Information Sciences 370–371 (2016) 92–102

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Tuning kernel parameters for SVM based on expected square distance ratio

Shen Yin*, Jiapeng Yin

Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Heilongjiang 150001, China

ARTICLE INFO

Article history:
Received 29 July 2015
Revised 14 July 2016
Accepted 20 July 2016
Available online 27 July 2016

Keywords:
Classification
Class separability
Kernel parameter
SVM

ABSTRACT

The performance of a support vector machine (SVM) depends highly on the selection of the kernel function type and relevant parameters. To choose the kernel parameters properly, methods analyzing the class separability have been widely adopted for their efficiency compared with other methods, such as the popular grid search algorithm. This paper proposes a novel index called the Expected Square Distance Ratio (ESDR), which can serve as a better class separability criterion than the existing ones. Experiments on real-world datasets show that, compared with common kernel parameter selection methods that utilize the between-class separation, the variations in ESDR with respect to the kernel parameter are much more in line with those of the classification accuracy, leading to better kernel parameters. Moreover, ESDR takes the exact data distribution into account and can thus be used to study the model selection problem of an SVM for certain forms of data distribution. As an example, we employ the ESDR to analyze the selection of RBF (Radial Basis Function) kernel parameters for Gaussian data classification.

1. Introduction

Since its development in the 1990s, a support vector machine (SVM) [32] has been widely used in the fields of pattern recognition [4,28,38,39,45] and regression estimation [10,17,33,41,44]. The performance of an SVM largely depends on the kernel function it adopts. Therefore, it is of great importance to choose the kernel type and set the corresponding parameters appropriately. However, to date, there have been no optimal methods that can lead to the correct kernel and its hyperparameters.

Despite this fact, there are still some effective approaches to selecting the proper model for an SVM, among which the grid search (GS) algorithm [16] is the most straightforward. When operating along with a cross-validation [15], GS can be quite effective and stable. However, GS suffers from a heavy computational burden because the SVM model has to be rebuilt for all combinations of parameters. To avoid searching the entire parameter space, some optimization algorithms, such as the genetic algorithm (GA) [7,26], particle swarm optimization (PSO) [31,43], simulated annealing (SA) technique [18,25], fruit fly optimization algorithm (FOA) [29], gravitational search algorithm (GSA) [24], social emotional optimization algorithm (SEOA) [46], firefly algorithm [5], and artificial chemical reaction optimization algorithm (ACROA) [1], can be applied to the SVM parameter tuning process. However, before the termination of such methods, the entire population has to be updated for many generations, and thus, they remain time-consuming processes. To solve this problem analytically, Chapelle et al.

[*] Corresponding author. Tel.: +8645186402350.
E-mail addresses: shen.yin@hit.edu.cn, shen.yin@uni-due.de (S. Yin), yjp.aaron@gmail.com (J. Yin).

http://dx.doi.org/10.1016/j.ins.2016.07.047
0020-0255/© 2016 Elsevier Inc. All rights reserved.

[6] first proposed a generalization error estimation called a radius-margin bound, and used a gradient descent algorithm to arrive at good parameters. However, this method is only applicable to the L2-SVM and needs to solve an extra quadratic optimization problem, which makes it very time-consuming. Under Bayesian interpretations of an SVM, Gold et al. [12] chose the parameters by maximizing the available evidence. Gomes et al. [13] bonded the initial parameter pair obtained from meta-learning with search techniques such as a PSO, and employed a hybrid algorithm to select the hyperparameters. Under the assumption of a Gaussian distribution, Wang et al. [37] determined the optimal super-parameter of a Gaussian kernel that leads to sufficient support vectors before eliminating the outliers. Utilizing the distances from the samples to the enclosing surfaces, [42] derived an optimization problem to select a Gaussian kernel parameter for a one-class SVM.

To better take the data distribution into consideration and tune the SVM from the perspective of the geometry, the issue of class separability is introduced. Among all of the class separability criteria, scatter-matrix based measures [8] are extensively adopted for their simplicity. As a commonly used scatter-matrix based measure that can be interpreted in terms of a Fisher Discriminant Analysis [36], criterion J_4 has been successfully applied to solve the model selection problem of an SVM [34]. In addition, Sun et al. [30] proposed analyzing the distance between two classes (DBTC) to obtain the desired kernel parameters. Wu and Wang [40] presented the inter-cluster distance in the feature space, which is equivalent to DBTC. In this paper, we put forward a new index called the Expected Square Distance Ratio (ESDR), which can quantify the class separability well. Compared with other criterions such as DBTC, the ESDR has a clearer intuitive meaning and is more accordant with the accuracy of SVM classification when employed to select the kernel parameters for an SVM, illustrating that it reflects the geometric structures of the feature space corresponding to certain kernels. Moreover, explicitly taking the data distribution into account, the ESDR can serve as a powerful tool for the study of SVM model selection for certain forms of data distribution, such as a Gaussian distribution.

The penalty coefficient C also has a significant influence on the performance of an SVM. Methods such as a grid search technique and radius-margin bound incorporate C and kernel parameters into a unified framework [20]. However, the kernel parameters determine the geometry of the feature space, whereas C weighs the margin maximization and error minimization, and has no intuitive meaning in terms of geometry. Thus, the class separability is usually applied to determine the kernel parameters rather than C for an SVM. In our proposed method, the ESDR is employed first to select the optimal kernel parameters, and the penalty coefficient C is then determined through a cross-validation and grid search technique.

The rest of this paper is organized as follows. Section 2 reviews the theoretical background of an SVM. Section 3 analyzes the properties of the ESDR and compares them with other criteria, and applies the ESDR for the parameter tuning of an RBF kernel on Gaussian data. The results of several experiments conducted on some real-world datasets and Gaussian data are provided in Section 4. Finally, Section 5 provides some concluding remarks regarding this research.

2. Related work

SVMs have been extensively used in many fields with a good performance level [4,10,17,28,33,38,39,41,44,45]. Before the SVM emerged as an outstanding machine learning method, classical learning approaches such as neural networks [14] merely follow the empirical risk minimization (ERM) rule, the key point of which is minimizing the training error, leading to the problem of overfitting. Based on the Vapnik–Chervonenkis theory (VC-theory), an SVM follows the structural risk minimization (SRM) rule, which not only minimizes the training error but also restricts the complexity of the learning machine, thus improving the generalization abilities [32]. Owing to its well-established mathematical foundations, an SVM has been successfully applied to numerous learning tasks under various conditions.

