An Analytical Approach to Fast Parameter Selection of Gaussian RBF Kernel for Support Vector Machine

ZHILIANG LIU1, MING J. ZUO1,2, XIAOMIN ZHAO2 AND HONGBING XU3
1School of Mechanical, Electronic, and Industrial Engineering
University of Electronic Science and Technology of China
Chengdu, 611731 P.R. China
E-mail: Zhiliang_Liu@uestc.edu.cn
2Department of Mechanical Engineering
University of Alberta
Edmonton, T6G2G8 Canada

The Gaussian radial basis function (RBF) is a widely used kernel function in support vector machine (SVM). The kernel parameter $\sigma$ is crucial to maintain high performance of the Gaussian SVM. Most previous studies on this topic are based on optimization search algorithms that result in large computation load. In this paper, we propose an analytical algorithm to determine the optimal $\sigma$ with the principle of maximizing between-class separability and minimizing within-class separability. An attractive advantage of the proposed algorithm is that no optimization search process is required, and thus the selection process is less complex and more computationally efficient. Experimental results on seventeen real-world datasets demonstrate that the proposed algorithm is fast and robust when using it for the Gaussian SVM.

Keywords: parameter selection, Gaussian radial basis function, class separability, support vector machine, distance similarity

1. INTRODUCTION

Support vector machine (SVM) is an important technique of supervised learning in the field of machine learning. By introducing the principle of structural risk minimization, SVM aims to find an optimized hyper-plane by which training instances of different classes are linearly separable. Because of its many attractive properties and a promising empirical performance [1, 2], SVM quickly collected attentions from researchers who have applied SVM to both science and engineering, e.g. condition monitoring and fault diagnosis [3-11]. Among existed kernels in SVM, the Gaussian radial basis function (RBF) kernel is a widely used one due to its attractive characteristics [1, 2], e.g. the property of structure-preserving. The Gaussian RBF kernel has a form of

$$\kappa(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right),$$

where $\sigma$ is the only parameter named by width of features.

In [5-8], the parameter $\sigma$ is specified by a default value, e.g. $\sigma = 1$. However, it is reported that $\sigma$ is crucial to robust performance of SVM whereas an arbitrary value of $\sigma$ cannot guarantee satisfactory performance [1]. For example, if the parameter $\sigma$ is close to zero, SVM tends to over-fitting since all training instances are used as support vectors in
this case. SVM has perfect predictions for all data in the training subset but may have poor performance on the test subset. If the parameter $\sigma$ tends to infinity, under-fitting occurs in SVM because all training instances are considered as one instance. All instances, either from the training subset or from the testing subset, are classified into one class. These two extreme cases also indicate that selecting a proper value of $\sigma$ is necessary and worth to do in practice.

**NOTATION**

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

$n$ the dimension of an instance $x_i$ in $X$, that is, the number of features;

$F_i$ the $i$th feature in $X$, $X = \{F_i\}^n$;

$L$ the number of classes (labels) that are used to label all instances in $X$;

$\theta(x_i, x_j)$ the angle between two vectors, say $x_i$ and $x_j$;

$\langle x_i, x_j \rangle$ the dot product operation between $x_i$ and $x_j$;

$\|x_i\|$ the Euclidean norm of $x_i$, and $\langle x_i, x_i \rangle = \|x_i\|^2$;

$\Phi(x_i)$ the instance in the kernel space via a feature mapping $\Phi$;

$\kappa(x_i, x_j)$ the kernel function. In this work, only the Gaussian RBF kernel is used;

$\sigma$ the parameter, namely the width of features, in the Gaussian RBF kernel, and the superscript star on it indicates an optimal value;

$W, B$ the within-class separability and the between-class separability, respectively;

$W', B'$ the within-class mean distance and the between-class mean distance, respectively;

$T'$ the total mean distance;

$N_i$ the number of instances in the $i$th class, and $1 \leq i \leq L$;

$N$ the total number of instances in $X$, and $N = \sum_{i=1}^{L} N_i$;

$x_i$ the instance of the $i$th class;

$J$ the objective function to measure class separability;

$(\cdot)^T$ the transpose operation of a vector or a matrix;

$\omega$ the two-dimensional weight vector for $W$ and $B$ in $J$, $\omega = [\omega_W, \omega_B]^T$;

$\text{Sign}(\cdot)$ the sign function that extracts the positive or negative sign of a real number;

$\text{Avg}(\cdot)$ the average function generating the arithmetic mean of all elements in a matrix;

$C$ the regularization parameter in SVM.

Exhaustive search for parameter selection of $\sigma$ is intractable since the definition domain of $\sigma$ ranges from zero to infinite. Grid search is an intuitive and simple way. By defining a finite set, grid search evaluates every possible solution (namely node) in the set by a criterion. The node that has the highest score on the criterion is selected as the optimal value of $\sigma$. The strategy of grid search is adopted in [10], and the classification accuracy of SVM is commonly used as the selection criterion. Grid search has two drawbacks; (1) It is time-consuming because it evaluates all the nodes in the set, and CPU time increase exponentially with the number of nodes in the set; (2) It cannot find
ANALYTICAL PARAMETER SELECTION OF GAUSSIAN SVM

the optimal \( \sigma \) if the set is improperly defined. This may happen due to lacking of prior knowledge.

Intelligent optimization methods such as genetic algorithm [11, 12], simulated annealing algorithm [13], particle swarm optimization algorithm [14-16], and gradient descent algorithm [17] have been used to select the optimal value of \( \sigma \). Classification accuracy is usually considered the objective function. However, classification accuracy of SVM does not depend only on \( \sigma \), while it could be affected by other parameters, e.g. the regularization parameter. Li et al. [18] proposed a parameter selection method for \( \sigma \) from another viewpoint (namely Li’s method). Li’s method searched for the optimal value of \( \sigma \) from the perspective of the Gaussian RBF kernel space that intrinsically results from the parameter \( \sigma \). Li’s method finds the optimal \( \sigma \) using the gradient search method. The reviewed parameter selection methods by using intelligent optimization search algorithms could take less computation time than that in grid search. However, they are at the cost of increasing the complexity of selection algorithms, which is likely the reason of why the parameter \( \sigma \) is often specified by a default value in many applications.

To improve efficiency of the selection process, in the present work, an analytical algorithm that is simple but efficient is proposed to find a good value of \( \sigma \). We define the objective function of class separability by introducing both within-class separability and between-class separability. This measure of class separability, in fact, is a function with respect to the parameter \( \sigma \). The optimal \( \sigma \) is thus defined as the one maximizing the class separability, i.e. the maximizer of the objective function. Since the maximizer can be analytically derived, the proposed method avoids the optimization search process, and thus computation load for parameter selection is significantly improved. Experimental results demonstrate that the proposed method is fast and robust for the Gaussian SVM.

The rest of the paper is organized as follows. Section 2 introduces the theoretical basis of support vector machine and the Gaussian RBF kernel. Li’s method is briefly described in this section. The proposed method is presented in Section 3. In Section 4, the proposed method and three reported methods are compared with each other on seventeen real-world datasets in terms of computation time, classification accuracy, and generalization ability. Pros and cons of these methods are discussed. Finally, conclusions are provided in Section 5.

