Support vector machine with parameter optimization by a novel hybrid method and its application to fault diagnosis

Xiaoyuan Zhang a,*, Daoyin Qiu b, c, Fuan Chen a

a College of Electrical Engineering, Henan University of Technology, Zhengzhou 450001, PR China
b Huazhong University of Science and Technology, Wuhan 430074, PR China
c North China University of Water Conservancy and Hydropower, PR China

A R T I C L E   I N F O
Article history:
Received 26 March 2014
Received in revised form 3 July 2014
Accepted 4 August 2014
Communicated by J. Zhang

Keywords:
Support vector machine
Parameter optimization
Differential evolution
Particle swarm optimization
Bare bones differential evolution
Fault diagnosis

A B S T R A C T
The performance of support vector machine (SVM) heavily depends on its parameters. The parameter optimization for SVM is still an ongoing research issue. The current parameter optimization methods either are easy to fall into local optimal solution, or are time consuming. Moreover, some optimization methods depend also on the choice of parameters for them, provoking thus a vicious circle. In view of this, a new hybrid method is proposed to optimize the parameters of SVM in this paper. It uses the inter-cluster distance in the feature space (ICDF) to determine a small and effective search interval from a larger kernel parameter search space, while a hybrid of the barebones particle swarm optimization and differential evolution (BBDE) is used to search the optimal parameter combination in the new search space. The ICDF shows the degree the classes are separated. The BBDE is a new, almost parameter-free optimization algorithm. Some benchmark datasets are used to evaluate the proposed algorithm. Furthermore, the proposed method is used to diagnose the faults of rolling element bearings. Experiments and engineering application show that the proposed method outperforms other methods both mentioned in this paper and published in other literature.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

 Nowadays, with the economics pressure to reduce the costs, the downtime of industrial plants and to shorten the time necessary to manufacture a product, the fault diagnosis (FD) plays an important role. FD makes it possible to determine the place, reason and time of the fault precisely. An effective and efficient FD method is an important issue most techniques have sought to. As a state-of-the-art pattern recognition method, support vector machine (SVM) has been vastly used in FD [1–5]. The theory of SVM is based on the structural risk minimization (SRM) principle [6]. It has a good performance in solving nonlinear and high dimensional pattern recognition problems with small-sample size. However, poor choice of parameter setting can dramatically decrease the generalization performance of SVM. The parameters of SVM includes the value of the regularization constant C and the Kernel type, with its respective parameters. Up to now, there is no systematic methodology or priori knowledge for determining the parameters of SVM. Thus, it will be desirable to have an effective and automatic model selection scheme to make SVM practical for real life applications, in particular, for people who are not familiar with parameters tuning in SVM [7].

Many studies have been carried out on SVM parameters optimization. A straightforward way is the simple exhaustive grid search (GS) in the parameter space [8]. GS trains SVMs with all desired combinations of parameters and screens them according to the training accuracy. It makes the training process time-consuming, and when the number of parameters exceeds two it will become intractable. Furthermore, to reach high performances, GS needs a good discretization of the search space in fixed values, which is a tricky task since a suitable sampling step varies from kernel to kernel and the grid interval may not be easy to locate without prior knowledge of the problem [9]. Therefore, many improved parameters selection methods have been proposed. Chapelle et al. [10,11] used gradient-based numerical optimization methods to find the optimal parameters for SVM. By minimizing some generalization bounds such as the leave-one-out (LOO) error bounds, the numerical methods are generally more efficient than GS. There are also other numerical optimization approaches in the literature [12–16]. However, owing to the non-convexity of the generalization bounds, these methods may get stuck into local optimum and cause instabilities. Moreover, these methods are sensitive to the initial point and require that the kernel functions and the bounds of generalization error have to be differentiable with respect to kernel and penalty constant parameters. To
overcome the above limitations, some intelligent evolutionary algorithms (EA), such as genetic algorithm (GA) [17–21], covariance matrix adaptation evolution strategy (CMA-ES) [22], particle swarm optimization (PSO) [23–26], artificial immunization algorithm (AIA) [4] and ant colony optimization algorithm (ACO) [27,28] have been used to optimize the SVM parameters for their better global search abilities. EAs do not need additional information of the problem, and are generally good at finding “acceptable good” or near-optimal solutions to problems. But they are not well suited to perform finely tuned search and thus not guaranteed to find the global optimum solution to a problem. Furthermore, EAs select the best parameter combination from the population evolved generation by generation, which requires re-training SVMs many times. Thus, they are still time-consuming, especially when the search space of the parameters is large. Moreover, the performance of most EAs depend also on the choice of parameters for them, such as, for instance, the inertia weight in PSO, provoking thus a vicious circle.

From the aforementioned literature review, it can be seen that the presence of local minima in the search space, the computational time required for model selection task and the vicious circle in SVM parameter optimization process have been considered the main challenges in the field. In view of this, a novel SVM parameter optimization algorithm is presented in this paper. In the proposed method, the range of the kernel parameter is first shortened by the use of an inter-cluster distance in the feature space (ICDF) measure heuristic, which can reduce the calculation amount of the parameter optimization process and improve the chances to find the optimal parameters. Afterwards, an almost parameter-free optimization method, bare bones differential evolution (BBDE), is used to optimize the parameter values. BBDE is very powerful yet almost no parameter to be tuned. It can avoid of the vicious circle of traditional EAs.

