Kernel Parameter Selection for Support Vector Machine Classification

Zhiliang Liu* and Hongbing Xu
School of Mechanical, Electronic, and Industrial Engineering University of Electronic Science and Technology of China
Chengdu, P. R. China, 611731

Received: 23/08/2012; Accepted: 19/06/2013

ABSTRACT

Parameter selection for kernel functions is important to the robust classification performance of a support vector machine (SVM). This paper introduces a parameter selection method for kernel functions in SVM. The proposed method tries to estimate the class separability by cosine similarity in the kernel space. The optimal parameter is defined as the one that can maximize the between-class separability and minimize the within-class separability. The experiments for several kernel functions are conducted on eight benchmark datasets. The results demonstrate that our method is much faster than grid search with comparable classification accuracy. We also found that the proposed method is an extension of a reported method in reference [2].

Keywords: parameter selection; kernel function; cosine similarity; support vector machine

NOTATION

U : a given dataset $U = \{\mathbf{x}_i; y_i\}^N$ that contains N instance-label pairs; the instance matrix $\mathbf{X} = \{\mathbf{x}_i\}^N$, and the label vector $\mathbf{Y} = \{y_i\}^N$;

L : the number of classes (labels) that are used to label all instances in $\mathbf{X}$;

$\theta(\mathbf{x}_i,\mathbf{x}_j)$: the angle between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$;

$\langle \mathbf{x}_i,\mathbf{x}_j \rangle$: the dot product operation between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$;

$\|\mathbf{x}_i\|$: the Euclidean norm of a vector $\mathbf{x}_i$, and $\langle \mathbf{x}_i,\mathbf{x}_i \rangle = \|\mathbf{x}_i\|^2$;

$\Phi(\mathbf{x}_i)$: the new instance in the kernel space by the feature mapping Φ;

*Corresponding author. Email: {zhiliang_liu, hbxu}@uestc.edu.cn

$\kappa(\mathbf{x}_i,\mathbf{x}_j)$: kernel function, and a bar on it mean the corresponding normalized kernel function;
σ : the width of features (spread parameter) in the Gaussian radial basis function;
r : the polynomial degree in the (normalized) polynomial kernel function;
W, B : the within-class separability and the between-class separability, respectively;
N_i : the number of instances in the ith class, $1 \le i \le L$;
N : the total number of instances in a dataset U, that is, $N = \Sigma\, N_i$;
J : the objective function used to measure the class separability;
$(\cdot)^T$: the transpose operation of a vector or a matrix;
$Sign(\cdot)$: the sign function that extracts the sign of a real number;
$Avg(\cdot)$: the average function simply adding all elements together and dividing by the total number of elements in a matrix.

1. INTRODUCTION

Support vector machine [1] is a supervised statistical learning technique in the field of machine learning. Based on the structural risk minimization, SVM tries to find an optimized hyper-plane in a kernel space where training instances are linearly separable. SVM is usually implemented through the so-called soft-margin SVM because of its attractive characteristics [2-4] listed as follows:

(1) It possesses good generality under the principle of structural risk minimization;
(2) It can deal with non-linear problems by kernel methods;
(3) It is robust to noisy instances after introducing slack variables;
(4) It produces sparse solutions since the optimal hyper-plane depends only on support vectors;
(5) It guarantees convergence.

Due to the merits mentioned above, applications of SVM are easily found in various fields of science and engineering, such as condition monitoring and fault diagnosis [5,6]. However, it is reported that robust performance of SVM requires a properly adjusted parameter in kernel functions [7], such as σ in the Gaussian radial basis function (RBF) and the degree in the polynomial kernel.

Grid search is widely used in parameter selection due to its simplicity. It conducts an exhaustive search on a finite set of parameters. The optimal parameter corresponds to the one that has the highest score on a criterion. Grid

search is adopted in [8] to determine the optimal σ. However, the success of grid search stands on an assumption that the optimal value lies in the pre-defined searching range; otherwise, grid search fails to work. Another problem is that grid search is quite time-consuming especially when using classification accuracy as the criterion.

Several intelligent methods have been proposed for parameter selection recently. Wang et al. [4] determined the optimal σ by minimizing Fisher discriminant function. Xu et al. [9] derived a formula to determine the optimal σ by considering parameter selection as a recognition problem. Ali and Smith-Miles [10, 11] proposed Bayesian and Laplace methods for selecting the degree in the polynomial kernel. However, the methods in [4, 9-11] limit their application to a specific kernel. These methods are not applicable for a different kernel function. Zhang et al. [12] computed the kernel parameters through optimizing an objective function designed for measuring the classification reliability of kernel minimum distance. Li et al. [2] proposed a parameter method for a group of kernels, i.e. normalized kernels. However, it is not applicable to non-normalized kernels, for example, the polynomial kernel.