2. REPORTED WORK

2.1 Support Vector Machine

Kernel method is a set of approaches that maps data from the feature space into the kernel space without knowing the mapping function \( \Phi \) explicitly. Kernel method enables SVM to find a hyper-plane in the kernel space, and thus achieve non-linear separation in the feature space. Kernel method is implemented by kernel functions that define inner product spaces as follows:

\[
\kappa(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle.
\]  

(1)

Many kernel functions have been developed according to Hilbert-Schmidt theory and Mercer condition, such as the linear kernel, the polynomial kernel, and the Gaussian
RBF kernel. Powered by a proper kernel, SVM is enabled to deal with not only linearly separable problems (e.g. by the linear kernel), but also linearly non-separable problems (e.g. by the Gaussian RBF kernel).

Next, we introduce SVM using a binary classification problem as an example. Given a training dataset $U$ containing $N$ instance-label pairs $(x_i, y_i)$, where $y_i \in \{+1, -1\}$ represents labels of the two classes. SVM seeks an optimally separable hyper-plane $f(x) = 0$ in the kernel space by maximizing the margin width between $f(x) = \pm 1$, where $f(x) = \mathbf{w}^T \Phi(x) + b$, $\mathbf{w}$ is a weight vector and $b$ is a scalar.

The margin width equals to

$$\left| (f(\Phi(x)) - 1) - (f(\Phi(x)) + 1) \right| \|\mathbf{w}\| = \frac{2}{\|\mathbf{w}\|}$$

(2)

The problem of maximizing the margin width defined in Eq. (2) is equivalent to the following optimization problem:

$$\mathbf{w}^*, b^* = \arg\min_{\mathbf{w}, b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\}$$

subject to $y_i \cdot f(\Phi(x_i)) \geq 1; \; \mathbf{w} \in \mathbb{R}^n; \; i = 1, 2, \ldots, N.$

(3)

The optimization problem of Eq. (3) is further transformed to the following equivalent dual problem by the Lagrange multiplier method [19]:

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} \mathcal{L}(\mathbf{a}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \kappa(x_i, x_j)$$

subject to $\sum_{i=1}^{N} (y_i \alpha_i) = 0; \; \alpha_i \geq 0; \; \mathbf{a} = \{\alpha_i\}^N; \; i = 1, 2, \ldots, N.$

(4)

After obtaining $\mathbf{a}^*$ by solving Eq. (4), solutions of Eq. (3) are expressed as

$$\mathbf{w}^* = \sum_{i=1}^{N} [\alpha_i^* y_i \Phi(x_i)] = \sum_{i=1}^{p} [\alpha_i^* y_i \Phi(x_i)]$$

$$b^* = \frac{1}{p} \sum_{i=1}^{p} [y_i - \alpha_i^* y_i \kappa(x_i, x_i)]$$

(5)

where $\alpha_i^*$ is the Lagrange multiplier, $i \in \{t: \alpha_i^* > 0\}$, and $p$ is the total number of elements in the set of $\{t: \alpha_i^* > 0\}$. Since $\alpha_i^* > 0$ for all support vectors and $\alpha_i^* = 0$ for the rest non-support vectors, $p$ is actually the number of support vectors.

The decision function of SVM is formed by

$$\hat{y} = f(x) = \text{Sign}\left( \sum_{i=1}^{p} [\alpha_i^* y_i \kappa(x_i, x)] + b^* \right).$$

(6)

In most practical cases, instances in the kernel space may still linearly non-separable. No solution could be found in Eq. (4). The so-called slack variable $\xi$ is hence intro-
duced into Eq. (3) to address this issue. The optimization problem of SVM turns to be
\[
\begin{align*}
\begin{pmatrix} w^*, b^* \end{pmatrix} &= \arg \min_{w, b} \left( \frac{1}{2} ||w||^2 + C \sum_{i=1}^{N} \xi_i \right) \\
\text{subject to} \quad & y_i \cdot f(\Phi(x_i)) \geq 1 - \xi_i; \quad \xi_i \geq 0; \quad C > 0; \quad w \in \mathbb{R}^n; \quad i = 1, 2, \ldots, N,
\end{align*}
\]  
(7)

where \( C \) is the regularization parameter.

Technically, the SVM model given in Eq. (3) with no slack variables is termed the hard-margin SVM, and the SVM model given in Eq. (7) involving slack variables is called the soft-margin SVM. The parameter \( C \) in the soft-margin SVM is also crucial to prediction performance [20], while we simply do a line search for the best \( C \) in this paper.

2.2 The Gaussian RBF Kernel

The previous section indicates that SVM training depends on the dot product in Eq. (1). Gramian matrix (also known as kernel matrix) [21] is such a matrix that contains all the dot product values of a training subset. That is, all information that SVM can learn about training instances is included in the Gramian matrix together with the label information. Given a dataset \( U \) and a kernel function, the Gramian matrix is expressed as

\[
G = \begin{bmatrix}
\kappa(x_1^{(1)}, x_1^{(i)}) & \cdots & \kappa(x_1^{(1)}, x_1^{(L)}) \\
\vdots & \ddots & \vdots \\
\kappa(x_1^{(L)}, x_1^{(1)}) & \cdots & \kappa(x_1^{(L)}, x_1^{(L)}) \\
\end{bmatrix}, \quad
K_y = \begin{bmatrix}
\kappa(x_1^{(i)}, x_1^{(1)}) & \cdots & \kappa(x_1^{(i)}, x_1^{(i)}) \\
\vdots & \ddots & \vdots \\
\kappa(x_1^{(i)}, x_1^{(L)}) & \cdots & \kappa(x_1^{(i)}, x_1^{(L)}) \\
\end{bmatrix}, \quad
K = \begin{bmatrix}
K_{11} & \cdots & K_{1L} \\
\vdots & \ddots & \vdots \\
K_{L1} & \cdots & K_{LL} \\
\end{bmatrix}, \quad
D = \begin{bmatrix}
K_{11}' & \cdots & K_{1L}' \\
\vdots & \ddots & \vdots \\
K_{L1}' & \cdots & K_{LL}' \\
\end{bmatrix}
\]  
(8)

where \( G^T = G \), \( K_{y} = K_{y}^T \), \( K_{y} = K_{y} \), \( i \) and \( j = 1, 2, \ldots, L \).

The Gramian matrix of the Gaussian RBF kernel is expressed as

\[
G = \begin{bmatrix}
K_{11} & \cdots & K_{1L} \\
\vdots & \ddots & \vdots \\
K_{L1} & \cdots & K_{LL} \\
\end{bmatrix} = \exp(-\frac{1}{2\sigma^2} D), \quad
K_y = \begin{bmatrix}
\|x_1^{(1)} - x_1^{(i)}\| & \cdots & \|x_1^{(1)} - x_1^{(L)}\| \\
\vdots & \ddots & \vdots \\
\|x_1^{(L)} - x_1^{(1)}\| & \cdots & \|x_1^{(L)} - x_1^{(L)}\| \\
\end{bmatrix}, \quad
D = \begin{bmatrix}
K_{11}' & \cdots & K_{1L}' \\
\vdots & \ddots & \vdots \\
K_{L1}' & \cdots & K_{LL}' \\
\end{bmatrix}, \quad
\]  
(11)
where $K_{ij} = \exp\left(-\frac{1}{2\sigma^2}K'_{ij}\right)$, $K'_{ij} = K'_{ji}$, $K'_{ij} = K'_{ji}$, $i$ and $j = 1, 2, \ldots, L$. $D$ is known as the Euclidean distance matrix. The Gramian matrix is related to both $\sigma$ and $D$. Since $D$ is fixed for a dataset, the only adjustable parameter is $\sigma$.