In the following part of the introduction, we briefly review the ICDF index and the BBDE algorithm. ICDF indicates the degree the classes are separated in the feature space. When kernel function is fixed, selecting kernel parameters is equivalent to selecting the feature space. A feature space in which the classes are more separated corresponds to a better kernel parameter value for SVM. In literature [29,30], Wu and Wang used ICDF as the separation index to choose the kernel parameters for SVM. In their study, for binary classification problem, the optimal kernel parameter is determined according to the largest value of ICDF. Then, the selected kernel parameter with different candidate penalty parameters are used to train SVM models. The parameters combination which results in a highest cross validation accuracy is selected as the best one. Since calculating ICDFs does not require the information of the trained SVMs, the time needed for the training process for different kernel parameters is saved. This method can get the parameter combinations which result in SVM models perform as good as the models chosen by the traditional GS method in testing accuracy, while the training time is shortened largely. However, in their method, the candidate penalty parameters and kernel parameters for calculating ICDFs are given as discretization. It needs to locate the interval of feasible solution and a suitable sampling step, which is a tricky task. Furthermore, in practice, most of the classification problems are multi-class, how to extend the binary SVM with parameter optimization by ICDF to multi-class problems is a challenging task.

Bare bones differential evolution (BBDE), proposed by Omran et al. [31], is a hybrid of the barebones particle swarm optimizer (PSO) [32] and differential evolution (DE) [33]. It is an almost parameter-free, self-adaptive, optimization algorithm. BBDE capitalizes on the strengths of both the barebones PSO and self-adaptive DE. The BBDE does not make use of the standard PSO parameters (i.e. inertia weight, acceleration coefficients, and velocity clamping), and also removes the DE scale parameter. The only parameter is the probability of recombination. This is a highly favorable characteristic for parameter optimization of SVM, as it can overcome the aforementioned vicious circle.

The proposed method makes full use of ICDF measure and BBDE algorithm, in which ICDF measure is responsible to determine a small and effective search interval both for binary classification problem and multi-class classification problem, from a larger kernel parameter search space, then BBDE is used to search the optimal parameter combination of SVM in the continuous intervals of kernel parameter and penalty parameter. That is, the exhaustive search of parameters can be performed in the priority area indicated by ICDF measure. The search range of kernel parameter in the proposed method is much smaller yet more effective than that of traditional methods. Moreover, BBDE has shown performance superior to that of PSO and other EAs in the widely used benchmark problems [31] with almost no parameters being set. The proposed method is tested on several benchmark datasets as well as fault diagnosis for rolling element bearings. Experiments and engineering application show that the proposed method outperforms other methods both mentioned in this paper and published in other literature.

The remaining of this paper is organized as follows. The brief introduction of SVM and its parameter selection are presented in Section 2. The proposed method is described in detail in Section 3. In Section 4 the proposed method and some other optimization methods are applied in some benchmark datasets and the experimental results are compared. In Section 5 the proposed method is applied in FD for rolling element bearings. Finally, the conclusion is drawn in Section 6.

2. Support vector machine and its parameter selection

Support vector machine (SVM) is a statistical classification method based on the structural risk minimization (SRM) approach proposed by Vapnik et al. [6]. Given a dataset with n examples \((x_i, y_i)\), where each \(x_i\) is an input data and \(y_i \in \{-1,1\}\) corresponds to its bipolar label, \(i = 1,2,...,n\). By using a nonlinear mapping \(\phi(x)\), the input data is mapped into a high dimensional feature space \(F\), in which the data are sparse and possibly more separable. Then, the maximum margin separating hyperplane \(w(x) + b = 0\) is built in \(F\), where \(w\) is a weight vector orthogonal to the hyperplane, \(b\) is an offset term. The margin is \(1/||w||\). Maximizing the margin \(1/||w||\) is equivalent to minimizing \(||w||^2\), whose solution is found after resolving the following quadratic optimization problem:

\[
\min_{w,b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i
\]

Restrictions : \(y_i(w(x_i) + b) \geq 1 - \xi_i\), and \(\xi_i \geq 0\) for \(i = 1,2,...,n\)

where \(C\) is the penalty parameter which imposes a trade-off between training error and generalization, \(\xi_i\) are slack variables. The restrictions are imposed to ensure that no training data should be within the margins. Nevertheless, they are relaxed by the slack variables to avoid an overfitting to noisy data. The number of training errors and examples within the margins allowed by the introduction of these variables is controlled by the minimization of the term \(\sum_i \xi_i\). By using the duality theory of optimization theory, the final decision function can be given by

\[
f(x) = \operatorname{sgn}(\sum_{x \in SVs} y_i a_i \phi(x) \phi(x_i) + b) + b
\]

where the constants \(a_i\) are called Lagrange multipliers and are determined in the optimization process, SVs corresponds to the set of support vectors, training examples for which the associated
Lagrange multipliers are larger than zero. These examples are those closest to the optimal hyperplane. For all other examples, the associated Lagrange multiplier is null, so they do not contribute to the determination of the final hypothesis. $sgn(x)$ is a signal function. When $f(x) > 0$, the predicted class is +1, if $f(x) < 0$, the class is −1, and if $f(x)=0$ the class is considered unknown or it is randomly assigned.

$$K(x_i,x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

is so called kernel function, which allows access to spaces of high dimensions without the need to know the mapping function $\phi(x)$ explicitly. From Eqs. (1) and (2) we can see that the performance of SVM largely depends on the type of kernel function, the parameter of kernel function and penalty parameter $C$.

There are several kernel functions, and the generally used kernel functions are given as follows:

Linear kernel

$$K(x_i,x_j) = \langle x_i, x_j \rangle$$

Polynomial kernel

$$K(x_i,x_j) = (\gamma \langle x_i, x_j \rangle + r)^d, \ \ \ \ \ \gamma > 0$$

RBF kernel

$$K(x_i,x_j) = \exp(-\gamma||x-x_i||^2), \ \ \ \ \ \gamma > 0$$

Sigmoid kernel

$$K(x_i,x_j) = \tan \theta(\gamma \langle x_i, x_j \rangle + r)$$

where $\gamma, d, r$ are kernel parameters for each kernel function.