The motivation of this work is to develop a general parameter selection method for kernel functions in SVM. Inspired by the concept of class separability, we establish an objective function of class separability in the kernel space with respect to the kernel parameter. Since distance similarity is less efficient in the high-dimensional kernel space [13], we adopt cosine similarity to measure the class separability. The class separability consists of the within-class separability and the between-class separability. In our definition, large class separability means small within-class separability but large between-class separability. Parameter selection turns to maximize the class separability with respect to the kernel parameter. The optimal kernel parameter is defined as the one with the largest class separability, i.e. the maximizer. The experiments are conducted on benchmark datasets for several kernels, including the Gaussian RBF kernel, the polynomial kernel, and the normalized polynomial kernel. The results demonstrate that the proposed method is much faster than grid search with comparable accuracy. We also prove that the proposed method is actually an extension of a reported method in [2].

The rest of the paper is organized as follows. In *Section* 2, we introduce the theory basis of the soft-margin SVM. In *Section* 3, we first describe the kernel functions and similarity measures in the kernel space; and then we propose our method. The proposed method is validated on benchmark datasets in *Section* 4. *Section* 5 includes the conclusions and description of future work.

2. THE SOFT-MARGIN SUPPORT VECTOR MACHINE

This section reviews the main ideas behind the soft-margin SVM. A more detailed reference is found in [14]. In this paper, we limit the use of SVM to the soft-margin SVM classification.

We now describe kernel methods briefly. Kernel methods are useful for the successful application of SVM. Kernel methods are a set of approaches to mapping data in the original feature space into the kernel space without ever knowing the mapping function Φ explicitly. Kernel functions define inner product spaces (Hilbert spaces) in the following way:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = < \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) > . \tag{1}$$

Powered by a proper kernel, SVM is enabled to deal with not only linear problems but also non-linear problems. In Section 3, several classical kernels and their normalized versions are introduced.

Given a training dataset U containing N instance-label pairs $\{\mathbf{x}_i, y_i\}$, where $y_i \in \{1, -1\}$ represents labels of two classes. SVM seeks an optimal hyper-plane $f(\Phi(\mathbf{x})) = \mathbf{w}^{\mathrm{T}}\Phi(\mathbf{x}) + b = 0$ in the kernel space by maximizing the margin width between $f(\Phi(\mathbf{x})) = \pm 1$, where $\mathbf{w}$ is a weight vector, and b is a scalar. The margin width equals $2/\|\mathbf{w}\|$ in SVM. The goal of maximizing the margin width is equivalent to the following optimization problem:

$$\mathbf{w}^*, \mathrm{b}^*, \xi_i^* = \arg\min_{\mathbf{w}, b} \left(\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \right), \tag{2}$$

$$\text{subject to } y_i \cdot f(\Phi(\mathbf{x}_i)) \geq 1 - \xi_i; \xi_i \geq 0; C > 0; \mathbf{w} \in \mathbb{R}^n; i = 1, 2, \cdots, N$$

where C is the regularization parameter; and ξ_i is the slack variable that is introduced for instances violating the edges of the margin. The optimization problem in eqn (2) is further transformed to the following equivalent dual problem:

$$\begin{aligned} &\boldsymbol{\alpha}^* = \arg\max_{\alpha} L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \right] \\ &\text{subject to } \sum_{i=1}^{N} (y_i \alpha_i) = 0; 0 \leq \alpha_i \leq C; \boldsymbol{\alpha} = \{\alpha_i\}^N; i = 1, 2, \cdots, N \end{aligned}, \tag{3}$$

After $\boldsymbol{\alpha}^*$ is available by solving eqn (3), $\mathbf{w}^*$ and b^* are computed by

$$\begin{cases} \mathbf{w}^* = \sum_{i=1}^{N} \left[\alpha_i^* y_i \Phi(\mathbf{x}_i) \right] = \sum_{t=1}^{p} \left[\alpha_t^* y_t \Phi(\mathbf{x}_t) \right] \\ b^* = \frac{1}{p} \sum_{t=1}^{p} \left[y_t - \alpha_t^* y_t \kappa(\mathbf{x}_t, \mathbf{x}_t) \right] \end{cases}, \tag{4}$$

where p is the total number of non-zero elements in $\boldsymbol{\alpha}^*$. Those instances that have non-zero elements in $\boldsymbol{\alpha}^*$ are called support vectors. So p is the number of support vectors.