For two arbitrary instances, say $x_i$ and $x_j$, the distance and the angle are two measures of their relationship. Because in the Gaussian RBF kernel space the norm of any instance in the Gaussian RBF kernel space is equal to one [22], the two basic metrics of distance and angle are computed by

$$\left\|\Phi(x_i) - \Phi(x_j)\right\|^2 = 2 - 2\exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right), \quad (12)$$

$$\cos \theta(\Phi(x_i), \Phi(x_j)) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right). \quad (13)$$

### 2.3 Class Separability

Class separability is a classical concept for describing how instances scatter in the feature space. Class separability considers the following two principles. **Principle I**: Instances from the same class should be as similar as possible; **Principle II**: Instances from different classes should be as different as possible.

The within-class separability and the between-class separability are usually employed to measure how these two principles are followed. In Li’s method, the within-class separability ($W$) and the between-class separability ($B$) are respectively estimated by

$$W = 1 - \text{Avg} \left[ \begin{array}{ccc} K_{11} & \cdots & K_{1L} \\ \vdots & \ddots & \vdots \\ K_{L1} & \cdots & K_{LL} \end{array} \right] = 1 - \frac{1}{\sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=1}^{N} K(x_i^j, x_k^j)}, \quad (14)$$

$$B = 1 - \text{Avg} \left[ \begin{array}{ccc} K_{12} & \cdots & K_{1L} \\ \vdots & \ddots & \vdots \\ K_{L1} & \cdots & K_{LL} \end{array} \right] = 1 - \frac{1}{\sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{i'=1}^{L} \sum_{j'=1}^{N} K(x_i^j, x_{i'}^{j'})}. \quad (15)$$

The optimization problem of minimizing $W$ and simultaneously maximizing $B$ is a multi-objective optimization problem. Li’s method combines $W$ and $B$ linearly into a single aggregate objective function (AOF) by assigning an equal weight to each of them.

The objective function of class separability becomes

$$J(\sigma) = 1 - W + B. \quad (16)$$

By this definition, parameter selection turns to be a one-dimensional optimization problem. And the optimal $\sigma$ is found by gradient search method. However, the optimization process leads the selection process complex and time-consuming.
3. PROPOSED ANALYTICAL METHOD

In this section, we first define two scalars based on distance similarity to estimate $W$ and $B$ in the feature space. Eq. (12) shows the relationship between the distance similarity in the feature space and that in the kernel space. In light of this relationship, two corresponding scalars are obtained to estimate $W$ and $B$ in the kernel space. The optimal $\sigma$ is defined as the one that can minimize $W$ and maximize $B$ simultaneously in the kernel space. In the following derivation, datasets are assumed to be Gaussian distributed so that the mean distance can be used to estimate the class separability in a right way [23].

In the feature space, the within-class mean distance ($W'$), the between-class mean distance ($B'$), and the total mean distance ($T'$) are respectively defined as follows:

$$W' = \text{Avg} \left( \left[ \begin{array}{c} K_{i1} \\ \vdots \\ K_{iL} \end{array} \right] \right) = \frac{1}{L} \sum_{i=1}^{L} \sum_{j=1}^{N_i} \sum_{k=1}^{N} \| x^{(i)} - x^{(j)} \|^2,$$

$$B' = \text{Avg} \left( \left[ \begin{array}{c} K_{i2} \\ \vdots \\ K_{ik} \\ \vdots \\ K'_{(L-1)L} \end{array} \right] \right) = \frac{1}{L} \sum_{i=1}^{L} \sum_{j=1}^{L} \sum_{i=1}^{N_j} \sum_{k=1}^{N} \| x^{(i)} - x^{(j)} \|^2,$$

$$T' = \text{Avg} (D) = \frac{1}{N^2} \sum_{i=1}^{L} \sum_{j=1}^{L} \sum_{i=1}^{N_j} \sum_{k=1}^{N} \| x^{(i)} - x^{(j)} \|^2.$$

$W'$, $B'$, and $T'$ have the following relationship:

$$T' = \left( \frac{\sum_{i=1}^{L} N_i^2}{N^2} \right) W' + \left( 1 - \frac{\sum_{i=1}^{L} N_i^2}{N^2} \right) B'.$$

Distance similarity (usually using Euclidean distance) is a popular measure to estimate the within-class separability and the between-class separability. Under the Gaussian assumption, $W'$ and $B'$ are used to estimate the within-class separability and the between-class separability, respectively. In light of Eq. (12), $W$ and $B$ in the kernel space can be respectively estimated by

$$W = 2 - 2 \exp\left(-\frac{1}{2\sigma^2} W'\right),$$

$$B = 2 - 2 \exp\left(-\frac{1}{2\sigma^2} B'\right).$$

The objective function of class separability is established by

$$J(\sigma) = \omega W \begin{bmatrix} -W \\ B \end{bmatrix} = \omega_T \left( 2 \exp\left(-\frac{1}{2\sigma^2} W'\right) - 2 \right) + \omega_B \left( 2 - 2 \exp\left(-\frac{1}{2\sigma^2} B'\right) \right),$$

where $\omega_T$ and $\omega_B$ are the weights for the within-class and between-class separability, respectively.
where \( \omega, \lambda = [\omega_W, \omega_B]^T \), is the weight vector with a constraint of \( \omega_W + \omega_B = 1 \). The selection of \( \omega \) is problem-dependent. A larger \( \omega_W \) treats the within-class separability as the more important separability measure than the between-class separability. If the between-class separability needs to be emphasized, \( \omega_B \) becomes large.

In this paper, we simply define “separable” by the two scalars: \( W' \) and \( B' \). And we consider cases of \( W' < B' \) to be separable and the other case to be non-separable. The proposed method considers only the former case. Note that this definition does not mean that the separable dataset can be definitely 100% correctly classified if \( W' < B' \).

By the definition in Eq. (23), a large value of the class separability means a small within-class separability but a large between-class separability. The optimal \( \sigma \) can be defined as the one that can maximize the class separability, i.e. the maximizer of the twice differentiable objective function. The maximizer is obtained if the first derivative of \( J(\sigma) \) is equal to zero and the corresponding second derivative of \( J(\sigma) \) is negative. In the following, we derive the maximizer of the objective function in Eq. (23), i.e. the optimal \( \sigma \).