Among these kernel functions, the RBF kernel can map the sample set from the input space into a high-dimensional feature space effectively, which is good for representing the complex nonlinear relationship between the input and output samples [8,13]. The linear kernel can be regarded as a special case of RBF kernel. The sigmoid kernel behaves similarly to the RBF for certain parameters and is not better than the RBF kernel in general. Even though the polynomial kernel may be an attractive alternative, but numerical difficulties tend to arise if a high degree is used, for example, a power of some minor value that 1 tends to 0 and of a major one that tends to infinity. Furthermore, the RBF kernel has often achieved a superior power of generalization with lower complexity than the polynomial kernel [28]. Recent works also suggested the use of RBF kernel function in the SVM model is appropriate and sufficient [8,13,16,20,30]. In view of the aforementioned reasons, the RBF is selected as kernel function for SVM in this study.

When RBF kernel is used, the kernel width $\gamma$ and the tradeoff variable $C$ should be set properly for SVM. To show the influence of various ($\gamma, C$) effect on the performance of SVM, Fig. 1 plots the fivefold average testing accuracy for four public available datasets: glass, vowel [34] and svmguide2, svmguide4 [8], in a three dimensional surface, where the $x$-axis and the $y$-axis are $\log_{10} C$ and $\log_{10} \gamma$, respectively. The $z$-axis is the fivefold average testing accuracy. Each mesh point in the ($x$, $y$)-plane stands for a parameter combination ($\gamma$, $C$). It is easy to see that the performance of SVM not only fluctuates dramatically with the changes of ($\gamma$, $C$), there also are many local maxima. Also these plots show that these surfaces have low-degree of regularity, which further hinders the use of traditional methods. Thus, it is significant to adopt some both effective and efficient algorithm to optimize the parameters of SVM.

3. The proposed method

3.1. Inter-cluster distance in the feature space

One of the important tricks of SVM is the introduction of kernels, which enables SVM to have the ability of dealing with infinite or nonlinear features in a high dimensional feature space. When kernel function is selected, the high dimensional feature space is only determined by kernel parameter. A better kernel parameter will result in a pair of more separated classes. The ICDF value can indicate the degree the classes are separated in the feature space.

In the current study, the Euclidian distance between the means of two classes in the feature space is adopted. For some other forms of ICDF can be found in [30,35].

Let the input vectors $x_i, i=1,...,m$, belong to class 1 and the input vectors $y_j, j=1,...,n$, belong to class 2, $\phi(\cdot)$ denotes the mapping function which maps the input vectors into the high dimensional feature space, $K(\cdot, \cdot)$ denotes the kernel function. The Euclidian distance between the means of two classes in the feature space is calculated by

$$\delta = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \phi(x_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(y_j)}$$

$$= \sqrt{\frac{1}{m} \sum_{i=1}^{m} \phi(x_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(y_j)}^2$$

$$= \sqrt{\frac{1}{m} \sum_{i=1}^{m} \phi(x_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(y_j)}$$

$$= \sqrt{\frac{1}{m} \sum_{i=1}^{m} \phi(x_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(y_j)}$$

$$= \sqrt{\frac{1}{m} \sum_{i=1}^{m} \phi(x_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(y_j)}$$

The Euclidian distance between the means of two classes in the feature space can indicate the class separation robustly. It performs well in most cases [30]. Here, Fisher’s Iris data [34] is taken as an example to describe this type of distance. Fig. 2 plots the two-dimensional graphic of Fisher’s Iris data and the distances between different class means, where the x-axis and the y-axis are petal length and petal width, respectively, $d_{setosa}$ denotes the distance between means of setosa and virginic, $d_{versicolor}$ denotes the distance between means of setosa and versicolor, $d_{veer}$ denotes the distance between means of versicolor and virginic. From Fig. 2 we can see that the location of class mean can indicate the class location, as the classes more separate away, the distance between class means increases.

3.2. Bare bones differential evolution

Barebones differential evolution (BBDE) algorithm, proposed by Omran [31], is a hybrid of the bare bones PSO [32] and self-adaptive DE. Both PSO and DE have their strengths and weaknesses. One of the problems which both algorithms share is that control parameters need to be optimized for each new problem, as best parameter values are problem dependent. PSO has the advantage that formal proofs exist to show that particles will converge to a single attractor. The barebones PSO utilizes this information by sampling candidate solutions, normally distributed around the formally derived attractor point. Additionally, the barebones PSO has no parameters to be tuned. DE has the advantage of not being biased towards any prior defined distribution for sampling mutational step sizes and its selection operator follows a hill-climbing process. Mutational step sizes are determined as differences between individuals in the current population. Furthermore, very efficient self-adaptive DE strategies have been developed that eliminate the need for optimizing control parameters [31].

The BBDE strategy capitalizes on the strengths of both barebones PSO and self-adaptive DE strategies, to form an efficient
A hybrid optimization algorithm. It outperforms DE, PSO, and bare-bones PSO with the added advantage of little, almost no parameter tuning. In BBDE, the mutation operator of DE is used to explore around the current attractor by adding a difference vector to the attractor. Crossover is done with randomly selected personal bests as these personal bests represent a memory of the best solution found by individuals since the start of the search process. A brief review of BBDE is given as follow.

Without loss of generality, global optimization problems can be transformed into solving the following minimization problem:

$$\text{min } f(x_1, x_2, \ldots, x_D)$$

s.t. $$x_j \leq x_j' \leq \frac{x_j}{C_0}$$

where $$D$$ is the dimension of solution space, $$x_j$$ and $$x_j'$$ are the lower and upper bound of $$x_j$$.