The predicted label for an unknown instance $\mathbf{x}$ is determined by

$$\hat{y} = f(\mathbf{x}) = Sign\left(\sum_{t=1}^{p} \left[\alpha_t^* y_t \kappa(\mathbf{x}_t, \mathbf{x}) \right] + \frac{1}{p} \sum_{t=1}^{p} \left[y_t - \alpha_t^* y_t \kappa(\mathbf{x}_t, \mathbf{x}_t) \right] \right). \tag{5}$$

The decision function in eqn (5) depends on support vectors together with their label information. Because p is usually much less than N, SVM produces sparse solutions.

SVM is initially developed to solve binary-class problems. Several approaches enable SVM to solve multi-class problems. One of the approaches is the one-against-all (OAA) approach that transforms a classification problem of L-class into L binary-class problems ($L \geq 3$) [1]. OAA is applied to the soft-margin SVM to solve multi-class problems in this paper.

3. THE PROPOSED METHOD

3.1. Kernel Functions and Similarity Measures

Kernel function is the most important component in kernel based techniques, such as SVM. Several kernel functions are developed according to the Hilbert-Schmidt theory and Mercer condition [15]. Three commonly used kernel functions are listed below:

(1) The Gaussian radial basis function

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right), \sigma \in (0, +\infty) \tag{6}$$

(2) The polynomial kernel

$$\kappa(x_i, x_j) = (<x_i, x_j> + 1)^r, r > 1, r \in \mathbb{Z}^+ \tag{7}$$

(3) The linear kernel

$$\kappa(\mathbf{x}_i,\mathbf{x}_j) = <\mathbf{x}_i,\mathbf{x}_j> \tag{8}$$

The normalized kernel [16] corresponding to an existing kernel function $\kappa(\mathbf{x}_i,\mathbf{x}_j)$ is defined as follows:

$$\bar{\kappa}(\mathbf{x}_i,\mathbf{x}_j) = \frac{\kappa(\mathbf{x}_i,\mathbf{x}_j)}{\sqrt{\kappa(\mathbf{x}_i,\mathbf{x}_i)}\sqrt{\kappa(\mathbf{x}_j,\mathbf{x}_j)}} . \tag{9}$$

It can be proved that the normalized version of the Gaussian RBF kernel is the same as itself because the norm of any instance is equal to one [17].

Distance and angle are two basic geometrical metrics in the kernel space. Both of them can be decomposed into forms of inner product as follows:

$$\left\|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\right\|^2 = \kappa(\mathbf{x}_i,\mathbf{x}_i) + \kappa(\mathbf{x}_j,\mathbf{x}_j) - 2\kappa(\mathbf{x}_i,\mathbf{x}_j), \tag{10}$$

$$\cos\theta(\Phi(\mathbf{x}_i),\Phi(\mathbf{x}_j)) = \frac{\kappa(\mathbf{x}_i,\mathbf{x}_j)}{\sqrt{\kappa(\mathbf{x}_i,\mathbf{x}_i)\kappa(\mathbf{x}_j,\mathbf{x}_j)}} . \tag{11}$$

3.2. The Algorithm of Parameter Selection

Class separability is a classic concept in feature selection for describing how instances scatter in a feature space [18]. It usually considers two principles.

Principle 1: Instances from the same class should be as similar as possible;

Principle 2: Instances from different classes should be as different as possible.

Class separability can be measured by either distance similarity or cosine similarity. Since the distance similarity is less efficient in the high-dimensional kernel space [13], we adopt cosine similarity in the proposed method. In a kernel space, cosine similarity matrix ($\mathbf{M}$) is defined by

$$\begin{aligned}
\mathbf{M} &= \begin{bmatrix} \mathbf{K}'_{11} & \cdots & \mathbf{K}'_{1L} \\ \vdots & \ddots & \vdots \\ \mathbf{K}'_{L1} & \cdots & \mathbf{K}'_{LL} \end{bmatrix}, \\
\mathbf{K}'_{ij} &= \begin{bmatrix} \cos\theta(\Phi(\mathbf{x}_1^{(i)}),\Phi(\mathbf{x}_1^{(j)})) & \cdots & \cos\theta(\Phi(\mathbf{x}_1^{(i)}),\Phi(\mathbf{x}_{N_j}^{(j)})) \\ \vdots & \ddots & \vdots \\ \cos\theta(\Phi(\mathbf{x}_{N_i}^{(i)}),\Phi(\mathbf{x}_1^{(j)})) & \cdots & \cos\theta(\Phi(\mathbf{x}_{N_i}^{(i)}),\Phi(\mathbf{x}_{N_j}^{(j)})) \end{bmatrix},
\end{aligned} \tag{12}$$

where $\mathbf{x}^{(i)}$ is an instance from the ith class.

