	98.68	98.438	88.35	96.35	93.97	86.51	87.59	78.30	92.82
RBF										
Recognition score	81.76	96.92	81.58	91.5	90.02	94.72	94.78	88.12	99.12	83.7
Reliability	75.822	96.96	88.87	94.94	93.11	94.86	97.07	91.49	99.3	86.59

Table 3: Learning time for different classifiers (second).

	Glass	Iris	Wine	German	Ionosphere	Sonar	Hepatitis	Bupa	Vowel	Heart
MOPSO-SVM	0.64	0.204	0.224	0.635	0.184	0.084	0.0436	0.191	11.85	0.118
SVM	2.74	1.138	1.12	0.663	0.179	0.116	0.0423	0.148	8.60	0.135
MLP	22.17	2.14	2.06	37.84	2.24	3.43	2.12	4.93	38.13	2.33
RBF	5.27	3.55	4.08	21.17	11.28	5.98	3.005	3.78	18.9	6.21

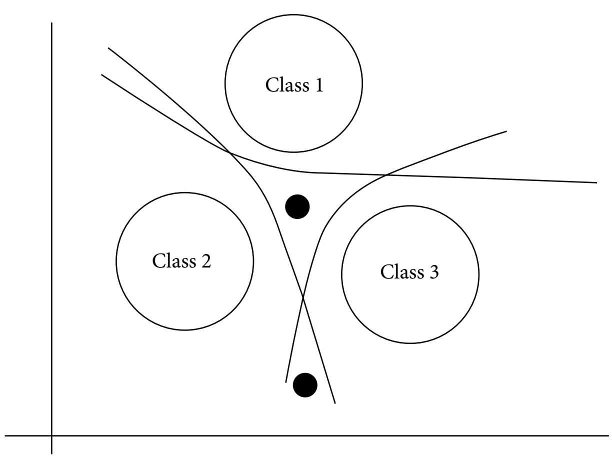


Figure 8: Samples which are considered as error samples.

proposed method with and without feature selection are shown. According to this table, feature selection process has improved the recognition score and reliability for most of the datasets. It means that feature selection process is an efficient preprocessing technique which not only has the ability to reduce the learning time of the classifier but also can improve its performance. This is an important issue especially in classifying or clustering high-dimensional data. From the reported results, it is clear that using heuristic algorithm to enhance the performance of the SVM for two objective functions is a successful idea because finding optimal parameters of SVM for different datasets and also reducing the dimension of the dataset are a hard task. For example for Sonar samples, which have 60 features, there exists 2^{60} feature subset, so it is very difficult to find the best feature subsets. Furthermore finding the optimal amounts of the parameters in order to improve the performance of the SVM is a difficult task. In fact finding an optimal SVM with optimal feature subset is an NP-hard problem which can be solved with heuristic algorithm. According to the reported results, MOPSO searches the solution space very effectively.

7. Conclusion

In this study multiobjective PSO has been used to tune the parameters of SVM and also perform feature selection for two objective functions and the performance of the proposed method (MOPSO-SVM) has been compared with single SVM, RBF, and MLP neural networks. According to the reported results, it can be seen that the proposed method gives reliabilities and recognition scores, comparable with RBF and MLP neural networks, which have shown their effectiveness in classifying overlapped datasets, and in some cases even gives better reliabilities and/or recognition scores than RBF and MLP, for example, for Glass, Iris, Wine, Ionosphere, Hepatitis, and Vowel datasets. Also the proposed method has less learning time in most of the experiments. Furthermore according to Tables 3 and 4, feature selection is an important preprocessing method which has positive effect both on learning time and on the accuracy of the classifier.

Actually the results shown in the previous section indicate that using heuristic algorithm to convert SVM from a normal classifier into an expert one was successful. Furthermore optimizing SVM in order to increase its reliability besides its accuracy by using a multiobjective heuristic algorithm is a successful idea according to the obtained results.

Table 4: MOPSO-SVM with and without feature selection.

	Glass	Iris	Wine	German	Ionosphere	Sonar	Hepatitis	Bupa	Vowel	Heart
With feature selection										
Recognition score	81.31	94.67	97.75	84.20	92.31	90.87	96.25	82.32	97.78	87.41
Reliability	92.94	97.93	100	89.89	93.99	90.85	92.095	82.06	99.89	87.3
Without feature selection										
Recognition score	71.50	95.33	97.19	84.30	92.02	87.98	95	81.16	96.06	84.07
Reliability	84.5	96.68	98.41	90.84	94.12	88.81	90.815	80.64	99.68	83.89

The reported results also show the power and effectiveness of MOPSO in searching the solution space. In other words, MOPSO is a powerful algorithm which can act very effectively in solving multiobjective optimization problems.

Competing Interests

The authors declare that they have no competing interests.

References

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[2] C.-L. Huang and C.-J. Wang, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Systems with Applications*, vol. 31, no. 2, pp. 231–240, 2006.

[3] B. Samanta, K. R. Al-Balushi, and S. A. Al-Araimi, "Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 7-8, pp. 657–665, 2003.

[4] C.-H. Wu, G.-H. Tzeng, Y.-J. Goo, and W.-C. Fang, "A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy," *Expert Systems with Applications*, vol. 32, no. 2, pp. 397–408, 2007.

[5] F. Melgani and Y. Bazi, "Classification of electrocardiogram signals with support vector machines and particle swarm optimization," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 5, pp. 667–677, 2008.

[6] 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.