The process of BBDE algorithm is illustrated in Fig. 3. It mainly contains the following steps:

Step 1: Initialize the population: the initial population of $$NP$$ vectors is stochastically randomly selected based on uniform probability distribution.

$$x_{ij}(0) = x_{ij}^L + \text{rand}(0, 1)(x_{ij}^U - x_{ij}^L)$$

where $$i = 1, 2, \ldots, NP$$, $$j = 1, 2, \ldots, D$$, $$NP$$ denotes population size, $$x_{ij}(0)$$ is the $$j$$-th dimension of $$x_i(0)$$, $$\text{rand}(0,1)$$ denotes the random number uniformly distributed in $$(0,1)$$.

Step 2: Evaluate the fitness of all particles: measure the fitness of each particle in the current population.

Step 3: Update local best value and global best value.

Step 4: Check termination conditions: if termination criterion is satisfied, then stop the algorithm; otherwise, enter to Step 5.

Please cite this article as: X. Zhang, et al., Support vector machine with parameter optimization by a novel hybrid method and its application to fault diagnosis, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.08.010
satisfy with a randomly selected personal best, positions, borrowing from the barebones PSO. Crossover is done stochastic weighted average of personal best and global best between exploration and exploitation. In the process of BBDE difference vector between randomly selected particles, bests represent a memory of best solutions found by individuals.

3.3. Performance measure for SVM parameter optimization

In the process of tuning SVM parameters, one form of performance measure served as fitness function is needed to evaluate the selected parameters. The performance of SVM is mainly referred to its generalization capability, namely the capability of recognizing the new data. Obviously, the true risk of the SVM classifier is the best one. But this quantity is not accessible as the data distribution in real world is unknown. Thus, we have to select estimates or bounds for the true risk of the SVM classifier as the performance measures.

The commonest performance measures are probably the leave-one-out (LOO) procedure and the k-fold cross-validation, both of which require the learning engine be trained multiple times to obtain a performance measure for each parameter combination. In LOO procedure one sample is left out in turn for testing, and the training and testing will be repeated l times, where l is the number of training samples. LOO procedure gives an almost unbiased estimate of the expected generalization error. However, it is very costly to actually compute. k-fold cross-validation is an alternative for LOO. In k-fold cross-validation, the training data is randomly split into k mutually exclusive subsets (the folds) of approximately equal sizes. One subset is left out in turn for testing, and the training and testing will be repeated k times. By averaging the test errors over the k trials it gives an estimate of the expected generalization error. Literature [13] studied the usefulness of several simple performance measures that are inexpensive to compute such as k-fold cross – validation, Xi-alpha bound, Generalized approximate cross-validation, Approximate span bound, VC bound and Radius-margin bound. Their results point out k-fold cross-validation gives an excellent estimate of the generalization error. It even gives a good estimate on small training set. The k-fold cross-validation estimate also has a very good correlation with the test error. Hence, in this study, we take the k-fold cross-validation error into consideration.

Let kCVMR denotes the mean of the k-folds’ misclassification rate (MR). With the use of cross-validation, kCVMR, Eq. (12) is deserved to be a criterion for generalization ability of a model.

\[
kCVMR = \frac{1}{k} \sum_{j=1}^{k} MR_j
\]

where, k is the size of folds. In this study, fivefold cross-validation is conducted, which is widely used and suggested in [8,13,26].

3.4. SVM parameter optimization by ICDF and BBDE

In the proposed method, the exhaustive search of parameters is performed in the priority area indicated by ICDF measure first. Then, BBDE is responsible to search the optimal parameter combination of SVM in the shortened interval of kernel parameter and interval of penalty parameter C.

For a classification problem, a larger kernel parameter search space, \([2^{-L}, 2^{L}]\), is given firstly, where L is a positive integer which controls the size of original search space of kernel parameter \(\gamma\). The search space is discretized as \([2^{-L}, 2^{-L+1}, \ldots, 2^L]\). Then, according to the problem whether is a binary classification problem (BCP) or is a multi-class classification problem (MCP), the search interval of kernel parameter is determined in different way. For BCP, the ICDF values for each discretization value of \(\{2^{-L}, 2^{-L+1}, \ldots, 2^L\}\) are calculated. The discretization value with the largest ICDF is selected as the search center of the kernel parameter. For MCP, assuming that the number of class is n, the MCP can be decomposed into \(n(n-1)/2\) class pair by using of one-against-one strategy. For each class pair, the discretization value with the largest ICDF can be found. Then, a small and effective search interval of kernel parameter covering the optimal kernel parameter of each class pair is determined.

The detailed process of the proposed method is presented Fig. 4. It mainly contains the following steps.

Step 1: Determine whether the problem is a binary classification problem or a multi-class problem.
Step 2: If the problem is a binary classification problem, ICDF values of \(\{2^{-L}, 2^{-L+1}, \ldots, 2^L\}\) are calculated to find the search center \(\gamma=2^l, j=-L, (-L+1), \ldots, L\), whose corresponding ICDF
Determine the search interval of $\gamma$ as $[\gamma_{\min}, \gamma_{\max}]$. Then, sort all $\gamma_i$ to find the minimum $\gamma_{\min}$ and the maximum $\gamma_{\max}$. At last, the search interval of $\gamma$ for multi-class problem is determined as $[\gamma_{\min}, \gamma_{\max}]$.

Step 3: The population of $(\gamma, C)$ is randomly initialized in the determined search interval of $\gamma$ and $[C_{\min}, C_{\max}]$.

Step 4: SVM is trained with each parameter combination $(\gamma, C)$, and its corresponding ICDF is calculated.