The within-class separability (W) and the between-class separability (B) are introduced to quantize the two principles in class separability. W can be estimated by average cosine similarity of instances in the same class; B can be estimated by average cosine similarity of instances in different classes. In this paper, W and B are estimated by

$$\begin{aligned} \mathrm{W} &= -Avg\left(\begin{bmatrix} \mathbf{K}'_{11} & & \\ & \ddots & \\ & & \mathbf{K}'_{LL} \end{bmatrix}\right) \\ &= -\frac{1}{\sum_{i=1}^{L} N_i^2} \sum_{i=1}^{L} \sum_{t=1}^{N_i} \sum_{k=1}^{N_i} \frac{\kappa(\mathbf{x}_t^{(i)}, \mathbf{x}_k^{(i)})}{\sqrt{\kappa(\mathbf{x}_t^{(i)}, \mathbf{x}_t^{(i)})\kappa(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(i)})}} \end{aligned}, \tag{13}$$

$$\begin{aligned} \mathrm{B} &= -Avg\left(\begin{bmatrix} \mathbf{K}'_{12} & \cdots & \mathbf{K}'_{1L} \\ & \ddots & \vdots \\ & & \mathbf{K}'_{(L-1)L} \end{bmatrix}\right) \\ &= -\frac{1}{\sum_{i=1}^{L} \sum_{\substack{j=1 \\ j \neq i}}^{L} N_i N_j} \sum_{i=1}^{L} \sum_{\substack{j=1 \\ j \neq i}}^{L} \sum_{t=1}^{N_i} \sum_{k=1}^{N_j} \frac{\kappa(\mathbf{x}_t^{(i)}, \mathbf{x}_k^{(j)})}{\sqrt{\kappa(\mathbf{x}_t^{(i)}, \mathbf{x}_t^{(i)})\kappa(\mathbf{x}_k^{(j)}, \mathbf{x}_k^{(j)})}} \end{aligned}. \tag{14}$$

The objective function of class separability is defined as

$$J = \mathrm{B} - \mathrm{W}. \tag{15}$$

By this definition, large class separability means small within-class separability and large between-class separability. W in eqn (13) and B in eqn (14) are functions with respect to the kernel parameter. The class separability is therefore a function with respect to the kernel parameter as well. In this sense, parameter selection becomes a one-dimensional optimization problem. The optimal parameter is defined as the one that maximizes the class separability in eqn (15), i.e. the maximizer.

If the objective function is continuous with respect to the kernel parameter, the maximizer could be found by one-dimensional search methods. For

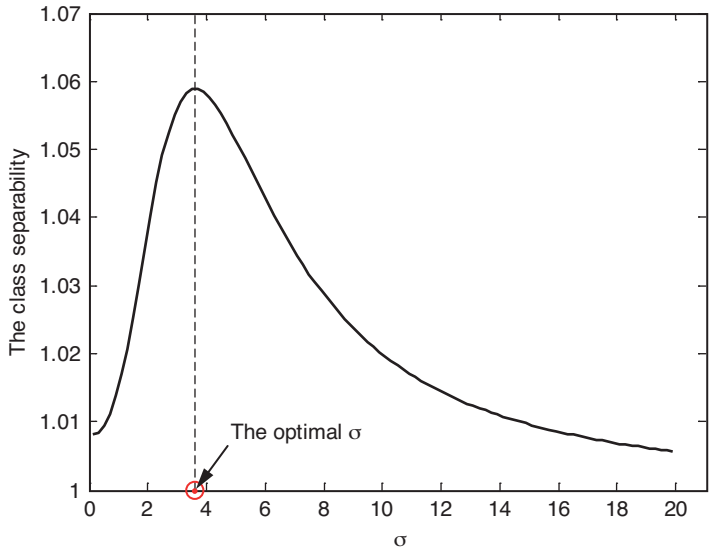

Figure 1. The class separability is computed on the Parkinson dataset [20] with respect to σ in the Gaussian RBF kernel: the maximizer exists.

example, in Fig. 1, the class separability is plotted with respect to σ in the Gaussian RBF kernel. Because the Gaussian RBF kernel is twice differentiable, the optimal σ could be found by Newton's method [19]. Derivatives of W and B are provided below:

$$D\mathrm{W}(\sigma) = -\frac{1}{\sum_{i=1}^{L} N_i^2} \sum_{i=1}^{L} \sum_{t=1}^{N_i} \sum_{k=1}^{N_i} \left[\kappa(\mathbf{x}_t^{(i)}, \mathbf{x}_k^{(i)}) \left\| \mathbf{x}_t^{(i)} - \mathbf{x}_k^{(i)} \right\|^2 \Big/ \sigma^3 \right], \tag{16}$$