[7] Q. Shen, W.-M. Shi, W. Kong, and B.-X. Ye, "A combination of modified particle swarm optimization algorithm and support vector machine for gene selection and tumor classification," *Talanta*, vol. 71, no. 4, pp. 1679–1683, 2007.

[8] J. Wei, Z. Jian-Qi, and Z. Xiang, "Face recognition method based on support vector machine and particle swarm optimization," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4390–4393, 2011.

[9] 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.

[10] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE Service Center, Perth, Australia, 1995.

[11] A. Abraham and L. Jain, "Evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, A. Abraham and R. Goldberg, Eds., Advanced Information and Knowledge Processing, pp. 1–6, Springer, London, UK, 2005.

[12] V. Pareto, *Cours d'Economie Politique, Volume I and II*, F. Rouge, Lausanne, Switzerland, 1896.

[13] M. Reyes-Sierra and C. A. C. Coello, "Multi-objective particle swarm optimizers: a survey of the state-of-the-art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287–308, 2006.

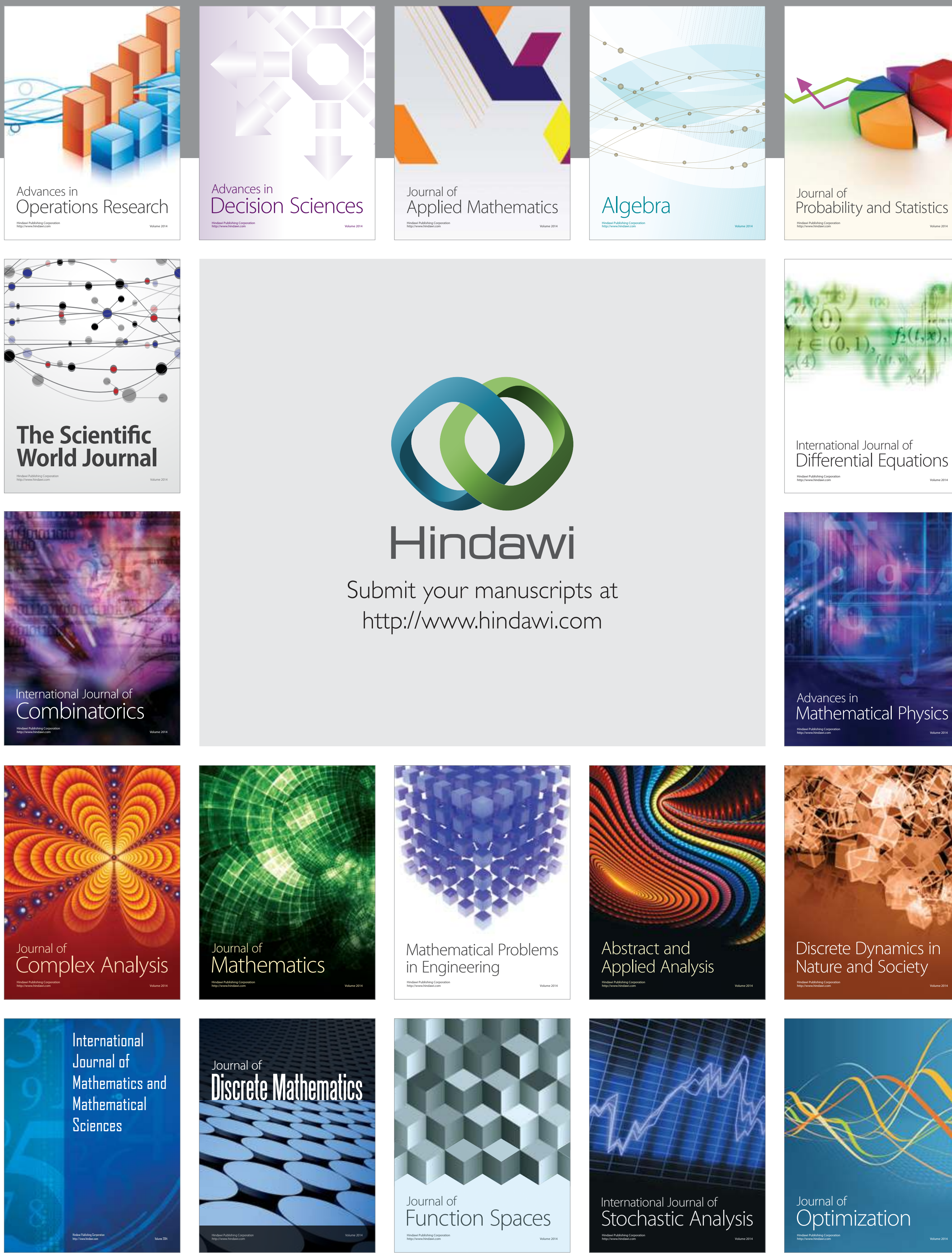
[14] M. Bhuvaneswari, *Application of Evolutionary Algorithms for Multi-Objective Optimization in VLSI and Embedded Systems*, Springer, Berlin, Germany, 2015.

[15] C. A. C. Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 1051–1056, Honolulu, Hawaii, USA, May 2002.

[16] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College, New York, NY, USA, 1996.

[17] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[18] S. Hettich, C. Blake, and C. Merz, "UCI repository of machine information and computer sciences," 1988, http://www.ics.uci.edu/~mlearn/MLRepository.htm.


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
Hindawi
Submit your manuscripts at
http://www.hindawi.com
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