Step 5: For each $(\gamma, C)$, according to Eq. (10) move to the next generation. Go to step 4.

Steps (4)–(5) are repeated until the termination condition is satisfied.

As an example, Fig. 5 shows the ICDFs of Iris data problem. We use the RBF kernel and set $\gamma \in \{2^{-30}, 2^{-20}, \ldots, 2^{-10}\}$, $C \in \{2^{-7}, 2^7\}$. In Fig. 5, $\gamma_{\min} = 2^{-3} = 0.125$, $\gamma_{\max} = 2^{-1} = 0.5$, so the new search interval of $\gamma$ is $[0.125, 0.5]$, which is much smaller than that of the traditional method.

We randomly select 40% samples from each class as training samples and the rest are testing samples. The results are that: best $C = 79.8$, best $\gamma = 0.2053$, and the classification accuracy is 98.89% which is satisfactory.

4. Experiments

In this study, we conduct two groups of experiments to evaluate the proposed method. In the first group of experiments, the proposed algorithm is compared with GS–SVM which uses GS to search the optimal parameters, BBDE–SVM which only uses BBDE to search the optimal parameters, and ICDF–GS–SVM which uses ICDF to determine the range of the kernel parameter and uses GS to optimize the parameter values. In the second group of experiments the proposed algorithm is compared with some algorithms reported in literature [14,25,36].

4.1. Experiment I

The first group of experiments aim to demonstrate the important role of ICDF and BBDE in the proposed hybrid method, and to illustrate the performance of the proposed method in dealing with multiclass problems. Hence, the proposed method is compared with GS–SVM, BBDE–SVM and ICDF–GS–SVM.

The experiments are conducted on 8 datasets that are multiclass problems. The basic information of these datasets is shown in Table 1, which presents, for each dataset, the name of the datasets, the number of classes (# classes), the number of features (# features) and the number of instances (# instances). The datasets glass, wine and iris were extracted from the UCI repository [34], while the datasets dna, letter and satimage were originally from the Statlog collection [37]. The datasets svmguide 2 and svmguide 4.

![ICDF](image)

**Fig. 5.** The ICDFs for Iris data.

<table>
<thead>
<tr>
<th>Data set</th>
<th># of classes</th>
<th># of features</th>
<th># of instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNA</td>
<td>3</td>
<td>180</td>
<td>3186</td>
</tr>
<tr>
<td>Letter</td>
<td>26</td>
<td>16</td>
<td>20,000</td>
</tr>
<tr>
<td>Satimage</td>
<td>6</td>
<td>36</td>
<td>6435</td>
</tr>
<tr>
<td>Glass</td>
<td>6</td>
<td>9</td>
<td>214</td>
</tr>
<tr>
<td>Wine</td>
<td>3</td>
<td>13</td>
<td>178</td>
</tr>
<tr>
<td>Iris</td>
<td>3</td>
<td>4</td>
<td>150</td>
</tr>
<tr>
<td>svmguide2</td>
<td>3</td>
<td>20</td>
<td>391</td>
</tr>
<tr>
<td>svmguide4</td>
<td>6</td>
<td>10</td>
<td>612</td>
</tr>
</tbody>
</table>
4 can be found in the LibSVM tool webpage [8]. The datasets dna, letter and satimage contain training, validation and testing data. Table 2 shows the number of training, validation and testing for these three datasets. The dataset svmguide4 contains 300 training samples and 312 testing samples. For the other datasets in the experiments each dataset is divided in a proportion of 70% for training and 30% for testing. For the datasets dna, letter and satimage that contain validation set, the validation error is used in the training phase to validate the generalization ability of each SVM model. For the rest datasets that do not contain validation set, the mean of the five folds’ misclassification rate which has been described in Section 3.3 is selected as the fitness function for parameter optimization (Table 3).

The computing platform is a PC with the following features: Intel Pentium IV 3.0 GHz CPU, 2 GB RAM, a Windows 7 operating system and the Visual Studio 2010 and MATLAB R2010a development environment. Hsu et al. [38] point out that normalization prevents the domination of attributes in higher numerical ranges over those in lower ranges. Hence, all datasets were normalized with zero mean and unit variance and were scaled to be in $[-1, 1]$. All the experiments are done by the help of LibSVM [8]. We use RBF kernel and set $\gamma \in [2^{-20}, 2^{-19}, \ldots, 2^{0}]$, $C \in [2^{-7}, 2^{-6}, \ldots, 2^{7}]$, hence a grid of 615 ($41 \times 15$) positions are considered.

(2) BBDE-SVM: Population size $NP=15$, probability of reproduction $CR=0.9$. When the number of iterations exceeds $G_m=200$, or when the fitness value is not change for 20 times iterations the algorithm is terminated.

(3) ICDF–GS–SVM: The candidate values of $\gamma$ are $\{2^{-20}, 2^{-19}, \ldots, 2^{0}\}$ for ICDF. The possible values of $C$ are $\{2^{-7}, 2^{-6}, \ldots, 2^{7}\}$. The GS search step is set as 1.

(4) The proposed ICDF–BBDE–SVM: The candidate values of $\gamma$ are $\{2^{-20}, 2^{-19}, \ldots, 2^{0}\}$. The search interval of $C$ is $[2^{-7}, 2^{7}]$. For BBDE, The rest parameters are set the same as in (2).

The testing accuracy and training time of these four methods are visualized in Table 4. For the proposed method and the ICDF–GS–SVM method, the training time is the sum of the time for calculating the ICDF values for all kernel parameters and the time for training SVMs.