$$D\mathrm{B}(\sigma) = -\frac{1}{\sum_{i=1}^{L} \sum_{\substack{j=1 \\ j \neq i}}^{L} N_i N_j} \sum_{i=1}^{L} \sum_{\substack{j=1 \\ j \neq i}}^{L} \sum_{t=1}^{N_i} \sum_{k=1}^{N_j} \left[\kappa(\mathbf{x}_t^{(i)}, \mathbf{x}_k^{(j)}) \left\| \mathbf{x}_t^{(i)} - \mathbf{x}_k^{(j)} \right\|^2 \Big/ \sigma^3 \right], \tag{17}$$

$$D^2\mathrm{W}(\sigma) =$$

$$-\frac{1}{\sum_{i=1}^{L} N_i^2} \sum_{i=1}^{L} \sum_{t=1}^{N_i} \sum_{k=1}^{N_i} \left[\kappa(\mathbf{x}_t^{(i)}, \mathbf{x}_k^{(i)}) \left(\left\| \mathbf{x}_t^{(i)} - \mathbf{x}_k^{(i)} \right\|^4 - 3\sigma^2 \left\| \mathbf{x}_t^{(i)} - \mathbf{x}_k^{(i)} \right\|^2 \right) \Big/ \sigma^6 \right], \tag{18}$$

$$D^2\mathrm{B}(\sigma) = -\frac{1}{\sum_{i=1}^{L}\sum_{\substack{j=1\\ j\neq i}}^{L} N_i N_j} \sum_{i=1}^{L}\sum_{\substack{j=1\\ j\neq i}}^{L}\sum_{t=1}^{N_i}\sum_{k=1}^{N_j} \left[\kappa(\mathbf{x}_t^{(i)},\mathbf{x}_k^{(j)})\left(\left\|\mathbf{x}_t^{(i)}-\mathbf{x}_k^{(j)}\right\|^4 - 3\sigma^2\left\|\mathbf{x}_t^{(i)}-\mathbf{x}_k^{(j)}\right\|^2\right)\Big/\sigma^6\right] . \quad (19)$$

If the objective function is discrete with respect to the kernel parameter, we can find the optimal parameter in a predefined discrete set. For example, in Fig. 2, the class separability is discrete with respect to the polynomial degree in the polynomial kernel. If we specify a set of possible solutions, for example, {2,3,...,20}, we can find the optimal r by the solution that reaches the highest score of the class separability in the set of {2,3,...,20}.

We declare early that the method in [2] (called *Li's method* in this paper) is a special case of the proposed method in normalized kernels. In the normalized kernel space, the norm of any instance is equal to one, i.e. $\|\Phi(\mathbf{x}_i)\|^2 = 1$. Cosine similarity in eqn (11) is simplified into

$$\cos\,\theta\,(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) = \kappa(\mathbf{x}_i, \mathbf{x}_j). \quad (20)$$

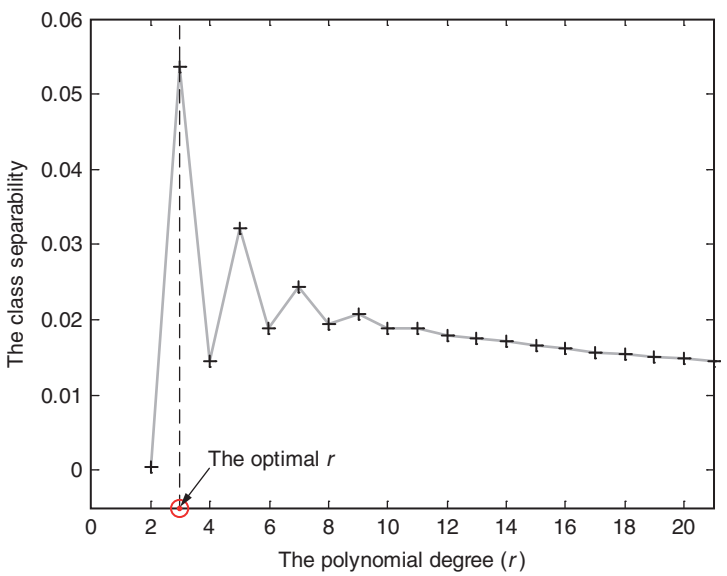

Figure 2. The class separability is computed on the Parkinson dataset [20] with respect to the polynomial degree (r) in the polynomial kernel: the maximizer exists.

The objective function of class separability in Li's method is then equivalent to the one in the proposed method. The proposed method could find the same optimal kernel parameter of normalized kernels as that in Li's method. However, Li's method limits its applications to only normalized kernels. The proposed method can deal with parameter selection for not only the normalized kernels but also other kinds of kernels, such as the polynomial kernel. Thus our method has the advantage of generality in practical applications.

4. VALIDATIONS

In this section, the proposed method is compared with grid search and Zhang's method [12] on eight benchmark datasets.