(1) Calculate the first derivative and the second derivative of \( J(\sigma) \).

\[
\frac{dJ(\sigma)}{d\sigma} = \frac{d}{d\sigma} \left[ \omega_W \left( 2 \exp(-\frac{1}{2\sigma^2}W') - 2 \right) + \omega_B \left( 2 - 2 \exp(-\frac{1}{2\sigma^2}B') \right) \right] \\
= \left( 2\omega_W W' \exp(-\frac{1}{2\sigma^2}W') - 2\omega_B B' \exp(-\frac{1}{2\sigma^2}B') \right) \sigma^{-3} \quad (24)
\]

\[
\frac{d^2J(\sigma)}{d\sigma^2} = \frac{d}{d\sigma} \left[ \left( 2\omega_W W' \exp(-\frac{1}{2\sigma^2}W') - 2\omega_B B' \exp(-\frac{1}{2\sigma^2}B') \right) \sigma^{-3} \right] \\
= \left( 2\omega_W W'^2 \exp(-\frac{1}{2\sigma^2}W') - 2\omega_B B'^2 \exp(-\frac{1}{2\sigma^2}B') \right) \sigma^{-6} \\
- \left( 6\omega_W W' \exp(-\frac{1}{2\sigma^2}W') - 6\omega_B B' \exp(-\frac{1}{2\sigma^2}B') \right) \sigma^{-4} \quad (25)
\]

(2) Let the first derivative in Eq. (24) equal to zero, and we get all stationary points.

\[
\frac{dJ(\sigma)}{d\sigma} = 0 \Leftrightarrow \left( 2\omega_W W' \exp(-\frac{1}{2\sigma^2}W') - 2\omega_B B' \exp(-\frac{1}{2\sigma^2}B') \right) \sigma^{-3} = 0 \\
\Leftrightarrow \sigma^* = \frac{B' - W'}{2 \times \log \left( \frac{\omega_B B' / \omega_W W'}{\omega_B B'} \right)} \quad (26)
\]

(3) Substitute \( \sigma \) in Eq. (25) by the stationary point obtained in Eq. (26), and test whether the second derivative is less than zero.

\[
\frac{d^2J(\sigma)}{d\sigma^2} < 0 \Leftrightarrow \left( 2\omega_W W'^2 \exp(-\frac{1}{2\sigma^2}W') - 2\omega_B B'^2 \exp(-\frac{1}{2\sigma^2}B') \right) \sigma^{-6} \\
- \left( 6\omega_W W' \exp(-\frac{1}{2\sigma^2}W') - 6\omega_B B' \exp(-\frac{1}{2\sigma^2}B') \right) \sigma^{-4} < 0
\]
ANALYTICAL PARAMETER SELECTION OF GAUSSIAN SVM

\[ \exp(\frac{B' - W'}{2\sigma^2}) < \omega_B (B' - 3\sigma^2) \]

Substituting \( \sigma \) by \( \sigma^* \) in Eq. (26), we have

\[ \frac{d^2 J(\sigma)}{d\sigma^2} \bigg|_{\sigma=\sigma^*} < 0 \Leftrightarrow \omega_B (W' - 3\sigma^*) < \omega_B (B' - 3\sigma^*) \]

\[ \Leftrightarrow W' < B' \] \( (27) \)

As stated earlier, the datasets are assumed Gaussian distributed and separable, that is \( W' < B' \). It makes Eq. (27) hold, and thus the stationary point in Eq. (26) is the maximizer and also the optimal \( \sigma \) we are looking for. According to Eq. (20), the optimal \( \sigma \) in Eq. (26) can be expressed as any two combinations of \( W', B', \) and \( T' \).

The proposed method is further interpreted as follows. From Eq. (10), the Gramian matrix is obtained from the Euclidean matrix with a transformation. The Euclidean matrix is fixed for a dataset. The parameter \( \sigma \) is the only factor to determine this transformation. The proposed method observes the statistical class characteristics from the Euclidean matrix to determine the optimal \( \sigma \). The transformation determined by the optimal \( \sigma \) tries to make a proper transformation of the Euclidean matrix so that the class characteristics are discriminant in the corresponding Gramian matrix. In Eq. (4), SVM training depends on the Gramian matrix together and the label information. Therefore, we could reach a well-trained model of SVM with the optimal \( \sigma \).

3.1 Selection of the Weight Vector

As mentioned earlier, selection of \( \omega \) is problem-dependent. We provide two intuitive and simple options for selecting the weight vector in this section.

First of all, we have to find the constraints of \( \omega \). In Eq. (26), the denominator in the square root must be positive because of the application condition of \( W' < B' \) in Eq. (27).

Together with the constraint of \( \omega_W + \omega_B = 1 \), the constraints for two elements in \( \omega \) are given below:

\[ 0 < \omega_W < \frac{B'}{B' + W'} , \] \( (28) \)

\[ \frac{W'}{B' + W'} < \omega_B < 1. \] \( (29) \)

If we choose \( \omega_W = \omega_B = 0.5 \), it is clear that the two conditions Eqs. (28) and (29) hold. Under this selection, the optimal \( \sigma \) is calculated by

\[ \sigma^* = \sqrt{\frac{B' - W'}{2 \times \log \left( \frac{B'}{W'} \right)}}. \] \( (30) \)

If we choose \( \omega_W = W'/(W' + B') \) and \( \omega_B = B'/(W' + B') \), the two conditions Eqs. (28) and (29) are also satisfied. This selection weights the between-class separability heavier than the within-class separability. This is very often desirable as we would to see that
different classes are clearly separable. The optimal $\sigma$ is calculated by

$$\sigma^*_\sigma = \sqrt{\frac{B^r - W^r}{4 \times \log \left( B^r / W^r \right)}}.$$ (31)

4. NUMERICAL VALIDATIONS

In this section four approaches of $\sigma$ selection are compared in terms of classification accuracy, computation time, and generalization. The four approaches are described as follows. In the first approach, $\sigma$ and $C$ are specified by default values, that is, $\sigma = C = 1$. The second approach uses grid search on a two-dimensional space established by $\sigma$ and $C$. Following the suggestion in [26], the discrete set of $\{2^{-10}, 2^{-9}, \ldots, 2^{3}\}$ is used as the searching set of $\sigma$, and $\{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$ is used for that of $C$. The optimal values of $\sigma$ and $C$ are the ones that reach the highest classification accuracy. The third approach is Li’s method that finds the optimal $\sigma$ from the viewpoint of the Gaussian RBF kernel. The fourth approach is the proposed method whose code package is available online at “http://zhiliang-liu.weebly.com/uploads/2/2/3/9/22397532/sigmaselection.txt”. We test the proposed method with two specific selections of $\omega$: $\omega_1 = [0.5, 0.5]^T$; $\omega_2 = \{\frac{W^r}{W^r + B^r}, \frac{B^r}{W^r + B^r}\}$.$W$.

SVM is used as the classifier to assess the performance of the four approaches. Seventeen real-world datasets used to test the four approaches are summarized in Table 1.

<table>
<thead>
<tr>
<th>No.</th>
<th>Dataset</th>
<th>Number of Classes</th>
<th>Number of Features</th>
<th>Number of Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Parkinson</td>
<td>2</td>
<td>22</td>
<td>195</td>
</tr>
<tr>
<td>2</td>
<td>Ionosphere</td>
<td>2</td>
<td>34</td>
<td>351</td>
</tr>
<tr>
<td>3</td>
<td>Sonar</td>
<td>2</td>
<td>60</td>
<td>208</td>
</tr>
<tr>
<td>4</td>
<td>Wine</td>
<td>3</td>
<td>13</td>
<td>178</td>
</tr>
<tr>
<td>5</td>
<td>Haberman’s Survival</td>
<td>2</td>
<td>3</td>
<td>306</td>
</tr>
<tr>
<td>6</td>
<td>Iris</td>
<td>3</td>
<td>4</td>
<td>150</td>
</tr>
<tr>
<td>7</td>
<td>Blood Transfusion</td>
<td>2</td>
<td>4</td>
<td>748</td>
</tr>
<tr>
<td>8</td>
<td>Diabetes</td>
<td>2</td>
<td>8</td>
<td>768</td>
</tr>
<tr>
<td>9</td>
<td>Libras Movement</td>
<td>15</td>
<td>90</td>
<td>360</td>
</tr>
<tr>
<td>10</td>
<td>Glass Identification</td>
<td>6</td>
<td>9</td>
<td>214</td>
</tr>
<tr>
<td>11</td>
<td>Handwritten Numeral$^{(1)}$</td>
<td>2</td>
<td>649</td>
<td>400</td>
</tr>
<tr>
<td>12</td>
<td>Vehicle Silhouettes</td>
<td>4</td>
<td>18</td>
<td>846</td>
</tr>
<tr>
<td>13</td>
<td>Breast Cancer</td>
<td>6</td>
<td>9</td>
<td>106</td>
</tr>
<tr>
<td>14</td>
<td>Pitting Damage</td>
<td>4</td>
<td>136</td>
<td>128</td>
</tr>
<tr>
<td>15</td>
<td>Mnist$^{(2)}$</td>
<td>2</td>
<td>784</td>
<td>13966</td>
</tr>
<tr>
<td>16</td>
<td>Ijcnn1$^{(2)}$</td>
<td>2</td>
<td>22</td>
<td>141691</td>
</tr>
<tr>
<td>17</td>
<td>Fourclass</td>
<td>2</td>
<td>2</td>
<td>862</td>
</tr>
</tbody>
</table>