The original SVM was proposed for binary classification. Considering the training set $(\boldsymbol{x}_i, y_i)$, $i = 1, 2, \ldots, n$, $\boldsymbol{x}_i \in \boldsymbol{R}^d$, where $\boldsymbol{x}_i$ is the training data vector of d-dimensions, $y_i \in \{-1, +1\}$ is the corresponding class label, and n is the size of the training set, the SVM constructs a separating hyperplane that results in zero training errors (assuming that the training set is linearly separable) and a maximal margin. The margin refers to the maximal width of the slab parallel to the hyperplane with no data points inside and where the optimal separating hyperplane is directly in the middle of the slab. For visualization, see Fig. 1 for the specific case of a two-dimensional space.

To obtain the optimal separating hyperplane $< \boldsymbol{\omega}, \boldsymbol{x} > +b = 0$, an optimization problem needs to be solved

$$\begin{aligned} &\min_{\boldsymbol{\omega}, b} \quad \frac{1}{2}\|\boldsymbol{\omega}\|^2 \\ &s.t. \quad y_i(< \boldsymbol{\omega}, \boldsymbol{x}_i > +b) \geq 1. \end{aligned} \tag{1}$$

For the sake of generalization, the separating hyperplane is occasionally allowed to not separate all of the training data. In this case, a soft margin is used, and the optimization problem (1) is transformed into

$$\begin{aligned} &\min_{\boldsymbol{w}, b, s} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i s_i \\ &s.t. \quad y_i(< \boldsymbol{w}, \boldsymbol{x}_i > +b) \geq 1 - s_i, s_i > 0 \end{aligned} \tag{2}$$

where s_i is called a slack variable, and C is the penalty coefficient.

In a real-world task, however, data are rarely linearly separable. To solve the problem of nonlinearity, the original data should be projected through nonlinear mapping $\Phi(\boldsymbol{x})$ onto a new space, usually with higher dimensions, where the data points are linearly separable.

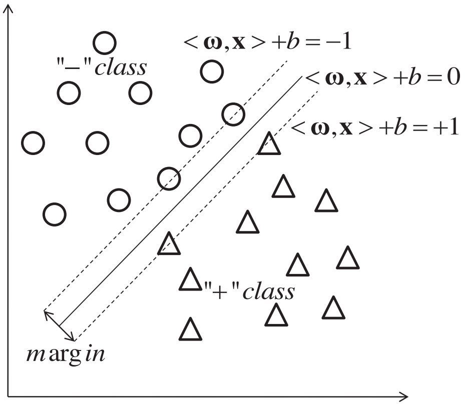

Fig. 1. Using SVM for classification in 2-D space.

Both (1) and (2) can be solved in dual form using the Lagrange method. In a nonlinear case, the dual form is

$$\begin{aligned} &\min_{\alpha} \quad \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j < \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) >_F \\ &s.t. \quad 0 \le \alpha_i \le C, \sum_i \alpha_i y_i = 0 \end{aligned} \tag{3}$$

where α_i is the Lagrange multiplier. The final decision function is

$$class(\boldsymbol{x}) = sign\left(\sum_i \alpha_i y_i < \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}) >_F + b\right). \tag{4}$$

During the entire construction process of an SVM, we only need to know the inner product in the new feature space, rather than the explicit form of nonlinear mapping $\Phi(\boldsymbol{x})$. The inner product $K(\boldsymbol{x}, \boldsymbol{y}) =< \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) >_F$, is defined as a kernel function. The introduction of a kernel function enriches a number of traditional methods and leads to more advanced methods such as a kernel principal component analysis (KPCA) [22], kernel clustering [11,27], kernel fisher discriminant analysis (KFDA) [2,9], and kernel partial least squares (KPLS) [35]. Kernel functions make an SVM applicable to a nonlinear case, thus considerably broadening its applicable fields. Different kernels and their parameter configuration correspond to different nonlinear problem-dealing behaviors. The frequently used kernel functions are as follows:

1. Polynomial

$$K(\boldsymbol{x}, \boldsymbol{y}) = (< \boldsymbol{x}, \boldsymbol{y} > + p)^q, q \in N. \tag{5}$$

2. Radial basis function (RBF)

$$K(\boldsymbol{x}, \boldsymbol{y}) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}}, \sigma > 0. \tag{6}$$

3. Sigmoid

$$K(\boldsymbol{x}, \boldsymbol{y}) = tanh(\gamma < \boldsymbol{x}, \boldsymbol{y} > + c), \gamma > 0, c < 0. \tag{7}$$

In this study, we focus solely on the parameter selection of an RBF kernel, which is always the first choice for the application of an SVM [15]. In addition, we only focus on binary classification problems because a multi-class SVM is a combination of multiple binary SVMs [16].

3. Expected square distance ratio (ESDR)

3.1. Definition of ESDR

Consider a set of points of two classes, $\boldsymbol{x}_{1,i}$ for one class, and $\boldsymbol{x}_{2,j}$ for the other, where $i = 1, 2, \ldots, n_1$, $j = 1, 2, \ldots, n_2$ and $n_1 + n_2 = n$. We conduct a sampling on this dataset twice with a replacement, one point at a time. Here, we only consider the Euclidean distance of these two points, which we note as d_1 if the two points are from different classes and d_2 otherwise. Intuitively, the more d_1 tends to be relative to d_2, the farther apart the two classes are, and the easier they are to be separated. Based on this simple rule, we propose the ESDR as

$$\begin{aligned} ESDR &= \frac{E(d_1^2)}{E(d_2^2)} \\ &= \frac{E(d^2(\boldsymbol{x}_{1,i}, \boldsymbol{x}_{2,j}))}{\frac{n_1}{n}E(d^2(\boldsymbol{x}_{1,i_1}, \boldsymbol{x}_{1,i_2})) + \frac{n_2}{n}E(d^2(\boldsymbol{x}_{2,j_1}, \boldsymbol{x}_{2,j_2}))}, \end{aligned} \tag{8}$$

where E denotes the expected value and d denotes Euclidean distance.