The used classifier in the experiments is SVM that is implemented by *svmtrain* and *svmclassify* in the Statistics Toolbox of MATLAB. OAA is applied when SVM has to deal with multi-class problems. Parameter selection is completed before SVM classification. C is optimized for a proper evaluation of performance. The searching ranges for kernel parameters and C are specified in Table 1 by following suggestions in [11,21]. In grid search, the selection criterion is the classification accuracy of SVM, while the selection criterion is the class separability defined in the proposed method. The three tested kernels are the Gaussian RBF kernel (kernel 1), the polynomial kernel (kernel 2) and the normalized polynomial kernel (kernel 3). Kernels 1 and 3 are normalized kernels whereas kernel 2 is a non-normalized kernel. Eight benchmark datasets from the University of California Irvine (UCI) repository [20] are used in experiments of validation. They are described in Table 2 with the number of classes, instances, and features.

Table 1. Parameter searching ranges setting.

Kernel	**Grid Search**		**Zhang's Method and The Proposed Method**	
	σ/r	C	σ/r	C
Kernel 1	$\{2^{-5}, 2^{-4}, ..., 2^{3}\}$	$\{2^{-10}, 2^{-9}, ..., 2^{10}\}$	$(0,+\infty)$	$\{2^{-10}, 2^{-9}, ..., 2^{10}\}$
Kernel 2	$\{2, 3, ..., 10\}$	$\{2^{-10}, 2^{-9}, ..., 2^{10}\}$	$\{2, 3, ..., 10\}$	$\{2^{-10}, 2^{-9}, ..., 2^{10}\}$
Kernel 3	$\{2, 3, ..., 10\}$	$\{2^{-10}, 2^{-9}, ..., 2^{10}\}$	$\{2, 3, ..., 10\}$	$\{2^{-10}, 2^{-9}, ..., 2^{10}\}$

Table 2. Information about benchmark datasets.

Dataset	Number of Classes	Number of Instances	Number of Features
Parkinson	2	195	22
Ionosphere	2	351	34
Sonar	2	208	60
Wine	3	178	13
Haberman's Survival	2	306	3
Iris	3	150	4
Glass Identification	6	214	9
Handwritten Numerals	2	400	649

Note: the handwritten numerals dataset consists of features of handwritten numerals ('0' ~ '9') extracted from a collection of Dutch utility maps. 200 instances per class (for a total of 2,000 instances) have been digitized in binary images. These digits are represented in terms of the 649 features. Due to limitation of computing resource, we only use 400 instances of two classes: '0' and '1'.

Classification accuracy and CPU time are two measures recorded during the experiments. *Classification accuracy* (CA) is computed by $N_c/(N_c+N_f)\times 100\%$, where N_c is the number of instances that are correctly classified, and N_f is the number of those falsely classified. The second measure records the time required in the three methods of parameter selection, i.e. *CPU time*. It is collected by two functions of *tic* and *toc* in MATLAB R2010a (32-bit) on a computer with a processor of AMD Athlon II X2 250 (3.0 GHz, 3.0 GHz) and an operating system of Windows XP Professional 2002 SP3 (32-bit).

We conduct thirty independent runs for the three approaches of parameter selection on the eight benchmark datasets. In each run, 5-fold cross-validation is employed to access the performance of classification accuracy. Experimental results are provided in Table 3. Classification accuracy is an average value over thirty runs. In grid search, CPU time is recorded for selecting the kernel parameter together with C because their selections are conducted at the same time. In Zhang's method and the proposed method, CPU time is separately recorded for the kernel parameter and C. For example, "0.12+0.81" in Table 3 means σ/r selection takes 0.12 seconds and C selection takes 0.81 seconds. Selected values for kernel parameters are provided in Table 3 as well.

Table 3. The experimental results.

