Proceedings of the Sustainable Research and Innovation Conference
JKUAT Main Campus, Kenya
6 - 7 October, 2021

Optimal Accuracy Selection for Gaussian Multiclass SVM through optimization of kernel scale and box constraint

J.N.Kamau [1*], P.K. Hinga[2] and S.I. Kamau [3]

Abstract— The best accuracy of multiclass max-win-voting SVM with Gaussian radial basis function (RBF) depends on optimal parameter selection; sigma (σ) and box constraint (C). Accuracy is the most important measure to evaluate the performance of SVM. There are training accuracy and test accuracy which 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 Gaussian radial basis function (RBF) is a widely used kernel function in SVM. The kernel parameter σ is most 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 σ 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. After optimal σ is selected, box constraint parameter is easily searched using simple iterative method. Experimental results on three real world datasets demonstrate that the proposed algorithm give best accuracy when using it for the Gaussian multiclass SVM.

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

I. 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 [3,4]. 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 $k(X_i, X_j) = \exp\left(-\frac{\|X_i - X_j\|^2}{2\sigma^2}\right)$ where σ is the only parameter named by width of features.

Liu et al [5] and Conar and Chattopadhyay [6] showed that the parameter σ is specified by a default value, *e.g.* $\sigma = 1$. However, it is reported that σ is crucial to robust performance of SVM whereas an arbitrary value of σ cannot guarantee satisfactory performance [1]. For example, if the parameter σ 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 σ 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 σ is necessary and worth to do in practice.

Exhaustive search for parameter selection of σ is intractable since the definition domain of σ 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 σ. The strategy of grid search is adopted in [7], 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 increases exponentially with the number of nodes in the set; (2) It cannot find the optimal σ if the set is improperly defined. This may happen due to lacking of prior knowledge.

Intelligent optimization methods such as genetic algorithm [8], simulated annealing algorithm [9], particle swarm optimization algorithm [10], and gradient descent algorithm [11] have been used to select the optimal value of σ. Classification accuracy is usually considered the objective function. However, classification accuracy of SVM does not depends on only σ, while it could be affected by other parameters, *e.g.* the regularization parameter. Li *et al.* [12] proposed a parameter selection method for σ from another viewpoint (namely *Li's method*). Li's method searched for the optimal value of σ from the perspective of the Gaussian RBF kernel space that intrinsically results from the parameter σ. Li's method finds the optimal σ 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, the reason of why the parameter σ 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 σ. 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 σ. The optimal σ 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, multiclass max-win-voting SVM 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 default method are compared with each other on three real-world datasets in terms of classification accuracy. Pros and cons of these methods are discussed. Finally, conclusions are provided in Section 5.

II. REPORTED WORK

A. *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 Φ 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:

$$k(X_i, X_j) = \langle \Phi(X_i), \Phi(X_j) \rangle \quad (1)$$

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 ($\mathbf{x}i$, yi), where $y_i \in \{+1, -1\}$ represents labels of the two classes. SVM seeks an optimally separable hyper-plane $f(\mathbf{x}) = 0$ in the kernel space by maximizing the margin width between $f(\mathbf{x}) = \pm 1$, where $f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\Phi(\mathbf{x}) + b$, $\mathbf{w}$ is a weight vector and b is a scalar.
The margin width equals to

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

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

$$W^*, b^* = \arg min_{w,b}\left(\frac{1}{2}\|w\|^2\right)$$
$$\text{Subject to } y_i.f\left(\Phi(x_i)\right) \geq 1; w \in \mathbb{R}^n; i = 1,2,\dots,N \quad (3)$$

The optimization problem of Eq. (3) is further transformed to the following equivalent dual problem by the Lagrange multiplier method [13]:

$$\alpha^* = \arg\max L(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\alpha_i\alpha_j y_i y_j\, k(X_i, X_j)\right]$$
$$\text{Subject to } \sum_{i=1}^{N}(y_i\alpha_i) = 0;\ \alpha_i \geq 0;\ \alpha = \{\alpha_i\}^N; i = 1,2,\dots,N. \quad (4)$$

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

$$\begin{cases} w^* = \sum_{i=1}^{N}[\alpha_i^* y_i \Phi(x_i)] = \sum_{t=1}^{p}[\alpha_t^* y_t \Phi(x_t)] \\ b^* = \frac{1}{p}\sum_{t=1}^{p}[y_t - \alpha_t^* y_t \mathrm{K}(x_t, x_t)] \end{cases} \quad (5)$$