In the feature space induced by kernel $K(\boldsymbol{x}, \boldsymbol{y})$, the distances between points can be calculated in the form of an L^2 norm:

$$\begin{aligned} d_F^2(\Phi(\boldsymbol{x}), \Phi(\boldsymbol{y})) &= <\Phi(\boldsymbol{x}) - \Phi(\boldsymbol{y}), \Phi(\boldsymbol{x}) - \Phi(\boldsymbol{y})) >_F \\ &= <\Phi(\boldsymbol{x}), \Phi(\boldsymbol{x})>_F - <\Phi(\boldsymbol{x}), \Phi(\boldsymbol{y})>_F - \\ &\quad <\Phi(\boldsymbol{y}), \Phi(\boldsymbol{x})>_F + <\Phi(\boldsymbol{y}), \Phi(\boldsymbol{y})>_F \\ &= K(\boldsymbol{x}, \boldsymbol{x}) - 2K(\boldsymbol{x}, \boldsymbol{y}) + K(\boldsymbol{y}, \boldsymbol{y}). \end{aligned} \tag{9}$$

The form of the ESDR in the feature space can then be easily expressed.

Supposing that the points of these two classes are drawn to two distributions, respectively, and that the probability density function (PDF) for each class is known, the exact value of the ESDR can then be obtained provided that the integration involved is computable.

In practice, however, the above-mentioned PDF is usually unknown. Even if it is known, the integration may not exist, for instance, if it has a Cauchy distribution [23]. Thus, we introduce an approximate form of ESDR based on the following approximations:

$$E(d_F^2(\Phi(\boldsymbol{x}_{1,i}), \Phi(\boldsymbol{x}_{2,j}))) = \frac{1}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} d_F^2(\Phi(\boldsymbol{x}_{1,i}), \Phi(\boldsymbol{x}_{2,j})), \tag{10}$$

$$E(d_F^2(\Phi(\boldsymbol{x}_{1,i_1}), \Phi(\boldsymbol{x}_{1,i_2}))) = \frac{1}{n_1^2}\sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_1} d_F^2(\Phi(\boldsymbol{x}_{1,i_1}), \Phi(\boldsymbol{x}_{1,i_2})), \tag{11}$$

$$E(d_F^2(\Phi(\boldsymbol{x}_{2,j_1}), \Phi(\boldsymbol{x}_{2,j_2}))) = \frac{1}{n_2^2}\sum_{j_1=1}^{n_2}\sum_{j_2=1}^{n_2} d_F^2(\Phi(\boldsymbol{x}_{2,j_1}), \Phi(\boldsymbol{x}_{2,j_2})). \tag{12}$$

3.2. Comparison with other criterions

From part 3.1, we can see that the farther the two classes are from each other, the larger the numerator of the ESDR is; separately, the more compact the two classes are, the smaller the denominator of the ESDR is. Thus, the ESDR can reflect the separability between classes well. To further illustrate the properties of the ESDR, we compared it with other criteria. Herein, we focus on two class separability measures, namely, the distance between two classes (DBTC) [30] and criteria J_4 [8,34], both of which have been successfully applied to tuning the kernel parameters for an SVM.

The DBTC is defined as the square distance between the means of two classes in the feature space. According to (9), the DBTC can be expressed as

$$\begin{aligned} DBTC &= d_F^2(\boldsymbol{m}_1^\Phi, \boldsymbol{m}_2^\Phi) \\ &= d_F^2(\frac{1}{n_1}\sum_{i=1}^{n_1}\Phi(\boldsymbol{x}_{1,i}), \frac{1}{n_2}\sum_{j=1}^{n_2}\Phi(\boldsymbol{x}_{2,j})) \\ &= \frac{1}{n_1^2}\sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_1} k(\boldsymbol{x}_{1,i_1}, \boldsymbol{x}_{1,i_2}) - \frac{2}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} k(\boldsymbol{x}_{1,i}, \boldsymbol{x}_{2,j}) \\ &\quad + \frac{1}{n_2^2}\sum_{j_1=1}^{n_2}\sum_{j_2=1}^{n_2} k(\boldsymbol{x}_{2,j_1}, \boldsymbol{x}_{2,j_2}). \end{aligned} \tag{13}$$

In the feature space, criteria J_4 is defined as

$$J_4 = \frac{tr(\boldsymbol{S}_b^\Phi)}{tr(\boldsymbol{S}_w^\Phi)}, \tag{14}$$

where *tr* denotes the trace of a matrix, and the between-class scatter matrix $\boldsymbol{S}_b^{\Phi}$ and the within-class scatter matrix $\boldsymbol{S}_b^{w}$ in the feature space are expressed as

$$\begin{aligned}
\boldsymbol{S}_b^{\Phi} &= \frac{1}{n}\sum_{c=1}^{2} n_c(\boldsymbol{m}_c^{\Phi}-\boldsymbol{m}^{\Phi})(\boldsymbol{m}_c^{\Phi}-\boldsymbol{m}^{\Phi})^T \\
\boldsymbol{S}_w^{\Phi} &= \frac{1}{n}\sum_{c=1}^{2}\sum_{i=1}^{n_c}(\Phi(\boldsymbol{x}_{c,i})-\boldsymbol{m}_c^{\Phi})(\Phi(\boldsymbol{x}_{c,i})-\boldsymbol{m}_c^{\Phi})^T \\
\boldsymbol{m}_c^{\Phi} &= \frac{1}{n_c}\sum_{i=1}^{n_c}\Phi(\boldsymbol{x}_{c,i}) \\
\boldsymbol{m}^{\Phi} &= \frac{1}{n}\sum_{c=1}^{2} n_c\boldsymbol{m}_c^{\Phi}.
\end{aligned}$$