Note: (1) The Handwritten Numeral 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. We only use 400 instances of two classes: ‘0’ and ‘1’. (2) The Ijcnn1 dataset consists of three sub-datasets: a training sub-dataset (35000 instances), a validation sub-dataset (14990 instances), and a test sub-dataset (91701 instances).
The first thirteen benchmark datasets are from the University of California Irvine (UCI) repository [24]. The Pitting Damage dataset is from an experiment in the Reliability Research Lab at the University of Alberta. The experiment aims to investigate advanced methods that can identify four pitting levels (classes), which are the baseline level, the slight level, the moderate level, and the severe level, on planet gears in a planetary gearbox test rig [25]. The last three datasets are from LIBSVM data that can be found at “http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets”.

Three measures are used to evaluate the performance of the four approaches. The first measure is classification accuracy defined as \( \frac{N_c}{N_c + N_f} \times 100\% \), where \( N_c \) is the number of instances that are correctly classified, and \( N_f \) is the number of those falsely classified. Accuracy is the most important measure. In our experiments, training accuracy and test accuracy are estimated on the training subset and the test subset, respectively.

The training accuracy is a reference to check over-fitting or under-fitting problems by comparing it with the test accuracy. If the training accuracy is high while the test accuracy is much lower, it implies that an over-fitting problem occurs. If both the test accuracy and the training accuracy are very low, an under-fitting problem occurs. The second measure considers the CPU time required in each approach for the same goal of \( \sigma \) selection. The third measure helps to check the consistency or generalization of classification accuracy and CPU time among the tested seventeen real-world datasets.

Since the parameter \( C \) affects classification accuracy, selection of \( C \) is necessary for a proper evaluation of the four approaches. \( C \) is specified by a default value in the first approach. The second approach utilizes grid search for \( C \) selection together with \( \sigma \) selection. No additional \( C \) selection is required. For Li’s method and the proposed method, \( C \) selection is conducted after \( \sigma \) selection by a line search in the set of \( \{2^{-10}, 2^{-9}, \ldots, 2^{10}\} \). The selected values of \( \sigma \) and \( C \) are summarized in Table 2 while CPU time elapsed on selecting \( \sigma \) and \( C \) in each method is summarized in Table 3. In grid search, CPU time is considered together on \( \sigma \) selection and \( C \) selection since they are selected simultaneously.

We notice that Eq. (26) considers only the ratio of the two elements in \( \omega \), \( \omega_B / \omega_W \). If \( \omega = \omega_1 \), the ratio is one, i.e. \( \omega_B / \omega_W = 1 \). The ratio under \( \omega = \omega_2 \) is equal to \( B'/W' \). The values of \( B', W', \) and \( B'/W' \) generated in the proposed method are provided in Table 2. From Table 2, the condition of \( W' < B' \) holds for all the seventeen datasets, and thus the proposed method is applicable for all the datasets.

Once the optimal values of \( \sigma \) and \( C \) are determined, the SVM model is trained on the training subset. Thirty independent runs are executed on each dataset using each approach. In each run, the training accuracy and the test accuracy are estimated by \( K \)-fold cross-validation (\( K = 5 \)). Results are saved in Table 3.