As for training time, it can be observed from Table 4 that the training time of the proposed method and ICDF–GS–SVM is much shorter than that of BBDE–SVM and GS–SVM, as the search interval of $\gamma$ determined by ICDF in the proposed method and ICDF–GS–SVM is much smaller than that of BBDE–SVM and GS–SVM. The shortened search intervals of $\gamma$ is shown in Table 5. The computational complexity of calculating ICDF is $O(2^{19})$ [30] while that of training SVM is experimentally shown to be $O(4^l)$ [39] in worst case, where $l$ is sample size. Hence, using ICDF to determine the appropriate $\gamma$ range to parameter search is more time saving than directly searching parameters in a large range of $\gamma$.

It also can be seen that, for most of the experiments, the training time of BBDE–SVM is a bit longer than that of GS–SVM and the training time of the proposed method is a bit longer than that of ICDF–GS–SVM. It is because that BBDE searches the optimal parameter combination from the population evolved generation

### Table 2

<table>
<thead>
<tr>
<th>Datasets</th>
<th># of training</th>
<th># of validation</th>
<th># of testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dna</td>
<td>1400</td>
<td>600</td>
<td>1186</td>
</tr>
<tr>
<td>Letter</td>
<td>10,500</td>
<td>4500</td>
<td>5000</td>
</tr>
<tr>
<td>Satimage</td>
<td>3104</td>
<td>1331</td>
<td>2000</td>
</tr>
</tbody>
</table>

### Table 3

The search intervals of $\gamma$ for the datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Dna</th>
<th>Letter</th>
<th>Satimage</th>
<th>Glass</th>
<th>Wine</th>
<th>Iris</th>
<th>Svmguide2</th>
<th>Svmguide4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search interval of $\gamma$</td>
<td>$[2^{-7}, 2^{-6}]$</td>
<td>$[2^{-2}, 2^{1}]$</td>
<td>$[2^{-4}, 2^{-1}]$</td>
<td>$[2^{-3}, 2^{1}]$</td>
<td>$[2^{-3}, 2^{-1}]$</td>
<td>$[2^{-7}, 2^{1}]$</td>
<td>$[2^{-3}, 2^{1}]$</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4

Comparison between the GS–SVM, BBDE–SVM, ICDF–GS–SVM and the proposed method.

<table>
<thead>
<tr>
<th>Data set</th>
<th>GS–SVM</th>
<th>BBDE–SVM</th>
<th>ICDF–GS–SVM</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing accuracy (%)</td>
<td>Training time (s)</td>
<td>Testing accuracy (%)</td>
<td>Training time (s)</td>
<td>Testing accuracy (%)</td>
</tr>
<tr>
<td>Dna</td>
<td>94.62</td>
<td>4167.83</td>
<td>94.72</td>
<td>4353.67</td>
</tr>
<tr>
<td>Letter</td>
<td>95.56</td>
<td>336786.87</td>
<td>96.23</td>
<td>35688.53</td>
</tr>
<tr>
<td>Satimage</td>
<td>89.76</td>
<td>120076.3</td>
<td>89.35</td>
<td>13787.28</td>
</tr>
<tr>
<td>Class</td>
<td>69.23</td>
<td>28.97</td>
<td>70.77</td>
<td>68.35</td>
</tr>
<tr>
<td>Wine</td>
<td>98.08</td>
<td>3.96</td>
<td>96.15</td>
<td>7.71</td>
</tr>
<tr>
<td>Iris</td>
<td>100</td>
<td>2.35</td>
<td>100</td>
<td>4.78</td>
</tr>
<tr>
<td>Svmguide2</td>
<td>83.51</td>
<td>63.82</td>
<td>85.78</td>
<td>83.28</td>
</tr>
<tr>
<td>Svmguide4</td>
<td>88.33</td>
<td>116.45</td>
<td>89.10</td>
<td>126.87</td>
</tr>
</tbody>
</table>
by generation, which requires re-training SVMs many times. In current experiment configuration, the population size $N_P = 15$, it will require $5^*15 = 75$ SVMs to be trained in one generation when adopt fivefold cross validation. And when the search intervals are large, the algorithm will need very many generations to reach the termination condition.

As for testing accuracy, it can be seen that the proposed method yields better testing rate in most of the datasets. The exception occurred for satimage and iris, where in satimage dataset the ICFD–GS–SVM method and the proposed method achieve the same testing rate which is higher than that of the other two methods, and in iris dataset, these four methods produce the same results. Furthermore, it also can be seen that, for most of the experiments, the testing accuracy of BBDE–SVM is a bit higher than that of GS–SVM and the testing accuracy of the proposed method is a bit higher than that of ICFD–GS–SVM. The reason is that the BBDE searches the optimal $(\gamma, C)$ in the continuous intervals of $\gamma$ and $C$, while the GS searches the parameter combination in the discrete grid. BBDE has more elaborate search ability than GS, which can avoid omitting the optimal parameters.

From the aforementioned combined results it can be conclude that the proposed method outperforms the other three methods in SVM parameter optimization.

4.2. Experiment II

To further evaluate the performance of the proposed method, we test it on five commonly used benchmark datasets from the UCI benchmark repository [34]: breast cancer, diabetes, heart, thyroid and titanic. These datasets, having been preprocessed by [40], consist of 100 random groups with training and testing sets (about 60%: 40%) describing the binary classification problem. The dataset in each group has already been normalized to zero mean and unit standard deviation. For the dataset in each group, the training set is used to select the optimal parameters in SVM based on fivefold cross-validation and establish the classifier, and then the test set is used to assess the classifier with the optimal parameters. Table 5 summarizes the general information of these datasets, including names of the datasets, number of groups (# of groups), number of training samples in each group (# of training samples), number of testing samples in each group (# of testing samples), and number of attributes (# of attributes).