By comparing (13) and (14), it is easy to find that

$$DBTC = \frac{n^2}{n_1 n_2} tr(\boldsymbol{S}_b^{\Phi}). \tag{15}$$

If the kernel used is normalized, such as RBF kernel (6), (13) can be rewritten as

$$\begin{aligned}
DBTC = &\frac{1}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} d_F^2(\boldsymbol{x}_{1,i},\boldsymbol{x}_{2,j}) \\
&-\frac{1}{2n_1^2}\sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_1} d_F^2(\boldsymbol{x}_{1,i_1},\boldsymbol{x}_{1,i_2}) \\
&-\frac{1}{2n_2^2}\sum_{j_1=1}^{n_2}\sum_{j_2=1}^{n_2} d_F^2(\boldsymbol{x}_{2,j_1},\boldsymbol{x}_{2,j_2}).
\end{aligned} \tag{16}$$

By comparing (16) with an approximate form of the ESDR, we can see that if $ESDR = \frac{a}{\frac{n_1}{n}b+\frac{n_2}{n}c}$, then $DBTC = a - \frac{1}{2}b - \frac{1}{2}c$. With a normalized kernel, the DBTC is indeed able to take account of the within-class distribution. However, as a class separability criterion, the DBTC has its flaws in theory. Because promoting the class separability means making *a* as large as possible relative to *b* and *c*, it is more reasonable to define the class separability measure in a ratio form rather than a subtraction form. When using RBF kernel (6) with a large σ, the distance between any two points in the feature space is approximately zero. Therefore, according to (13), the DBTC will be approximately zero as well. However, an RBF-SVM with a large σ can occasionally still perform well in terms of classification. Although the mapped data are squeezed into a very restricted area, two classes can still be quite apart. Compared with the DBTC, the ESDR can take this factor into consideration, as shown in the experimental results in Section 4. In addition, the ESDR can be applied to any type of kernel, without an extra demand for normalization as with the DBTC.

It is easy to prove that the denominator of the ESDR is twice that of criteria J_4. In addition, ignoring the constant term, the numerator of the ESDR and the numerator of criteria J_4, denote the means of the square distances and the square of the mean distance, respectively, between every pair of points belonging to two classes separately. Thus, from a Euclidean perspective, the ESDR is a better ratio for the consistency between the denominator and numerator. At first glance, the expressions of the ESDR and criteria J_4 are similar. However, the slight difference between the numerators of the ESDR and criteria J_4 can actually lead to great disparities in their properties, as shown in the cases of data 4 and 5 in Section 4. In addition, because J_4 is based on the trace of the matrix, and the physical meaning of the trace is not as direct as that of the distance, we can state that the ESDR is defined in a more direct and proper way. Thus, in some cases, the ESDR can reflect the class separability better than J_4, as shown in the next experimental section. In addition, the ESDR is a normalized criterion with its value being 1 for completely mixed up classes, making the analysis in the feature space clearer and more convenient.

3.3. ESDR for Gaussian data with RBF kernel

The original form of ESDR (8) in the feature space takes in account the data distribution analytically and can thus be employed to study the model selection for an SVM under a specific distribution. To illustrate this important feature, we analyzed the ESDR for Gaussian data with an RBF kernel.

We consider the point set described in part 3.1, and assume that the two classes are subject to two Gaussian distributions, respectively, as $\boldsymbol{x}_{1,\,i} \sim N(\boldsymbol{\mu}_1,\ \boldsymbol{\Sigma}_1)$, $\boldsymbol{x}_{1,\,i} \sim N(\boldsymbol{\mu}_2,\ \boldsymbol{\Sigma}_2)$, where $\boldsymbol{\mu}_1 = (\mu_{11},\mu_{12},\ldots,\mu_{1d})$, $\boldsymbol{\mu}_2 = (\mu_{21},\mu_{22},\ldots,\mu_{2d})$, $\boldsymbol{\Sigma}_1 = diag(\sigma_{11}^2,\sigma_{12}^2,\cdots,\sigma_{1d}^2)$ and $\boldsymbol{\Sigma}_2 = diag(\sigma_{21}^2,\sigma_{22}^2,\cdots,\sigma_{2d}^2)$. This indicates that all of the *d* dimensions are independent of

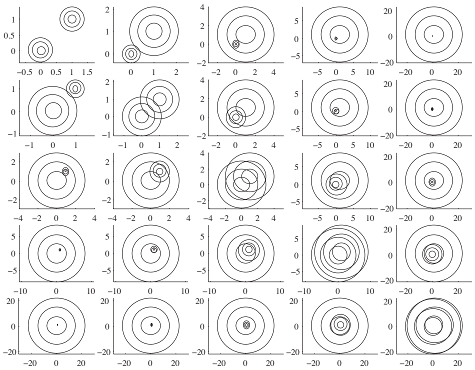

Fig. 2. Visualization of Gaussian data combinations.

one another. Choosing RBF (6) as the kernel, and substituting all of the above into (8), we obtain

$$E(d_1^2) = 2 - 2\prod_{p=1}^{d}\left(\frac{\sigma}{\sqrt{\sigma_{1p}^2+\sigma_{2p}^2+\sigma^2}}\cdot e^{-\frac{(\mu_{1p}-\mu_{2p})^2}{2(\sigma_{1p}^2+\sigma_{2p}^2+\sigma^2)}}\right) \tag{17}$$

$$E(d_2^2) = \frac{2n_1}{n}\left(1-\prod_{p=1}^{d}\frac{\sigma}{\sqrt{2\sigma_{1p}^2+\sigma^2}}\right)+\frac{2n_2}{n}\left(1-\prod_{p=1}^{d}\frac{\sigma}{\sqrt{2\sigma_{2p}^2+\sigma^2}}\right). \tag{18}$$

From (8), (17), and (18), we can see that, as $\sigma \to 0$, $ESDR \to 1$, the two classes are mixed completely, and as $\sigma \to \infty$, according to L'Hôpital's rule, the value of the ESDR approaches that of the ESDR in the original space. This is true for data under any distribution. In fact, for an RBF kernel with $\sigma \to \infty$, the structure of the corresponding feature space is the same as that of the original space. Considering three arbitrary points P_1, P_2, and P_3 in an Euclidean space, we note the distance between P_1 and P_2 as $d(P_1, P_2)$, and that between P_1 and P_3 as $d(P_1, P_3)$. Then, in the feature space induced by the RBF kernel, by applying L'Hôpital's rule, we obtain

$$\lim_{\sigma\to\infty}\frac{d_F(P_1,P_2)}{d_F(P_1,P_3)} = \lim_{\sigma\to\infty}\sqrt{\frac{2-2e^{-\frac{d^2(P_1,P_2)}{2\sigma^2}}}{2-2e^{-\frac{d^2(P_1,P_3)}{2\sigma^2}}}} = \frac{d(P_1,P_2)}{d(P_1,P_3)}, \tag{19}$$

and according to the definition of the ESDR, we can easily draw the above conclusion.