Where $\alpha_i^*$is the Lagrange multiplier, $t = \in \{t: \alpha_t^* > 0\}$, and p is the total number of elements in the set of $\{t: \alpha_t^* > 0\}$, 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 is formed by

$$\hat{y} = f(x) = Sign\left(\sum_{t=1}^{p}[\alpha_t^* y_t \mathrm{K}(x_t, x] + b^*\right) \quad (6)$$

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 ξi is hence introduced into Eq. (3) to address this issue. The optimization problem of SVM turns to be

$$w^*, b^*, \xi_i^* = \arg\min\left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i\right) \quad (7)$$
$$\text{Subject to } y_i.f\left(\Phi(X_i)\right) \geq 1 - \xi_i;\ \xi_i \geq 0; C > 0; w \in \mathbb{R}^n; i = 1,2,\dots,N,$$

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 [14], while we simply do a line search for the best *C* in this paper.

B. Max Wins Voting (MWV) SVM

SVMs were initially intended for binary classification. Many methods have been proposed for multi-class SVMs and the prominent approach is to reduce the single multiclass problem into multiple binary classification problems [15]. Three widely used types of methods are: Winner-Takes-All (WTA) SVM, Max Wins Voting (MWV) SVM and Directed Acyclic Graph (DAG) SVM. J.Kamau [16] proved that MWV has the highest accuracy in image classification among the three methods and we select this method to optimize its accuracy.

For max wins voting (MWV), classification is done by a strategy for the one versus one method. In total one will get *C*(*C*-1)/2 binary SVMs after constructing a binary SVM for each pair of classes. Each SVM gives one vote to the winning class when applied to a new test data and the test data is labeled with the class having most labels. MWV selects the class with the smallest index if there are two identical votes. The mathematical formula is shown as follow. The *ij*th (*i* = 1,2, …, *C*-1, *j* = *i* + 1, …, *C*) individual binary SVM is trained with all data in the *i*th class with +1 label plus all data of the *j*th class with −1 label, so as to distinguish *i*th class from *j*th class. The decision function of *ij*th SVM is:

$$f_{ij}(x) = \sum_{n=1}^{N_1+N_j} y_n^{ij}\alpha_n^{ij}\, k\left(x_n^{ij}, x\right) - b_{ij}, i = 1,2,\dots,C-1, j = i+1, i+2,\dots,C \quad (8)$$

$$y_n^{ij} = \begin{cases} +1 & x_n^{ij} \in ith\ class \\ -1 & x_n^{ij} \in jth\ class \end{cases} \quad (9)$$

where N_i and N_j denotes the total number of *i*th class and *j*th class, respectively. $y_n^{ij} \in \{+1, -1\}$ depends on the class label of x_n^{ij}. If x_n^{ij} belongs to *i*th class, $y_n^{ij} = +1$; otherwise x_n^{ij} belongs to *j*th class, $y_n^{ij} = -1$. α_n^{ij} is the Lagrange coefficient; and b_{ij} is the bias term. α_n^{ij} and b_{ij} are obtained by training the *ij*th individual SVM. The output of *ij*th SVM is the sign function of its decision function, namely:

$$O_{ij}(x) = sgn\left(f_{ij}(x)\right) \quad (10)$$

if $f_{ij}(x) > 0$, then the output $O_{ij}(x)\ is + 1$, denoting x belongs to *i*th class; otherwise output is -1, denoting $x$belongs to *j*th class.

C. 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) 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:

$$G = \begin{bmatrix} \begin{bmatrix} k(X_1^{(1)}, X_1^{(1)}) \dots k(X_1^{(1)}, X_{N_1}^{(1)}) \\ \vdots \quad \ddots \quad \vdots \\ k(X_{N_1}^{(1)}, X_1^{(1)}) \dots k(X_{N_1}^{(1)}, X_{N_1}^{(1)}) \end{bmatrix} \dots \begin{bmatrix} k(X_1^{(1)}, X_1^{(L)}) \dots k(X_1^{(1)}, X_{N_L}^{(1)}) \\ \vdots \quad \ddots \quad \vdots \\ k(X_{N_L}^{(1)}, X_1^{(L)}) \dots k(X_{N_L}^{(1)}, X_{N_L}^{(1)}) \end{bmatrix} \\ \vdots \quad \ddots \quad \vdots \\ \begin{bmatrix} k(X_1^{(L)}, X_1^{(1)}) \dots k(X_1^{(1)}, X_{N_1}^{(1)}) \\ \vdots \quad \ddots \quad \vdots \\ k(X_{N_L}^{(L)}, X_1^{(1)}) \dots k(X_{N_L}^{(L)}, X_{N_1}^{(1)}) \end{bmatrix} \dots \begin{bmatrix} k(X_1^{(L)}, X_1^{(L)}) \dots k(X_1^{(L)}, X_{N_L}^{(L)}) \\ \vdots \quad \ddots \quad \vdots \\ k(X_{N_L}^{(L)}, X_1^{(L)}) \dots k(X_{N_L}^{(L)}, X_{N_L}^{(L)}) \end{bmatrix} \end{bmatrix}$$