Dataset	Kernel	Grid Search			Zhang's Method			The Proposed Method		
		CA (%)	CPU time (s)	σ/r	CA (%)	CPU time (s)	σ/r	CA (%)	CPU time (s)	σ/r
Parkinson	Kernel 1	94.01	12.17	2	93.51	0.16+0.83	2.59	93.35	0.12+0.81	3.61
	Kernel 2	91.16	11.60	3	91.78	0.05+0.58	3	91.50	0.04+0.62	3
	Kernel 3	93.76	10.88	5	91.58	0.09+1.36	5	91.66	0.08+1.24	3
Ionosphere	Kernel 1	95.12	32.25	4	93.80	0.31+2.22	5.82	95.23	0.29+2.27	3.99
	Kernel 2	92.73	17.71	2	92.68	0.16+1.43	2	92.74	0.14+1.62	2
	Kernel 3	93.30	30.84	3	92.82	0.34+2.58	3	92.49	0.30+2.53	2
Sonar	Kernel 1	86.41	12.89	8	83.66	0.12+0.88	18.31	87.31	0.11+0.87	5.88
	Kernel 2	82.09	12.24	3	82.52	0.05+0.75	3	82.79	0.04+0.72	3
	Kernel 3	87.02	12.10	3	87.63	0.11+1.41	3	87.34	0.09+1.38	3
Wine	Kernel 1	98.89	28.52	4	98.49	0.07+3.01	19.24	98.48	0.06+2.94	3.03
	Kernel 2	98.66	24.98	2	98.31	0.06+1.73	2	98.19	0.03+1.66	3
	Kernel 3	97.75	26.87	3	98.47	0.09+3.12	3	98.27	0.07+3.07	3
Haberman's Survival	Kernel 1	73.11	21.19	8	71.03	0.37+1.22	1.59	71.37	0.35+1.29	1.40
	Kernel 2	70.39	11.25	2	69.95	0.13+1.01	2	71.06	0.11+1.05	2
	Kernel 3	69.64	20.16	2	70.34	0.25+1.57	2	69.95	0.22+1.56	2
Iris	Kernel 1	97.09	22.19	4	95.71	0.10+1.42	1.32	95.84	0.08+1.45	1.51
	Kernel 2	96.67	12.11	2	96.31	0.03+1.38	2	96.67	0.02+1.28	3
	Kernel 3	95.87	20.34	3	96.40	0.06+2.29	3	95.96	0.04+2.32	3
Glass Identification	Kernel 1	71.31	67.36	2	67.02	0.13+4.03	0.66	70.37	0.12+4.08	2.13
	Kernel 2	65.39	58.07	3	64.36	0.05+3.88	3	66.18	0.04+3.94	2
	Kernel 3	70.92	64.29	5	68.73	0.11+5.12	5	68.44	0.09+5.08	2
Numerals Handwritten	Kernel 1	50.00	0.00	8	99.68	0.37+1.33	2546.12	99.53	0.36+1.29	1149.10
	Kernel 2	100.0	36.31	3	100.0	0.11+1.01	3	100.0	0.11+1.05	3
	Kernel 3	99.64	67.59	2	99.68	0.25+1.55	2	99.63	0.23+1.56	3

In terms of classification accuracy, the proposed method is comparable with grid search and Zhang's method in most datasets. "Comparable" here means differences of classification accuracy between the three methods are within 1%. Grid search has bad generality since it sometimes does not work. For example, when the Gaussian RBF kernel is used in the handwritten numerals dataset, the classification accuracy is 50% while the training accuracy is 100% (not shown in Table 3). SVM has perfect empirical risk minimization but shows the worst generality ability. This implies the over-fitting problem occurs. The reason is explained as follows. The optimal σ (1149.10 estimated by the proposed method) is far beyond the upper limitation specified in grid search, i.e. 8. Grid search could not select a proper parameter value and thus turns to be over-fitting. The observations in the handwritten numerals dataset show not only the drawback of grid search but also the importance of a robust method of parameter selection in SVM.

In terms of CPU time, the proposed method is much faster than grid search in all datasets, and it is slightly better than Zhang's method. In Table 3, CPU time recorded in grid search is the time elapsed for selecting both the kernel parameter and C. In order to make a fair comparison, we consider the CPU time in the proposed method for C selection. For example, when the Gaussian RBF kernel is used in the Parkinson dataset, CPU time taken in grid search is about 13 times that in the proposed method (12.17s/(0.12s+0.81s)). This shows the significant advantage of the proposed method.

In the experiments, we use classification accuracy of SVM as the selection criterion. Training and test processes repeated in parameter selection cost large computational resources. The situation is even worse in multi-class problems. It is because SVM needs additional strategies to deal with multi-class problems. For example, the OAA strategy used in this paper pushes SVM to be trained L times for one L-class problem. Moreover, CPU time of grid search increases along with the length of the searching range. A large searching range may produce a better result but takes more time in selection. CPU time in the proposed method is also proportional to the length of the searching range. However, it takes less CPU time than grid search because the proposed method avoids training and test processes in SVM. In general, grid search is time-consuming mainly because of two aspects: (1) the searching strategy is lazy; (2) the criterion is usually inefficient, such as the classification accuracy of SVM. The proposed method is computationally effective because of two characteristics: (1) the selection criterion is fast to compute; (2) the optimization method, e.g. Newton's method, is introduced into parameter selection for the Gaussian RBF kernel.