We follow the same experimental setup as in Refs. [14,25]. Firstly, on each of the first five groups of every benchmark dataset, the parameter combination $(\gamma, C)$ is optimized by the proposed method. Then, the final parameters are computed as the median of obtained five estimations. At last, the mean error rate and standard deviation of SVMs with the final parameter combination on all 100 groups of each dataset are taken as the results of the corresponding algorithm. The results are compared with those of literature [14,25,36].

In literature [14], a fast method based on an approximation of the gradient of the empirical error along with incremental learning is used to tune the parameters of SVM. In [25], PSO integrated pattern search (PS) and a novel probabilistic selection strategy (PSO–PS) is used to optimize the SVM parameters. Literature [36] presents a nested Variable Neighborhood Search (VNS) algorithm to tune the parameters of SVM. In the nested VNS algorithm, a complex model is considered as nested within another model. Thus it is applicable to parameter tuning problems for different kernel models, especially for multiple kernel learning model (MKL). However, in this paper, we do not consider the MKL model, a basic VNS algorithm is sufficient to tune the parameters for SVM with RBF kernel model. In particular, the authors of literature [36], only consider the first of those 100 partitions to compute the reported testing error.

The comparison results are shown in Table 5 and Fig. 5, which show the mean testing error rates (%) and the standard deviations (%). obtained by literature [14,25] and testing error obtained by literature [36]. For convenience of comparison, the best results for every dataset are bolded. The first column shows the name of the datasets. The second column reports the results obtained by the method proposed in [14]. The results obtained by PSO–PS [25] are reported in the third column. The fourth column shows the results reported in literature [36]. The last column reports the results of the proposed method. It can be seen that the proposed method yields better results on three datasets: breast cancer, thyroid and titanic, than the other three methods. On datasets: diabetes and heart, PSO–PS yields better results. In general, our results are similar to those obtained by literature [25], and outperform the results obtained by [14,36] (Fig. 6 and Table 6).

![Graph](image-url)

**Table 6** Test error found by different algorithms for selecting the SVM parameters (%).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast cancer</td>
<td>25.48 ± 4.38</td>
<td>24.37 ± 4.61</td>
<td>28.57</td>
<td>24.32 ± 4.20</td>
</tr>
<tr>
<td>Diabetes</td>
<td>23.41 ± 1.68</td>
<td>23.17 ± 1.35</td>
<td>24.67</td>
<td>23.31 ± 1.33</td>
</tr>
<tr>
<td>Heart</td>
<td>15.96 ± 3.13</td>
<td>15.95 ± 3.11</td>
<td>20.00</td>
<td>16.15 ± 2.70</td>
</tr>
<tr>
<td>Thyroid</td>
<td>4.70 ± 2.07</td>
<td>3.4 ± 2.07</td>
<td>5.33</td>
<td>3.30 ± 2.02</td>
</tr>
<tr>
<td>Titanic</td>
<td>22.90 ± 1.16</td>
<td>21.58 ± 1.06</td>
<td>22.92</td>
<td>21.48 ± 0.54</td>
</tr>
</tbody>
</table>

Please cite this article as: X. Zhang, et al., Support vector machine with parameter optimization by a novel hybrid method and its application to fault diagnosis, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.08.010.
Following the same experimental configuration, both the authors of literature [14,25] have done the comparison study to illustrate the outstanding performance of their method. In [14], the authors compared their work with the methods proposed in literatures [11,40]. They proved that their results are similar to those obtained by literature [11] or [40]. In [25] the authors compared their method with the methods proposed in [15,40,41]. They concluded that their method yields higher and more robust performance than others. Thus, we have reason to believe that the proposed method also outperforms those in literature [11,15,40,41]. Therefore, the proposed method can be a promising alternative for SVMs parameters optimization.

5. Application in fault diagnosis for rolling element bearings

Rolling element bearings are the interfaces between the stationary and the rotating part of the machines. Faults occurring in the bearings may lead to fatal breakdowns of machines and can drive to unacceptably long time maintenance stops, which will result in large personal casualties and economical loss. Therefore, it is significant to detect fast, accurately and easily the existence and severity of the faults in the bearings.

The use of vibration signals is quite common in the field of condition monitoring and diagnostics of rotating machinery and damage detection [42–48]. It is possible to obtain vital diagnosis information from the vibration signals by using some signal processing techniques. Then, feature extraction and selection are undertaken. At last, the pattern recognition methods are used to diagnose the faults. In this section, the proposed method is applied in FD for rolling element bearings.

5.1. Data collection

Data collection: the data used in this paper comes from Case Western Reserve University (CWRU) bearing data center [49]. The bearings are installed in a motor driven mechanical system, as shown in Fig. 7. The bearings used in this work are deep groove ball bearings manufactured by SKF. Three faults, outer race fault, inner race fault and ball fault, are introduced into the drive-end bearing of the motor. The defect sizes (diameter, depth) of these three faults are the same: 0.007 in or 0.021 in. Each bearing was tested under the four different loads, 0, 1, 2 and 3 hp. The bearing data set was obtained from the experimental system under the four different operating conditions: (1) normal condition; (2) with inner race fault; (3) with outer race fault; and (4) with ball fault. Data was collected at 12,000 samples/s and 48,000 samples/s.

In the current study, we use the datasets that collected at 12,000 samples/second under 2 hp load. The defect sizes are 0.007 or 0.021 in. Each original signal is divided into 100 signals, and the length of the signal after divided is 4096 data points. Consequently, 600 samples covering 6 types’ faults are produced. Each class faults is divided in a proportion of 70% for training and 30% for testing, as list in Table 7.