$$G = \begin{bmatrix} K_{11} \dots K_{1L} \\ \vdots \quad \ddots \quad \vdots \\ K_{L1} \dots K_{LL} \end{bmatrix} \quad (11)$$

$$K_{ij} = \begin{bmatrix} k\left(X_1^{(i)}, X_1^{(j)}\right) \dots k\left(X_1^{(i)}, X_{N_j}^{(j)}\right) \\ \vdots \quad \ddots \quad \vdots \\ k\left(X_{N_i}^{(i)}, X_1^{(j)}\right) \dots k\left(X_{N_i}^{(i)}, X_{N_j}^{(j)}\right) \end{bmatrix} \quad (12)$$

Where $G^T = G, K_{ij}^T = K_{ij}, K_{ij} = K_{ji}, i\ and\ j = 1,2,\dots,L.$
The Gramian matrix of the Gaussian RBF kernel is expressed as

$$G = \begin{bmatrix} K_{11} \dots K_{1L} \\ \vdots \quad \ddots \quad \vdots \\ K_{L1} \dots K_{LL} \end{bmatrix} = \exp\left(-\frac{1}{2\sigma^2} D\right) \quad (13)$$

$$D = \begin{bmatrix} K'_{11} \dots K'_{1L} \\ \vdots \quad \ddots \quad \vdots \\ K'_{L1} \dots K'_{LL} \end{bmatrix}, K'_{ij} = \begin{bmatrix} \left\|X_1^{(i)} - X_1^{(j)}\right\|^2 \dots \left\|X_1^{(i)} - X_{N_j}^{(j)}\right\|^2 \\ \vdots \quad \ddots \quad \vdots \\ \left\|X_{N_i}^{(i)} - X_1^{(j)}\right\|^2 \dots \left\|X_{N_i}^{(i)} - X_{N_j}^{(j)}\right\|^2 \end{bmatrix} \quad (14)$$

Where $\mathrm{K_{ij}} = \exp\left(-\frac{1}{2\sigma^2}\mathrm{K'_{ij}}\right), \mathrm{K'^{T}_{ij}} = \mathrm{K'_{ij}}, \mathrm{K'_{ij}} = \mathrm{K'_{ji}}, \mathrm{i\ and\ j} = 1,2,\dots,\mathrm{L}.$ D is known as the euclidean distance matrix. The Gramian matrix is related to both σ and **D**. Since **D** is fixed for a dataset, the only adjustable parameter is σ. 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 [17], the two basic metrics of distance and angle are computed by

$$\left\|\Phi(X_i) - \Phi\left(X_j\right)\right\|^2 = 2 - 2\exp\left(-\frac{\left\|X_i - X_j\right\|^2}{2\sigma^2}\right) \quad (15)$$

$$cos\theta\ \left(\Phi(X_i), \Phi\left(X_j\right)\right) = \exp\left(-\frac{\left\|X_i - X_j\right\|^2}{2\sigma^2}\right) \quad (16)$$

D. 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

$$W = 1 - Avg\left(\begin{bmatrix} K_{11} & & \\ & \ddots & \\ & & K_{LL} \end{bmatrix}\right) = 1 - \frac{1}{\sum_{i=1}^{L} N_i^2} \sum_{i=1}^{L} \sum_{t=1}^{N_i} \sum_{k=1}^{N_i} K\left(X_t^{(i)}, X_k^{(i)}\right), \quad (17)$$

$$B = 1 - Avg\left(\begin{bmatrix} & K_{12} & \cdots & K_{1L} \\ & & \ddots & \\ K_{21} & & & \\ \vdots & \ddots & & K_{(L-1)L} \\ K_{L1} & \cdots & K_{L(l-1)} & \end{bmatrix}\right) = 1 - \frac{1}{\sum_{i=1}^{L} \sum_{\substack{j=1 \\ j\neq i}}^{L} N_i N_j} \sum_{i=1}^{L} \sum_{\substack{j=1 \\ j\neq 1}}^{L} \sum_{t=1}^{N_i} \sum_{k=1}^{N_j} K\left(X_t^{(i)}, X_k^{(i)}\right) \quad (18)$$

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

$$J(\sigma) = 1 - W + B \quad (19)$$

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

III. 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. (15) 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 σ 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 [18].