To further illustrate the impact of the RBF kernel parameter σ on the class separability of Gaussian data in the feature space, we consider the Gaussian data described at the beginning of this section with a wider range of parameter combinations. In Fig. 2, for image (i, j) in the ith row and jth column, we suppose $\boldsymbol{\mu}_1 = (0, 0)$, $\boldsymbol{\Sigma}_1 = diag(e^{2(i-3)}, e^{2(i-3)})$, $\boldsymbol{\mu}_2 = (1, 1)$, and $\boldsymbol{\Sigma}_2 = diag(e^{2(j-3)}, e^{2(j-3)})$, and suppose that the two classes have same number of points. For visualization, the dispersion level of each class is represented by concentric circles centered at the mean, and the radius of the smallest circle equals the standard deviation of each dimension, whereas the radius of the other two circles are two- and three-times that of the smallest circle, respectively. Fig. 3 shows how the ESDR in the feature space varies with $\log_2\sigma$ for the corresponding Gaussian data in Fig. 2. From Fig. 3, we can see that for two classes drawn from different Gaussian distributions, when $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are nearly the same, the ESDR varies along a step-shaped curve with respect to σ, and as the difference between $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ increases, the curve transforms gradually toward an impulse-shaped curve.

4. Numerical experiments

To verify the ability of the ESDR to reflect the class separability in the feature space, we conducted several numerical experiments on real-world datasets and Gaussian data. Following the above analysis, we only focused on an RBF kernel. All of the experiments, including the implementation of an SVM, were conducted using Matlab.

4.1. Real world datasets

In our experiments, we employed five binary datasets from the UC Irvine machine-learning repository. Table 1 contains detailed information on these five datasets. To evaluate the real performance of an SVM with different kernel parameters, we apply a ten-fold cross-validation along with a grid-search method [15]. More specifically, for each dataset with parameter combination (σ, C), the dataset is equally divided into ten subsets, and each subset is tested once using the classifier trained

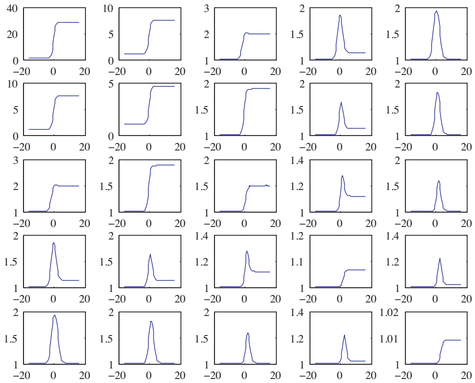

Fig. 3. ESDR with respect to $\log_2\sigma$ for corresponding cases in Fig. 2.

Table 1
Detailed information on the five datasets (including their names, number of examples of both classes, and dimension).

Data	Name	#(Class 1,Class 2)	Dimension
1	Banknote authentication	(762,610)	4
2	Planning relax [3]	(130,52)	12
3	Vertebral column	(210,100)	6
4	Indian liver patient	(414,165)	10
5	Tic-tac-toe endgame	(626,332)	9

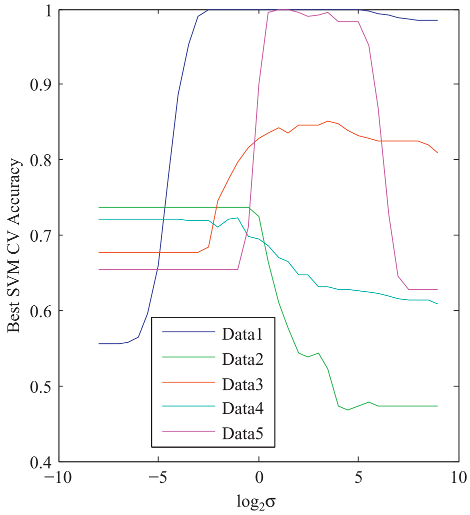

Fig. 4. The best CV accuracy with respect to $\log_2\sigma$.

with the remaining nine subsets. The total accuracy is the cross-validation accuracy, or CV accuracy for short. Here, σ is set within $\{2^{-8}, 2^{-7.5}, \cdots, 2^{9}\}$, and for each σ, C is set within $\{2^{-1}, 2^{-0.5}, \cdots, 2^{16}\}$, and the best CV accuracy is obtained for every given σ. Before the CV, to avoid an arbitrarily dominating feature, all feature vectors of each dataset are scaled to those with zero mean and unit variance [19]. Fig. 4 shows the best CV accuracy with respect to $\log_2\sigma$ for each dataset.

With the same pre-scaling work, the variations in ESDR, DBTC, and J_4 with respect to the kernel parameter are shown in Fig. 5–7, respectively. Comparing these three figures with Fig. 4, we found that the ESDR conforms much better than the DBTC and J_4 for all five benchmark datasets. Specifically, an increase in the ESDR does not lead to a decrease in the CV accuracy, and a decrease in the ESDR does not lead to an increase in the CV accuracy. In the case of the DBTC, a huge discrepancy exists between the variation in the DBTC and that of the CV accuracy on data 1, 3, and 4. The variation of J_4 agrees with that of the CV accuracy on data 1 and 2; however, a non-ignorable inconformity is found on data 3, and a reverse trend appears on data 4 and 5. The results show that as a good class separability criterion, a larger value of the ESDR corresponds to a higher SVM classification accuracy, indicating its ability to quantify the class separability in

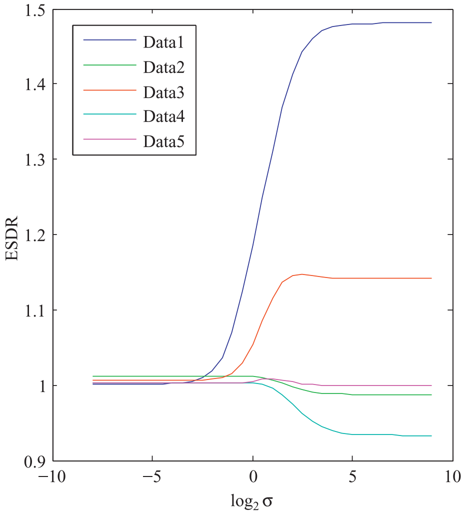

Fig. 5. ESDR with respect to $\log_2\sigma$.