From Table 3, we can see that in terms of classification accuracy, the training accuracy is usually higher than the test accuracy for all approaches. This means that the four approaches perform well on empirical risk minimization in SVM. We have to check generalization ability of the four approaches from the perspective of test accuracy. The first approach using default values of \( C \) and \( \sigma \) works worst among the four approaches. That is, the first approach usually has low test accuracy. Efforts made by the other approaches can significantly improve test accuracy for most of the datasets. Test accuracy of the first approach varies a lot with datasets, so it shows bad generalization ability. The first approach is comparable with other approaches only if the optimal \( \sigma \) is close to the
Table 2. Selected parameters for the seventeen datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Approach</th>
<th>$\sigma$</th>
<th>$C$</th>
<th>$B'$</th>
<th>$W'$</th>
<th>$B'/W'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parkinson</td>
<td>Default</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>2.00</td>
<td>$2^{1}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>3.61</td>
<td>$2^{1}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_1$</td>
<td>4.71</td>
<td>$2^{4}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_2$</td>
<td>3.33</td>
<td>$2^{1}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parkinson</td>
<td>Default</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>4.00</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>3.99</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_1$</td>
<td>5.75</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_2$</td>
<td>4.06</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ionosphere</td>
<td>Default</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>4.00</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>3.99</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_1$</td>
<td>7.73</td>
<td>$2^{4}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_2$</td>
<td>5.46</td>
<td>$2^{4}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sonar</td>
<td>Default</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>4.00</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>5.88</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_1$</td>
<td>9.71</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_2$</td>
<td>5.46</td>
<td>$2^{4}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wine</td>
<td>Default</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>4.00</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>3.03</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_1$</td>
<td>3.34</td>
<td>$2^{4}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_2$</td>
<td>2.36</td>
<td>$2^{10}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Haberman’s Survival</td>
<td>Default</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>8.00</td>
<td>$2^{10}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>8.59</td>
<td>$2^{3}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_1$</td>
<td>13.44</td>
<td>$2^{3}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_2$</td>
<td>9.51</td>
<td>$2^{5}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iris</td>
<td>Default</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>4.00</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>1.42</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_1$</td>
<td>1.64</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_2$</td>
<td>1.16</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blood Transfusion</td>
<td>Default</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>0.02</td>
<td>$2^{5}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>2.18</td>
<td>$2^{1}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_1$</td>
<td>2.03</td>
<td>$2^{1}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_2$</td>
<td>1.44</td>
<td>$2^{1}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diabetes</td>
<td>Default</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>8.00</td>
<td>$2^{4}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>2.39</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_1$</td>
<td>2.84</td>
<td>$2^{4}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_2$</td>
<td>2.80</td>
<td>$2^{1}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Libras Movement</td>
<td>Default</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>4.00</td>
<td>$2^{1}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>6.53</td>
<td>$2^{4}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_1$</td>
<td>8.48</td>
<td>$2^{4}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The proposed method $\omega = \omega_2$</td>
<td>5.99</td>
<td>$2^{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glass Identification</td>
<td>Default</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>1.00</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>2.14</td>
<td>$2^{0}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 2. (Cont’d) Selected parameters for the seventeen datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>$\omega = \omega_1$</th>
<th>$\omega = \omega_2$</th>
<th>$\sigma$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handwritten</td>
<td>Default</td>
<td>1.00</td>
<td>1.00</td>
<td>1636.9</td>
<td>952.57</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>8.00</td>
<td>2$^{10}$</td>
<td>37.42</td>
<td>31.57</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>22.86</td>
<td>2$^{10}$</td>
<td>19.72</td>
<td>8.68</td>
</tr>
<tr>
<td></td>
<td>The proposed</td>
<td>$\omega = \omega_1$</td>
<td>25.14</td>
<td>277.90</td>
<td>245.80</td>
</tr>
<tr>
<td></td>
<td>method</td>
<td>$\omega = \omega_2$</td>
<td>17.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Parkinson</td>
<td>Default</td>
<td>90.55</td>
<td>90.55</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>94.29</td>
<td>100.0</td>
<td>87.12</td>
<td>87.12</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>93.74</td>
<td>99.73</td>
<td>8.77</td>
<td>8.70</td>
</tr>
<tr>
<td></td>
<td>The proposed</td>
<td>$\omega = \omega_1$</td>
<td>93.83</td>
<td>93.83</td>
<td>0.0026</td>
</tr>
<tr>
<td></td>
<td>method</td>
<td>$\omega = \omega_2$</td>
<td>93.93</td>
<td>93.93</td>
<td>0.0028</td>
</tr>
<tr>
<td></td>
<td>Parkinson</td>
<td>Default</td>
<td>87.12</td>
<td>87.12</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>94.29</td>
<td>100.0</td>
<td>87.12</td>
<td>87.12</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>93.74</td>
<td>99.73</td>
<td>8.77</td>
<td>8.70</td>
</tr>
<tr>
<td></td>
<td>The proposed</td>
<td>$\omega = \omega_1$</td>
<td>93.83</td>
<td>93.83</td>
<td>0.0026</td>
</tr>
<tr>
<td></td>
<td>method</td>
<td>$\omega = \omega_2$</td>
<td>93.93</td>
<td>93.93</td>
<td>0.0028</td>
</tr>
<tr>
<td></td>
<td>Parkinson</td>
<td>Default</td>
<td>87.12</td>
<td>87.12</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>94.29</td>
<td>100.0</td>
<td>87.12</td>
<td>87.12</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>93.74</td>
<td>99.73</td>
<td>8.77</td>
<td>8.70</td>
</tr>
<tr>
<td></td>
<td>The proposed</td>
<td>$\omega = \omega_1$</td>
<td>93.83</td>
<td>93.83</td>
<td>0.0026</td>
</tr>
<tr>
<td></td>
<td>method</td>
<td>$\omega = \omega_2$</td>
<td>93.93</td>
<td>93.93</td>
<td>0.0028</td>
</tr>
<tr>
<td></td>
<td>Parkinson</td>
<td>Default</td>
<td>87.12</td>
<td>87.12</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>94.29</td>
<td>100.0</td>
<td>87.12</td>
<td>87.12</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>93.74</td>
<td>99.73</td>
<td>8.77</td>
<td>8.70</td>
</tr>
<tr>
<td></td>
<td>The proposed</td>
<td>$\omega = \omega_1$</td>
<td>93.83</td>
<td>93.83</td>
<td>0.0026</td>
</tr>
<tr>
<td></td>
<td>method</td>
<td>$\omega = \omega_2$</td>
<td>93.93</td>
<td>93.93</td>
<td>0.0028</td>
</tr>
<tr>
<td></td>
<td>Parkinson</td>
<td>Default</td>
<td>87.12</td>
<td>87.12</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>94.29</td>
<td>100.0</td>
<td>87.12</td>
<td>87.12</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>93.74</td>
<td>99.73</td>
<td>8.77</td>
<td>8.70</td>
</tr>
<tr>
<td></td>
<td>The proposed</td>
<td>$\omega = \omega_1$</td>
<td>93.83</td>
<td>93.83</td>
<td>0.0026</td>
</tr>
<tr>
<td></td>
<td>method</td>
<td>$\omega = \omega_2$</td>
<td>93.93</td>
<td>93.93</td>
<td>0.0028</td>
</tr>
<tr>
<td></td>
<td>Parkinson</td>
<td>Default</td>
<td>87.12</td>
<td>87.12</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>94.29</td>
<td>100.0</td>
<td>87.12</td>
<td>87.12</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>93.74</td>
<td>99.73</td>
<td>8.77</td>
<td>8.70</td>
</tr>
<tr>
<td></td>
<td>The proposed</td>
<td>$\omega = \omega_1$</td>
<td>93.83</td>
<td>93.83</td>
<td>0.0026</td>
</tr>
<tr>
<td></td>
<td>method</td>
<td>$\omega = \omega_2$</td>
<td>93.93</td>
<td>93.93</td>
<td>0.0028</td>
</tr>
</tbody>
</table>