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:

$$W' = Avg\left(\begin{bmatrix} K'_{11} & & \\ & \ddots & \\ & & K'_{LL} \end{bmatrix}\right) = \frac{1}{\sum_{i=1}^{L} N_i^2} \sum_{i=1}^{L} \sum_{t=1}^{L} \sum_{k=1}^{N_i} \left\| X_t^{(i)} - X_k^{(i)} \right\|^2 \quad (20)$$

$$B' = Avg\left(\begin{bmatrix} K'_{12} & \cdots & K'_{1L} \\ & \ddots & \vdots \\ & & K'_{(L-1)L} \end{bmatrix}\right) = \frac{1}{\sum_{i=1}^{L} \sum_{j=i\neq 1}^{L} N_i N_j} \sum_{i=1}^{L} \sum_{j=i\neq 1}^{L} \sum_{t=1}^{N_i} \sum_{k=1}^{N_j} \left\| X_t^{(i)} - X_k^{(i)} \right\|^2 \quad (21)$$

$$T' = Avg(D) = \frac{1}{N^2} \sum_{i=1}^{L} \sum_{j=1}^{L} \sum_{t=1}^{N_i} \sum_{k=1}^{N_j} \left\| X_t^{(i)} - X_k^{(i)} \right\|^2 \quad (22)$$

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

$$T' = \left(\sum_{i=1}^{L} N_i^2 / N^2\right) W' + \left(1 - \sum_{i=1}^{L} N_i^2 / N^2\right) B' \quad (23)$$

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 betweenclass separability, respectively. In light of Eq. (15), *W* and *B* in the kernel space can be respectively estimated by

$$W = 2 - 2\exp\left(-\frac{1}{2\sigma^2} W'\right) \quad (24)$$

$$B = 2 - 2\exp\left(-\frac{1}{2\sigma^2} B'\right) \quad (25)$$

The objective function of class separability is established by

$$J(\sigma) = \omega^T \begin{bmatrix} -W \\ B \end{bmatrix} = \omega_w \left(2\exp\left(-\frac{1}{2\sigma^2} W'\right) - 2\right) + \omega_B \left(2 - 2exp\left(-\frac{1}{2\sigma^2} B'\right)\right) \quad (26)$$

Where $\boldsymbol{\omega}$, $\boldsymbol{\omega} = [\omega_W, \omega_B]^T$, is the weight vector with a constraint of $\omega_W + \omega_B = 1$. The selection of $\boldsymbol{\omega}$ is problem-dependent. A larger ω_W treats the within-class separability as the more important separability measure than the between-class separability. If the betweenclass separability needs to be emphasized, ω_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. (26), a large value of the class separability means a small within-class separability but a large between-class separability. The optimal σ 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. (26), *i.e.* the optimal σ.

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

$$\frac{dJ(\sigma)}{d\sigma} = \frac{d}{d\sigma}\left[\omega_w\left(2exp\left(-\frac{1}{2\sigma^2}W'\right)-2\right) + \omega_B\left(2-2exp\left(-\frac{1}{2\sigma^2}B'\right)\right)\right]$$
$$= \left(2\omega_w W' exp\left(-\frac{1}{2\sigma^2}W'\right) - 2\omega_B B' exp\left(-\frac{1}{2\sigma^2}B'\right)\right)\sigma^{-3} \quad (27)$$

$$\frac{d^2J(\sigma)}{d\sigma^2} = \frac{d}{d\sigma}\left[\left(2\omega_w W' exp\left(-\frac{1}{2\sigma^2}W'\right) - 2\omega_B B' exp\left(-\frac{1}{2\sigma^2}B'\right)\right)\sigma^{-3}\right]$$
$$= \left(2\omega_w W'^2 exp\left(-\frac{1}{2\sigma^2}W'\right) - 2\omega_B B'^2 exp\left(-\frac{1}{2\sigma^2}B'\right)\right)\sigma^{-6} - \left(6\omega_w W' exp\left(-\frac{1}{2\sigma^2}W'\right) - 6\omega_B B' exp\left(-\frac{1}{2\sigma^2}B'\right)\right)\sigma^{-4} \quad (28)$$