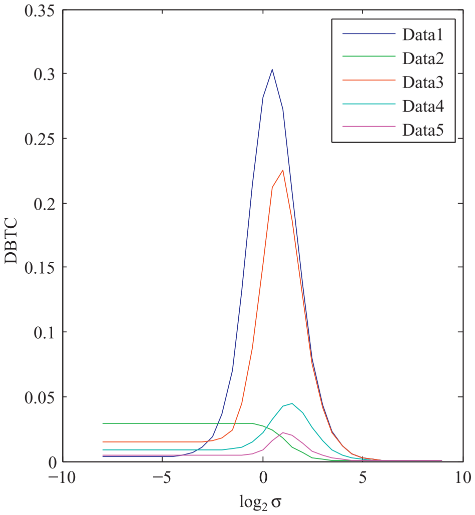

Fig. 6. DBTC with respect to $\log_2\sigma$.

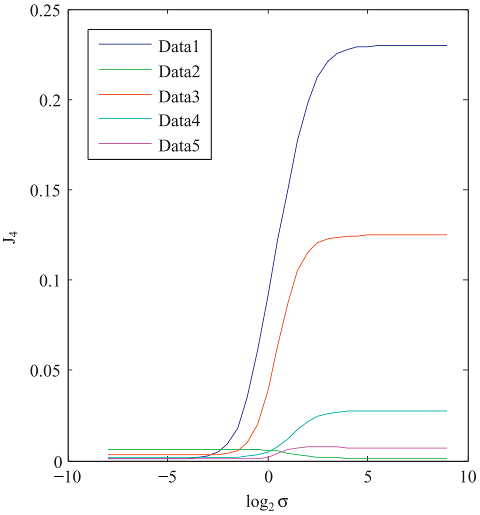

Fig. 7. J_4 with respect to $\log_2\sigma$.

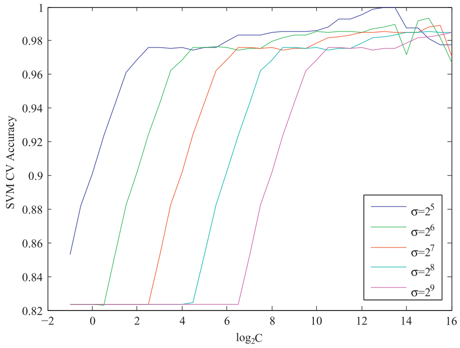

Fig. 8. CV accuracy of data 1 with respect to $\log_2 C$ for a large σ.

the feature space. However, this property cannot be guaranteed by the DBTC or J_4. Thus, the advantages of the ESDR over previous criteria, as discussed in Section 3, are verified.

There also exists some slight inconformity between the ESDR and CV accuracy, which can be ignored owing to the fact that the CV accuracy is just an approximation of the real generalization error. However, this fails to explain the declining part at the end of curve 1 (corresponding to data 1) in Fig. 4. As can be seen in Fig. 5, the corresponding part slowly increases to a fixed level. Actually, this fact is not a contradiction to the above experiment results. According to [21], as $\sigma \to \infty$, an SVM classifier with an RBF kernel and penalty coefficient C behaves almost the same as a linear SVM classifier with penalty coefficient $\tilde{C}$, where $C = \tilde{C}\sigma^2$. Therefore, for an SVM with an RBF kernel, if σ is relatively large, C also has to be sufficiently large to avoid missing the best classification accuracy. Fig. 8 shows the variation in CV accuracy with respect to $\log_2 C$ in the case of data 1 with a relatively large σ. From Fig. 8, we can easily conclude that the above inconsistency is caused by the limited search range of parameter C. Provided that C can be set as arbitrarily large, the end part of curve 1 in Fig. 4 will be approximately horizontal, and thus is more in line with the ESDR. The case of data 3 suffers from the same problem to a lesser extent.

Actually, the convergent property referred to above can also be explained using the ESDR. From Section 3, we know that, as $\sigma \to \infty$, the ESDR of the feature space converges to that of the original Euclidean space. Therefore, with a relatively large σ, the class separability in the feature space is close to that in the linear space, and the corresponding SVM classifier with an RBF kernel behaves like a linear SVM. However, as $\sigma \to \infty$, the SVM may suffer from numerical problems as the value of the RBF kernel is approximately infinitesimal. Generally, however, the parameters we choose in practice are not that extreme, and will not exceed the computing requirement of the computer.

Furthermore, as stated in Section 3, the ESDR is defined in a normalized ratio form, and its value is 1 for data with completely mixed up classes. In Figs. 4 and 5, comparing the maximum value of every curve, it is easy to find that, apart from the case of data 5, a bigger maximum ESDR corresponds to a higher maximum CV accuracy. Thus, given the value of the ESDR for a certain space, we can obtain a general idea regarding the extent to which the classes are separated from each other without referring to other information. This is a unique property of the ESDR.

As a class separability measure, the ESDR can certainly be applied to choosing the kernel parameters for an SVM. Here, we still use the above five datasets and choose parameter pair (σ, C) from the same values in Fig. 4. We mainly focus on four methods. The first is a simple exhausted grid search technique combined with a ten-fold CV. The remaining three are based on a class separation, namely, ESDR/DBTC/J_4-CV, the implementation of which can be described through the following procedure.