### Table 7

<table>
<thead>
<tr>
<th>The number of fault samples</th>
<th>For training</th>
<th>For testing</th>
<th>Defect size (in)</th>
<th>Operating condition</th>
<th>Label of classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>70</td>
<td>30</td>
<td>0.007</td>
<td>Outer race</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>70</td>
<td>30</td>
<td>0.007</td>
<td>Inner race</td>
<td>2</td>
</tr>
<tr>
<td>100</td>
<td>70</td>
<td>30</td>
<td>0.021</td>
<td>Outer race</td>
<td>4</td>
</tr>
<tr>
<td>100</td>
<td>70</td>
<td>30</td>
<td>0.007</td>
<td>Inner race</td>
<td>5</td>
</tr>
<tr>
<td>100</td>
<td>70</td>
<td>30</td>
<td>0.021</td>
<td>Ball</td>
<td>6</td>
</tr>
</tbody>
</table>

5.2. Feature extraction

Two time-domain features, Eqs. (13) and (14), and two frequency-spectrum features, Eqs. (15) and (16), are selected for FD of the rolling element bearings. These four features provide dominant fault-related information of vibration signals [48].

\[
T_1 = \frac{\sum_{n=1}^{N} x(n)}{N} \quad (13)
\]

\[
T_2 = \sqrt{\frac{1}{N-1} \sum_{n=1}^{N} (x(n) - T_1)^2} \quad (14)
\]

where \(x(n)\) is a signal series, \(n=1,2,…,N\), and \(N\) is the number of data points

\[
F_1 = \frac{\sum_{k} s^2(k)}{K} \quad (15)
\]

\[
F_2 = \frac{\sum_{k} (s(k) - F_1)^2}{K-1} \quad (16)
\]

where \(s(k)\) is a spectrum for \(k=1,2,…,K\), and \(K\) is the number of spectrum lines.

5.3. Fault diagnosis by proposed method

All of the data are scaled to be in [0, 1]. By using the first 3 attributes, three-dimensional data distribution of the fault samples is shown in Fig. 7. Where b007, b021, ir007, ir021, or007, or021 denote ball fault with 0.007 in, ball fault with 0.021 in, inner race fault 0.007 in, outer race fault 0.007 in, inner race fault 0.021 in, and outer race fault 0.021 in, respectively.

**Fig. 7.** Schematic diagram of the experimental setup.

**Fig. 8.** The data distribution of 6 classes’ faults of rolling element bearings.
fault with 0.007 in, inner race fault with 0.021 in, outer race fault with 0.007 in and outer race fault with 0.021 in, respectively (Fig. 8).

The experiments configuration is the same as Section 4.1. Table 8 shows the experimental results. For comparison, the GS-DE and DE-SVM are also utilized to solve the same six-class classification problem. DE–SVM uses DE to search the optimal parameters in the parameter space. The parameters of DE are set as follows: population size \( Np=10 \), probability of reproduction \( Cr=0.9 \), scaling factor \( F=0.5 \), as suggested in Storn and Price [33]. When the number of iterations exceeds \( Gm=200 \), or when the fitness value is not change for 20 iterations the algorithm is terminated. The experiment is repeated for 30 times. Table 8 shows the results. The value of \( \gamma \) and \( C \) are the optimal parameters in one time experiment. The mean diagnosis accuracy (%) and the standard deviations (%) are reported in the fourth column of Table 8, while the mean cost time (s) and the standard deviations are reported in the fifth column.

From Table 8 we can see that the proposed approach has the highest average diagnosis accuracy, 99.79%, with the average cost time, 19.42 s. The result is quite a remarkable achievement from an engineering point of view. DE–SVM also has high diagnosis accuracy but it is time consuming. As for the standard deviations of diagnosis accuracy and cost time, the proposed method also outperforms DE–SVM. It is obvious that the proposed algorithm has stronger stability than that of DE–SVM, and the diagnosis ability of the proposed algorithm is significantly stronger than that of GS–SVM. These combined results indicate that the proposed ICDF–BBDE–SVM is a powerful FD algorithm for rolling element bearings.

6. Conclusions

In this study, a new hybrid method, ICDF combined with BBDE, is used to optimize the parameters of SVM. The ICDF serves as a promising method for determining the effective search interval of kernel parameter \( \gamma \). The BBDE, which is a new, almost parameter-free optimization algorithm, is adopted to search the optimal parameter combination of SVM. The proposed method is applied in some benchmark datasets as well as fault diagnosis of rolling element bearings. Comparing to some methods both mentioned in this paper and published in other literature, the results show that the proposed method is both effective and computationally efficient in searching the optimal parameters for multiclass SVM, and thus is feasible for fault diagnosis of rolling element bearings. However, currently, the penalty parameter \( C \) is not incorporated into the proposed strategies, or else the performance might be further improved. And in the proposed method, the fitness value is the fivefold cross validation testing accuracy which will train five SVMs for one parameter combination, which maybe time consuming. Some effective and efficient performance measures for SVM should be incorporated into the proposed method. Future research will be focused on the above two points.

Acknowledgment

The work described in this paper is supported by the National Natural Science Foundation of China under Grant No. 51409095. In addition, thanks are due to the free download of the rolling element bearing fault data sets provided by the Case Western Reserve University, Rockwell Science Office of Naval Research and CVX. The authors would also like to thank the reviewers for their comments and suggestions that helped in revising the paper to its present form.

References


Xiaoyuan Zhang is a Lecturer at the College of Electrical Engineering, Henan University of Technology, PR China. He received his B.Sc. and M.Sc. in Thermal Energy and Power Engineering, from North China University of Water Conservancy and Hydropower, PR China, in 2004, 2007, respectively, and received his Ph.D. in Hydraulic and Hydro-Power Engineering from Huazhong University of Science and Technology, PR China, in 2012. His research interests are fault diagnosis for rotating machinery, signal processing and pattern recognition.