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

$$\frac{dJ(\sigma)}{d\sigma} = 0 \Leftrightarrow \left(2\omega_w W' exp\left(-\frac{1}{2\sigma^2}W'\right) - 2\omega_B B' exp\left(-\frac{1}{2\sigma^2}B'\right)\right)\sigma^{-3} = 0$$
$$\Leftrightarrow \sigma^* = \sqrt{\frac{B'-W'}{2\times log(\omega_B B'/\omega_W W')}} \quad (29)$$

(3) Substitute σ in Eq. (28) by the stationary point obtained in Eq. (29), and test whether the second derivative is less than zero.

$$\frac{d^2J(\sigma)}{d\sigma^2} < 0 \Leftrightarrow \left(2\omega_w W'^2 exp\left(-\frac{1}{2\sigma^2}W'\right) - 2\omega_B B'^2 exp\left(-\frac{1}{2\sigma^2}B'\right)\right)\sigma^{-6}$$
$$\left(6\omega_w W' exp\left(-\frac{1}{2\sigma^2}W'\right) - 6\omega_B B' exp\left(-\frac{1}{2\sigma^2}B'\right)\right)\sigma^{-4} < 0$$
$$\Leftrightarrow \omega_W W'(W'-3\sigma^2) exp\left(\frac{B'-W'}{2\sigma^2}\right) < \omega_B B'(B'-3\sigma^2)$$

Substituting σ by σ^* in Eq. (29), we have

$$\frac{d^2J(\sigma)}{d\sigma^2}|_{\sigma=\sigma^*} < 0 \Leftrightarrow \omega_W W'(W'-3\sigma^{*2})\frac{\omega_B B'}{\omega_W W'} < \omega_B B'(B'-3\sigma^{*2})$$
$$\Leftrightarrow W' < B' \quad (30)$$

As stated earlier, the datasets are assumed Gaussian distributed and separable, that is *W*' < *B*'. It makes Eq. (30) hold, and thus the stationary point in Eq. (29) is the maximizer and also the optimal σ we are looking for. According to Eq. (23), the optimal σ in Eq. (29) can be expressed as any two combinations of *W*', *B*', and *T*'.

The proposed method is further interpreted as follows. From Eq. (13), the Gramian matrix is obtained from the Euclidean matrix with a transformation. The Euclidean matrix is fixed for a dataset. The parameter σ is the only factor to determine this transformation. The proposed method observes the statistical class characteristics from the Euclidean matrix to determine the optimal σ. The transformation determined by the optimal σ 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 σ.

A. Selection of the Weight Vector

As mentioned earlier, selection of **ω** 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 **ω**. In Eq. (29), the denominator in the square root must be positive because of the application condition of *W*'< *B*' in Eq. (30).

Together with the constraint of $\omega_W + \omega_B = 1$, the constraints for two elements in **ω** are given below:

$$0 < \omega_w < \frac{B'}{B'+W'} \quad (31)$$

$$\frac{B'}{B'+W'} < \omega_b < 1 \quad (32)$$

If we choose $\omega_W = \omega_B = 0.5$, it is clear that the two conditions Eqs. (31) and (32) hold. Under this selection, the optimal σ is calculated by