1. For certain data, scale all feature vectors to those with zero mean and unit variance.
2. Compute the value of ESDR/DBTC/J_4 for every σ from $\{2^{-8}, 2^{-7.5}, \cdots, 2^9\}$.
3. Obtain the desired kernel parameter, as $\sigma_{optimal}$, which leads to the maximum value of the ESDR, DBTC, and J_4.
4. With $\sigma_{optimal}$, apply a ten-fold CV for each discrete C from $\{2^{-1}, 2^{-0.5}, \cdots, 2^{16}\}$.
5. Select the best C corresponding to the highest CV accuracy.

Table 2 shows the experimental results.

From Table 2, we see that for all datasets except for dataset 1, the ESDR achieves a better CV accuracy than DBTC and J_4. The suboptimal performances on dataset 1 are due to the limited search range of C, as previously analyzed. Furthermore, the ESDR obtains parameters that lead to a CV accuracy comparable to that of a grid search technique with much less time consumption. As the computational complexity of the ESDR is fairly low, we can choose better kernel parameters from a dense number of options and even adopt a gradient method.

Table 2
Comparison of the best CV accuracy (Acc) achieved by different parameter selecting techniques and their corresponding running time *T*.

Data	Grid search-CV		ESDR-CV		DBTC-CV		J_4-CV	
	Acc	T (s)	Acc	T (s)	Acc	T (s)	Acc	T (s)
1	1	3630.7	0.9847	96.7550	1	61.0177	0.9847	96.5371
2	0.7363	4297.8	0.7363	16.9723	0.7363	18.9390	0.7363	16.9654
3	0.8516	5317.2	0.8452	688.4488	0.8419	593.8834	0.8258	22.9523
4	0.7237	1671.9	0.7219	73.3224	0.6649	1948.6	0.6079	35.5591
5	1	24181	1	242.1531	1	242.1573	0.9906	1213.8

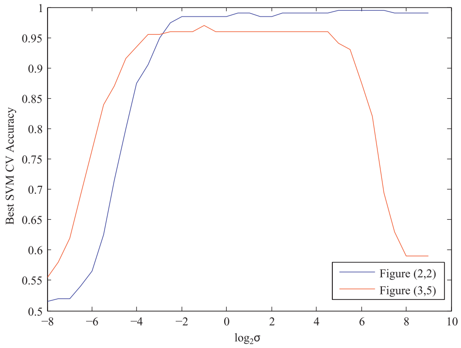

Fig. 9. The best CV accuracy with respect to $\log_2\sigma$ for the two cases in Fig. 3.

4.2. Gaussian data

In part 3.3, Figs. 2 and 3 clearly show how the ESDR changes with respect to σ for different Gaussian data. Considering the satisfying results from the last section, we believe that the ESDR can actually represent the class separability in the feature space. Herein, we did not verify all 25 cases, but simply considered two, images (2, 2) and (3, 5) in Figs. 2 and 3, respectively, to test the real behaviors of an SVM classifier. Here, σ is still set within $\{2^{-8}, 2^{-7.5}, \cdots, 2^{9}\}$, and the value range of C is broadened to $\{2^{-6}, 2^{-5.5}, \cdots, 2^{20}\}$. For each case, we generate 100 samples for each class according to the corresponding Gaussian distribution. Fig. 9 shows the best CV accuracy with respect to $\log_2\sigma$. Comparing Fig. 9 with Fig. 3, we can see that the variation in CV accuracy of the SVM agrees well with that of the ESDR in both cases, indicating that the ESDR can actually serve as a powerful tool for analyzing the model selection for an SVM under a specific data distribution.

5. Conclusion

In the construction of an SVM, selecting a proper kernel and tuning its parameters are crucial. Among all methods used for a model selection, those based on the class separability are quite efficient. In this paper, we proposed a new class separability measure called the Expected Square Distance Ratio (ESDR) and applied it to the choice of RBF kernel parameter σ for an SVM. After analyzing its properties and comparing it with other class separability measures (the DBTC and J_4), we found that the ESDR is defined in a more proper and consistent manner with no theoretical flaws. Experimental results on real-world datasets indicate that the ESDR is more consistent with the accuracy of the corresponding SVM classifier, and can also choose the kernel parameter with a higher level of accuracy compared with other methods. In addition, we employed the ESDR on Gaussian data to analytically show the variation in class separation with respect to σ. The results reveal that the ESDR is capable of dealing with the SVM model selection problem under an exact form of data distribution.

The experimental results demonstrate that the ESDR can be used for selecting a model of an SVM for any kernel and any form of data distribution, and can even be generalized to other kernel methods. In addition, the concept of the ESDR can be generalized to other dissimilarity measures besides the Euclidean distance function. Further explorations of the above possibilities are areas of future research.

References

[1] H. Ao, J. Cheng, Y. Yang, T.K. Truong, The support vector machine parameter optimization method based on artificial chemical reaction optimization algorithm and its application to roller bearing fault diagnosis, J. Vibration Control 21 (12) (2015) 2434–2445.
[2] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Comput. 12 (10) (2000) 2385–2404.
[3] R. Bhatt, Planning-relax dataset for automatic classification of eeg signals, UCI Mach. Learn. Reposit.
[4] C.J. Burges, A tutorial on support vector machines for pattern recognition, Data Mining Knowl. Discov. 2 (2) (1998) 121–167.
[5] C.-F. Chao, M.-H. Horng, The construction of support vector machine classifier using the firefly algorithm, Comput. Intell. Neurosci. 2015 (2015) 2.
[6] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Mach. Learn. 46 (1-3) (2002) 131–159.

[7] M. Fu, Y. Tian, F. Wu, Step-wise support vector machines for classification of overlapping samples, Neurocomputing 155 (2014) 159–166.

[8] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic press, 2013.

[9] F. Gao, J. Mei, J. Sun, J. Wang, E. Yang, A. Hussain, Target detection and recognition in sar imagery based on kfda, J. Syst. Eng. Electron. 26 (4) (2015) 720.

[10] H. Ge, Y. Jiang, F. Lian, Y. Zhang, S. Xia, Quantitative determination of aflatoxin b1 concentration in acetonitrile by chemometric methods using terahertz spectroscopy, Food Chem. 209 (2016) 286–292.

[11] M. Girolami, Mercer kernel-based clustering in feature space, Neural Netw. IEEE Trans. 13 (3) (2002) 780–784.