5. CONCLUSIONS

This paper introduces a parameter selection method for kernel functions in support vector machine classification. The optimal parameter in the proposed method is defined as the one that can maximize the class separability in the kernel space. The proposed method is applied to the three kernels and validated on the eight benchmark datasets. The experimental results demonstrate that our method is much faster than grid search with a comparable or even higher accuracy. By the idea of maximizing the class separability, the proposed method is potentially applied to any kernel functions with one parameter. Generally, the proposed method possesses the following characteristics:

(1) The proposed method can be used in parameter selection for kernel functions of one parameter variable.
(2) The proposed method is computationally economic by avoiding training and test process of SVM.
(3) The proposed method can directly deal with multi-class problems.
(4) The proposed method can be easily implemented without knowing any prior information.

REFERENCES

[1] Vapnik, V.N., Statistical Learning Theory, Wiley-Interscience, New York, 1998.
[2] Li, C.-H., Ho, H.-H., Liu, Y.-L., Lin, C.-T., Kuo, B.-C. and Taur, J.-S., An Automatic Method for Selecting the Parameter of the Normalized Kernel Function to Support Vector Machines, *Journal of Information Science and Engineering*, 2012, 28(1), 1–15.
[3] Gualdrón, O., Brezmes, J., Llobet, E., Amari, A., Vilanova, X., Bouchikhi, B. and Correig, X., Variable Selection for Support Vector Machine Based Multisensor Systems, *Sensor and Actuators B: Chemical,* 2007, 122(1), 259–268.
[4] Wang, W., Xu, Z., Lu, W. and Zhang, X., Determination of the Spread Parameter in the Gaussian Kernel for Classification and Regression, *Neurocomputing*, 2003, 55(3-4), 643–663.
[5] Yuan, S. and Chu, F., Support Vector Machines-Based Fault Diagnosis for Turbo-Pump Rotor, *Mechanical Systems and Signal Processing*, 2006, 20(4), 939–952.
[6] Qu, J., Liu, Z., Zuo M.J. and Huang, H.-Z., Feature Selection for Damage Degree Classification of Planetary Gearboxes Using Support Vector Machine, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2011, 225(9), 2250–2264.
[7] Villa, A., Fauvel, M., Chanussot, J., Gamba, P. and Benediktsson, J.A., Gradient Optimization for Multiple Kernel's Parameters in Support Vector Machines

Classification, *Proceedings of IEEE International Conference on Geoscience and Remote Sensing Symposium*, Boston, 2008.

[8] Widodoa, A., Kimb, E.Y., Sonc, J.-D., Yang, B.-S., Tanb, A.C.C., Gud, D.-S., Choid, B.-K. and Mathewb, J., Fault Diagnosis of Low Speed Bearing Based on Relevance Vector Machine and Support Vector Machine, *Expert Systems with Applications*, 2009, 36(3), 7252–7261.

[9] Xu, Z., Dai, M. and Meng, D., Fast and Efficient Strategies for Model Selection of Gaussian Support Vector Machine, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2009, 39(5), 1292–1307.

[10] Ali, S. and Smith, K.A., Automatic Parameter Selection for Polynomial Kernel, *Proceedings of IEEE International Conference on Information Reuse and Integration*, IEEE, Piscataway, 2003.

[11] Ali, S. and Smith-Miles, K., On Optimal Degree Selection for Polynomial Kernel With Support Vector Machines: Theoretical and Empirical Investigations, *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 2007, 11(1), 1–18.

[12] Zhang D., Chen S. and Zhou Z.-H., Learning the Kernel Parameters in Kernel Minimum Distance Classifier, *Pattern Recognition*, 2006, 39(1), 133–135.

[13] Apaydin, T. and Ferhatosmanoglu, H., Access Structures for Angular Similarity Queries, *IEEE Transactions on Knowledge and Data Engineering*, 2006, 18(11), 1512–1525.

[14] Burges, C.J.C., A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining Knowl. Discovery*, 1998, 2(2), 121–167.

[15] Courant, R. and Hilbert, D., *Methods of Mathematical Physics*, J. Wiley, New York, 1953.

[16] John, S.T. and Nello, C., *Kernel Methods for Pattern Analysis*, Cambridge Univ. Press, Cambridge, 2004.

[17] Liu, Z., Zuo, M.J. and Xu, H., A Gaussian Radial Basis Function Based Feature Selection Algorithm, *Proceedings of IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, IEEE, Ottawa, 2011.

[18] Wang, L., Feature Selection With Kernel Class Separability, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008, 30(9), 1534–1546.

[19] Chong, E.K.P. and Żak, S.H., *An Introduction to Optimization*, 3rd edn, John Wiley & Sons Inc., Hoboken, 2008.

[20] Frank, A., Asuncion, A., UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2010.

[21] Hsu, C.-W., Chang, C.-C. and Lin, C.-J., A Practical Guide to Support Vector Classification, *Technical Report*, Department of Computer Science, National Taiwan University, 2010.