### Table 3. Experimental results for the seventeen datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Test Accuracy</th>
<th>Training Accuracy</th>
<th>CPU Time$^{(1)}$ (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parkinson</td>
<td>Default</td>
<td>90.55</td>
<td>100.0</td>
<td>N/A$^{(3)}$</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>94.29</td>
<td>100.0</td>
<td>11.1474</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>93.74</td>
<td>99.96</td>
<td>0.0536</td>
</tr>
<tr>
<td></td>
<td>The proposed</td>
<td>$\omega = \omega_1$</td>
<td>93.83</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td>method</td>
<td>$\omega = \omega_2$</td>
<td>93.93</td>
<td>100.0</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>Default</td>
<td>87.12</td>
<td>99.73</td>
<td>N/A$^{(3)}$</td>
</tr>
</tbody>
</table>
Table 3. (Cont’d) Experimental results for the seventeen datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Grid search</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>95.45</td>
<td>97.70</td>
<td>31.3750</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>95.46</td>
<td>97.70</td>
<td>0.2409</td>
</tr>
<tr>
<td>The proposed method</td>
<td>$\omega = \omega_1$</td>
<td>95.24</td>
<td>98.93</td>
<td>0.0093</td>
</tr>
<tr>
<td></td>
<td>$\omega = \omega_2$</td>
<td>95.42</td>
<td>97.66</td>
<td>0.0101</td>
</tr>
<tr>
<td>Sonar</td>
<td>Default</td>
<td>57.58</td>
<td>100.0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>87.25</td>
<td>100.0</td>
<td>11.7400</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>87.47</td>
<td>100.0</td>
<td>0.1246</td>
</tr>
<tr>
<td>The proposed method</td>
<td>$\omega = \omega_1$</td>
<td>87.18</td>
<td>100.0</td>
<td>0.0041</td>
</tr>
<tr>
<td></td>
<td>$\omega = \omega_2$</td>
<td>87.98</td>
<td>100.0</td>
<td>0.0042</td>
</tr>
<tr>
<td>Wine</td>
<td>Default</td>
<td>91.78</td>
<td>100.0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>98.76</td>
<td>100.0</td>
<td>24.1715</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>98.05</td>
<td>100.0</td>
<td>0.0603</td>
</tr>
<tr>
<td>The proposed method</td>
<td>$\omega = \omega_1$</td>
<td>98.32</td>
<td>100.0</td>
<td>0.0018</td>
</tr>
<tr>
<td></td>
<td>$\omega = \omega_2$</td>
<td>97.75</td>
<td>100.0</td>
<td>0.0025</td>
</tr>
<tr>
<td>Haberman’s Survival</td>
<td>Default</td>
<td>67.51</td>
<td>98.41</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>71.96</td>
<td>74.54</td>
<td>17.5888</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>72.57</td>
<td>75.18</td>
<td>1.2986</td>
</tr>
<tr>
<td>The proposed method</td>
<td>$\omega = \omega_1$</td>
<td>73.53</td>
<td>75.12</td>
<td>0.0050</td>
</tr>
<tr>
<td></td>
<td>$\omega = \omega_2$</td>
<td>72.91</td>
<td>75.30</td>
<td>0.0050</td>
</tr>
<tr>
<td>Iris</td>
<td>Default</td>
<td>96.73</td>
<td>98.82</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>97.76</td>
<td>98.54</td>
<td>20.9765</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>96.76</td>
<td>98.61</td>
<td>0.0468</td>
</tr>
<tr>
<td>The proposed method</td>
<td>$\omega = \omega_1$</td>
<td>96.96</td>
<td>98.88</td>
<td>0.0015</td>
</tr>
<tr>
<td></td>
<td>$\omega = \omega_2$</td>
<td>96.67</td>
<td>98.74</td>
<td>0.0016</td>
</tr>
<tr>
<td>Blood Transfusion</td>
<td>Default</td>
<td>70.92</td>
<td>74.39</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>72.29</td>
<td>92.04</td>
<td>129.4745</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>71.26</td>
<td>73.17</td>
<td>1.8137</td>
</tr>
<tr>
<td>The proposed method</td>
<td>$\omega = \omega_1$</td>
<td>70.82</td>
<td>74.24</td>
<td>0.0271</td>
</tr>
<tr>
<td></td>
<td>$\omega = \omega_2$</td>
<td>71.22</td>
<td>74.11</td>
<td>0.0311</td>
</tr>
<tr>
<td>Diabetes</td>
<td>Default</td>
<td>72.77</td>
<td>94.16</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>75.11</td>
<td>76.77</td>
<td>139.9139</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>74.54</td>
<td>85.34</td>
<td>2.0769</td>
</tr>
<tr>
<td>The proposed method</td>
<td>$\omega = \omega_1$</td>
<td>74.41</td>
<td>79.26</td>
<td>0.0301</td>
</tr>
<tr>
<td></td>
<td>$\omega = \omega_2$</td>
<td>74.58</td>
<td>85.79</td>
<td>0.0342</td>
</tr>
<tr>
<td>Libras Movement</td>
<td>Default</td>
<td>54.31</td>
<td>100.0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>87.97</td>
<td>99.73</td>
<td>479.8261</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>87.79</td>
<td>100.0</td>
<td>0.2094</td>
</tr>
<tr>
<td>The proposed method</td>
<td>$\omega = \omega_1$</td>
<td>87.17</td>
<td>100.0</td>
<td>0.0152</td>
</tr>
<tr>
<td></td>
<td>$\omega = \omega_2$</td>
<td>87.81</td>
<td>99.75</td>
<td>0.0160</td>
</tr>
<tr>
<td>Glass Identification</td>
<td>Default</td>
<td>70.74</td>
<td>93.74</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>70.74</td>
<td>94.95</td>
<td>60.2588</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>70.78</td>
<td>87.87</td>
<td>0.0627</td>
</tr>
<tr>
<td>The proposed method</td>
<td>$\omega = \omega_1$</td>
<td>70.25</td>
<td>92.90</td>
<td>0.0023</td>
</tr>
<tr>
<td></td>
<td>$\omega = \omega_2$</td>
<td>70.95</td>
<td>92.96</td>
<td>0.0024</td>
</tr>
<tr>
<td>Handwritten Numeral</td>
<td>Default</td>
<td>50.00</td>
<td>100.0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Grid search</td>
<td>94.88</td>
<td>100.0</td>
<td>59.9100</td>
</tr>
<tr>
<td></td>
<td>Li’s method</td>
<td>99.66</td>
<td>100.0</td>
<td>1.5479</td>
</tr>
<tr>
<td>The proposed method</td>
<td>$\omega = \omega_1$</td>
<td>99.65</td>
<td>100.0</td>
<td>0.0980</td>
</tr>
<tr>
<td></td>
<td>$\omega = \omega_2$</td>
<td>99.57</td>
<td>100.0</td>
<td>0.1001</td>
</tr>
</tbody>
</table>
## Table 3. (Cont’d) Experimental results for the seventeen datasets.

| Dataset          | Default | Grid search | Li’s method | The proposed method 
|------------------|---------|-------------|-------------|----------------------
|                  |         |             |             | $\omega = \omega_1$ |
|                  |         |             |             | $\omega = \omega_2$ |
| Vehicle Silhouettes | 72.42   | 85.41       | 83.58       | 83.62                |
|                  | 99.80   | 95.36       | 99.49       | 99.52                |
|                  |         |             |             | 1.3577               |
|                  |         |             |             | 4.0449               |
|                  |         |             |             | 47.3123              |
|                  |         |             |             | 49.1333              |
| Breast Cancer    | 68.27   | 71.08       | 66.41       | 67.63                |
|                  | 87.63   | 97.04       | 82.48       | 83.46                |
|                  |         |             |             | 0.0012               |
|                  |         |             |             | 2.0589               |
|                  |         |             |             | 2.0146               |
| Pitting Damage   | 48.01   | 96.21       | 94.81       | 96.66                |
|                  | 100.0   | 100.0       | 100.0       | 100.0                |
|                  |         |             |             | 0.0031               |
|                  |         |             |             | 1.8477               |
|                  |         |             |             | 1.8717               |
| Mnist38          | 50.91   | 99.01       | 99.69       | 99.64                |
|                  | 100.0   | 100.0       | 100.0       | 100.0                |
|                  |         |             |             | 152.0985             |
|                  |         |             |             | 4.3871e3             |
|                  |         |             |             | 4.3838e3             |
| Ijcnn1           | 96.15   | 98.22       | 98.17       | 98.17                |
|                  | 100.0   | 100.0       | 100.0       | 100.0                |
|                  |         |             |             | 2.3324e3             |
|                  |         |             |             | 1.6991e3             |
|                  |         |             |             | 4.3874e4             |
|                  |         |             |             | 4.3826e3             |
| Fourclass        | 80.74   | 98.81       | 98.72       | 98.72                |
|                  | 99.61   | 100.0       | 100.0       | 100.0                |
|                  |         |             |             | 0.0217               |
|                  |         |             |             | 7.8723               |
|                  |         |             |             | 7.9140               |

Note:
1. CPU time 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).
2. Shading text indicates the relatively best values of test accuracy and CPU time for $\sigma$ selection.
3. N/A, not available. No $\sigma$ selection is executed in the first approach.

The default value of $\sigma$, e.g., the Iris dataset. Otherwise, the test accuracy of the first approach suffers severely, and even tends to over-fitting, such as the Sonar dataset, the Libras Movement dataset, the Handwritten Numeral dataset, and the Pitting Damage dataset. Therefore, it is strongly suggested to avoid performing the Gaussian SVM with a default value of $\sigma$.