[12] C. Gold, A. Holub, P. Sollich, Bayesian approach to feature selection and parameter tuning for support vector machine classifiers, Neural Netw. 18 (5) (2005) 693–701.

[13] T.A. Gomes, R.B. Prudêncio, C. Soares, A.L. Rossi, A. Carvalho, Combining meta-learning and search techniques to select parameters for support vector machines, Neurocomputing 75 (1) (2012) 3–13.

[14] L.K. Hansen, P. Salamon, Neural network ensembles, Pattern Anal. Mach. Intell. IEEE Trans. 12 (10) (1990) 993–1001.

[15] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification, 2003.

[16] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, Neural Netw. IEEE Trans. 13 (2) (2002) 415–425.

[17] J. Hu, J. Qi, Y. Peng, Q. Ren, Predicting electrical evoked potential in optic nerve visual prostheses by using support vector regression and case-based prediction, Inf. Sci. 290 (2015) 7–21.

[18] F. Imbault, K. Lebart, A stochastic optimization approach for parameter tuning of support vector machines, in: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 4, IEEE, 2004, pp. 597–600.

[19] I. Jolliffe, Principal Component Analysis, Wiley Online Library, 2002.

[20] S.S. Keerthi, Efficient tuning of svm hyperparameters using radius/margin bound and iterative algorithms, Neural Netw. IEEE Trans. 13 (5) (2002) 1225–1229.

[21] S.S. Keerthi, C.-J. Lin, Asymptotic behaviors of support vector machines with gaussian kernel, Neural Comput. 15 (7) (2003) 1667–1689.

[22] K.I. Kim, K. Jung, H.J. Kim, Face recognition using kernel principal component analysis, Signal Process. Lett. IEEE 9 (2) (2002) 40–42.

[23] C.-Y. Lee, X. Yao, Evolutionary programming using mutations based on the lévy probability distribution, Evol. Comput. IEEE Trans. 8 (1) (2004) 1–13.

[24] C. Li, X. An, R. Li, A chaos embedded gsa-svm hybrid system for classification, Neural Comput. Appl. 26 (3) (2015) 713–721.

[25] S.-W. Lin, Z.-J. Lee, S.-C. Chen, T.-Y. Tseng, Parameter determination of support vector machine and feature selection using simulated annealing approach, Appl. soft Comput. 8 (4) (2008) 1505–1512.

[26] A.C. Lorena, A.C. De Carvalho, Evolutionary tuning of svm parameter values in multiclass problems, Neurocomputing 71 (16) (2008) 3326–3334.

[27] S. Maldonado, E. Carrizosa, R. Weber, Kernel penalized k-means: A feature selection method based on kernel k-means, Inf. Sci. 322 (2015) 150–160.

[28] S. Maldonado, R. Weber, F. Famili, Feature selection for high-dimensional class-imbalanced data sets using support vector machines, Inf. Sci. 286 (2014) 228–246.

[29] L. Shen, H. Chen, Z. Yu, W. Kang, B. Zhang, H. Li, B. Yang, D. Liu, Evolving support vector machines using fruit fly optimization for medical data classification, Knowl. Based Syst. 96 (2016) 61–75.

[30] J. Sun, C. Zheng, X. Li, Y. Zhou, Analysis of the distance between two classes for tuning svm hyperparameters, Neural Netw. IEEE Trans. 21 (2) (2010) 305–318.

[31] Y. Tian, M. Fu, F. Wu, Steel plates fault diagnosis on the basis of support vector machines, Neurocomputing 151 (2015) 296–303.

[32] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, 2000.

[33] V. Vapnik, S.E. Golowich, A. Smola, Support vector method for function approximation, regression estimation, and signal processing, Adv. Neural Inf. Proces. Syst. (1997) 281–287.

[34] L. Wang, Feature selection with kernel class separability, Pattern Anal. Mach. Intell. IEEE Trans. 30 (9) (2008) 1534–1546.

[35] M. Wang, G. Yan, Z. Fei, Kernel pls based prediction model construction and simulation on theoretical cases, Neurocomputing (2015).

[36] W. Wang, Z. Xu, W. Lu, X. Zhang, Determination of the spread parameter in the gaussian kernel for classification and regression, Neurocomputing 55 (3) (2003) 643–663.

[37] X. Wang, F. Huang, Y. Cheng, Super-parameter selection for gaussian-kernel svm based on outlier-resisting, Measurement 58 (2014) 147–153.

[38] Y. Wang, X. Wang, W. Liu, Unsupervised local deep feature for image recognition, Inf. Sci. 351 (2016) 67–75.

[39] W.-T. Wong, F.Y. Shih, J. Liu, Shape-based image retrieval using support vector machines, fourier descriptors and self-organizing maps, Inf. Sci. 177 (8) (2007) 1878–1891.

[40] K.-P. Wu, S.-D. Wang, Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space, Pattern Recognit. 42 (5) (2009) 710–717.

[41] Q. Wu, R. Law, E. Wu, J. Lin, A hybrid-forecasting model reducing gaussian noise based on the gaussian support vector regression machine and chaotic particle swarm optimization, Inf. Sci. 238 (2013) 96–110.

[42] Y. Xiao, H. Wang, W. Xu, Parameter selection of gaussian kernel for one-class svm, Cybernet. IEEE Trans. 45 (5) (2015) 941–953.

[43] Z. Xue, P. Du, H. Su, Harmonic analysis for hyperspectral image classification integrated with pso optimized svm, Selected Topics Appl. Earth Observ. Remote Sensing, IEEE J. 7 (6) (2014) 2131–2146.

[44] W. Yan, H. Shao, X. Wang, Soft sensing modeling based on support vector machine and bayesian model selection, Comput. Chem. Eng. 28 (8) (2004) 1489–1498.

[45] S. Yin, X. Gao, H.R. Karimi, X. Zhu, Study on support vector machine-based fault detection in tennessee eastman process, Abstract and Applied Analysis, vol. 2014, Hindawi Publishing Corporation, 2014.

[46] Y. Zhang, P. Zhang, Machine training and parameter settings with social emotional optimization algorithm for support vector machine, Pattern Recognit. Lett. 54 (2015) 36–42.