Grid search in most cases works well in terms of test accuracy, which is at the cost of CPU time. The success of this approach relies on the assumption that the optimal $\sigma$ is within the search set; otherwise, the test accuracy cannot be guaranteed in a high performance. Take the Handwritten Numeral dataset for example. Without prior knowledge, the optimal $\sigma$ is still searched in the set of $\{2^{-10}, 2^{-9}, ..., 2^3\}$; however, the optimal $\sigma$ (25.14 suggested by the proposed method with $\omega = \omega_1$) is a bit far from this range. Grid
search cannot find this value and thus has a bad performance. It is worth pointing out that SVM works badly if \( \sigma \ll (B' \text{ or } W') \) or \( \sigma \gg (B' \text{ or } W') \). These two cases in the Gaussian SVM perform similarly as cases of \( \sigma \to 0 \) and \( \sigma \to \infty \), respectively. In the Handwritten Numeral dataset, the optimal \( \sigma \) selected by grid search is eight that is smaller than the mean distances, either \( B' \) (1636.9) or \( W' \) (952.57). It likely results in over-fitting. Xu and Meng [2] stated that the structure of the dataset disappears under these two extreme cases.

Li’s method is comparable to the proposed method in terms of test accuracy. These two approaches have good generalization abilities to reach high and robust test accuracy. However, in the Breast Cancer dataset, the two approaches works a little bit worse than the grid search. It is mainly because that class separability is under estimated with a small training size (106 instances of 6 classes).

In terms of CPU time, the proposed method greatly outperforms both Li’s method and grid search. It spends much less time but maintains a comparable accuracy. Grid search is the most time-consuming approach because SVM training and test processes need to repeat many times in the process of selecting the parameter \( \sigma \). Li’s method avoids SVM training and test processes and is thus effective on reducing computation burden. For example, in the Libras Movement dataset, CPU time in Li’s method is about 1/100 of that in grid search. The proposed method makes further improvement in CPU time on top of Li’s method. It spends only about 1/13 of that in Li’s method. Two reasons are stated below: (1) the proposed method reduces the time complexity of Li’s method from \( O(N^2 \times n \times \text{iteration}) \) to \( O(N^2 \times n) \); and (2) the computation time of the Euclidean distance matrix used in the proposed method (i.e. Eq. (11)) is less than that of the Gramian matrix used in Li’s method (i.e. Eq. (10)). Moreover, Li’s method is more complex than the proposed method, and an initial starting point has to be properly specified in advance.

The computation burden of grid search is heavy and strongly related to the number of classes, features, and instances, e.g. it takes about 686 seconds in the Vehicle Silhouettes dataset that has the most number of instances among the seventeen datasets. CPU time of grid search is also proportional to the searching range. So it is difficult to apply this method to large size datasets. Li’s method is also sensitive to the number of classes, features, and instances. For example, in the Diabetes dataset, the CPU time significantly increases because of the large number of instances (768); in the Handwritten Numeral dataset, CPU time increases greatly due to the large number of features (649).

From Tables 2 and 3, we observe that all approaches delivered different values of \( \sigma \) but similar performance. Taking the Iris dataset for example, the performance is certainly sigma sensitive in a large region, say (0, 100). If we focus on a smaller region, say (1, 4), the performance is relatively insensitive. Thus, the Iris dataset is sigma sensitive in (0, 100) and sigma insensitive in (1, 4). It can be explained by the so-called good region and bad region observed by Keerthi and Lin [27]. It separates the hyper parameter space into two regions: an over fitting/under fitting region and a good region which most likely has the hyper parameter set with the best classification accuracy. Therefore, if producing similar classification accuracy, the values of \( \sigma \) may be different. In particular, if a dataset has a very small good region, the proposed method can also successfully locate the optimal \( \sigma \) that has a comparably high accuracy. Figs. 1 (a) and (b) show the classification accuracy on the Fourclass dataset as a function of \( \sigma \) in a large range (i.e. (0, 100)) and in a small range (i.e. (0, 2.5)), respectively. The best achievable accuracy is 100%. If we
define the good region by an accuracy range from 99% to 100%, we observe a very tiny good sigma region from about 0.04 to about 0.60 in Fig. 1 (b). Accuracy decreases quickly if \( \sigma \) goes far from this region, e.g. the accuracy is about 80% if \( \sigma = 4 \). This phenomenon is shown in Fig. 1 (a). In this case with a very small good region, the proposed method can easily find an optimal value (0.55) that is within the so-called good region, even though the accuracy (99.54%) is slightly worse than the maximum accuracy. The best sigma value (0.64) obtained by Li’s method is very close to the good region with a slightly worse accuracy (98.81%). This example demonstrates the ability of the proposed method to provide a very good \( \sigma \) value quickly, even though the good region is very small.

![Fig. 1. Classification accuracy with respect to \( \sigma \) on the four-class dataset.](image)

**5. CONCLUSIONS**

In this paper, a fast and robust parameter selection method is proposed for the Gaussian radial basis function in SVM classification. The theoretical basis and interpretation of the analytical selection method is provided in this paper. This method evaluates \( \sigma \) from the viewpoint of class separability in the kernel space. The optimal \( \sigma \) is defined as the one with the maximum class separability. An analytical solution of \( \sigma \) is found provided that the within-class mean distance (\( W' \)) is less than the between-class mean distance (\( B' \)). In this work, two formulas are provided corresponding to two specific weight vectors: \( \omega_1 = [0.5, 0.5]^T \); \( \omega_2 = [W'/(W'+B'), B'/(W'+B')]^T \). Experimental results on the seventeen real-world datasets demonstrate that the proposed method is very fast and robust.

**ACKNOWLEDGMENT**

The authors thank three anonymous reviewers and the associate editor for their constructive comments and helpful suggestions.
REFERENCES


Zhiiliang Liu (刘志亮) received the B.S. degree in School of Electrical and Information Engineering, Southwest University for Nationalities, Chengdu, in 2006. He visited University of Alberta as a visiting scholar from 2009 to 2011. He got his Ph.D. degree on June 2013 from the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu. His research interests mainly include pattern recognition, data mining, fault diagnosis and prognosis.
Ming J. Zuo (左明健) received the Ph.D. degree in 1989 in Industrial Engineering from Iowa State University, Ames, Iowa, U.S.A. His research interests include system reliability analysis, maintenance modeling and optimization, signal processing, and fault diagnosis. He is Associate Editor of IEEE Transactions on Reliability, Department Editor of IIE Transactions (2005-2008, 2011-present), Regional Editor for North and South American region for International Journal of Strategic Engineering Asset Management, and Editorial Board Member of Reliability Engineering and System Safety, Journal of Traffic and Transportation Engineering, International Journal of Quality, Reliability and Safety Engineering, and International Journal of Performability Engineering. He is Fellow of the Institute of Industrial Engineers (IIE), Fellow of the Engineering Institute of Canada (EIC), Founding Fellow of the International Society of Engineering Asset Management (ISEAM), and Senior Member of IEEE.

Xiaomin Zhao (赵晓敏) got her Ph.D. degree from the Mechanical Engineering Department at the University of Alberta in 2012. Her research interest includes fault diagnosis, condition monitoring and preventive maintenance.

Hongbing Xu (徐红兵) received the M.S. degree in Automatic Control from Southeast University, China, in 1988, the Ph.D. degree in Automation from University of Electronic Science and Technology of China in 2000. Currently, he is a Professor with School of Automation Engineering in Electronic Science and Technology of China. His research interests are renewable power generation and intelligent information processing.