$$\sigma_1^* = \sqrt{\frac{B'-W'}{2\times log(B'/W')}} \quad (33)$$

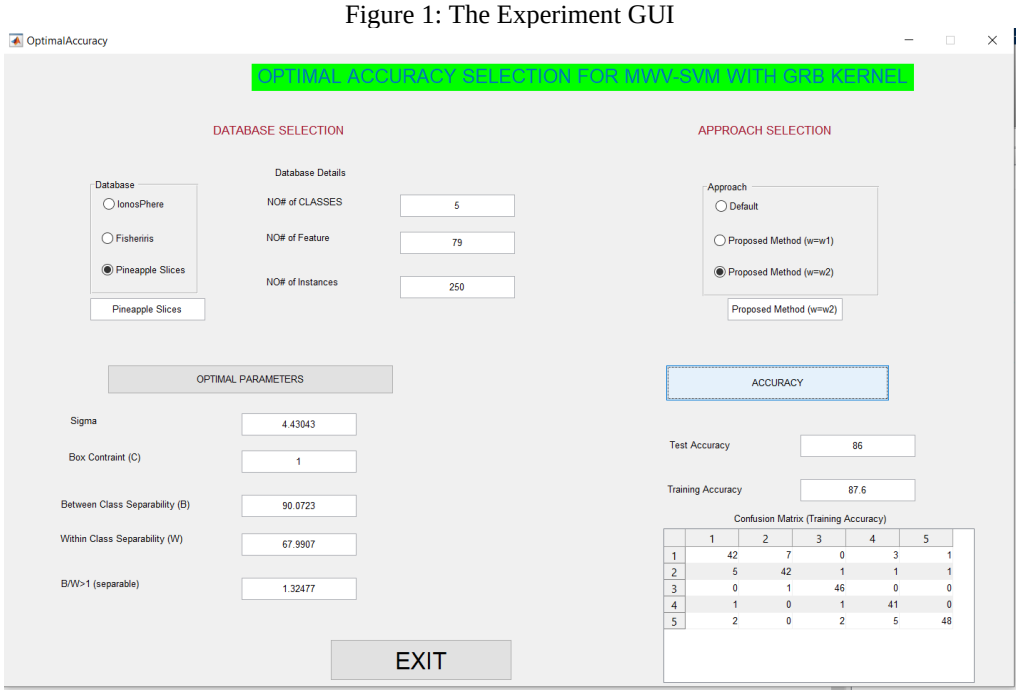
If we choose $\omega_W = W'/(W'+B')$ and $\omega_B = B'/(W'+B')$, the two conditions Eqs. (31) and (32) are also satisfied. This selection weights the between-class separability heavier than the within-class separability. This is very often desirable as we would like to see that different classes are clearly separable. The optimal σ is calculated by:

$$\sigma_2^* = \sqrt{\frac{B'-W'}{4\times log(B'/W')}} \quad (34)$$

IV. NUMERICAL VALIDATIONS

The simulation was carried out on a HP 15 laptop with core i7 1.8GHz base frequency and 8GB memory running under 64-bit Microsoft Windows 10 operating system. The algorithm was developed in-house on the Matlab 2018a (The Mathworks ©) platform. The figure 1 below is the Graphical User Interface (GUI) of the experiment.

Figure 1: The Experiment GUI



In this section two approaches of σ selection are compared in terms of classification accuracy. The two approaches are described as follows. In the first approach, σ and C are specified by default values, that is, $\sigma = C = 1$. The second approach is the proposed method. We test the proposed method with two specific selections of $\boldsymbol{\omega}$: $\boldsymbol{\omega}1 = [0.5, 0.5]^T$; $\boldsymbol{\omega}2 = [W'/(W' + B'), B'/(W' + B')]^T$.

SVM is used as the classifier to assess the performance of the two approaches. Three real-world datasets used to test the two approaches are summarized.

Table 1: Summary of the three Datasets

No	Dataset	Number of Classes	Number of Features	Number of Instances
1	Ionosphere	2	34	351
2	Fisheriris	3	4	150
3	Pineapple Slices	5	79	250

Classification accuracy is used to evaluate the performance of the two approaches. Classification accuracy is defined as $Nc/(Nc+Nf)\times100\%$, where Nc is the number of instances that are correctly classified, and Nf is the number of those falsely classified. Accuracy is the most important performance measure. Since the parameter C *(box constraint)* affects classification accuracy, selection of C is necessary for a proper evaluation of the two approaches. C is specified by a default value in the first approach. The second approach utilizes grid search for C selection together with σ selection. The selected values of σ and C are summarized in Table 2.

Table 2: Selected parameters for three datasets

Dataset	Approach		σ	C	B'	W'	B'/W'
Ionosphere	Default		1	1	78.1863	55.2614	1.41484
	The proposed Method	$\omega = \omega_1$	5.74727	10			
		$\omega = \omega_2$	4.06393	10			
Fisheriris	Default		1	1	10.8171	2.20571	4.90416
	The proposed Method	$\omega = \omega_1$	1.64556	1			
		$\omega = \omega_2$	1.16358	1			
Pineapple Slices	Default		1	1	90.0723	67.9907	1.32477
	The proposed Method	$\omega = \omega_1$	6.26557	7			
		$\omega = \omega_2$	4.43043	10			

Once the optimal values of σ and C are

determined, the SVM model is trained on the training subset. Twenty independent runs are executed on each dataset using each approach.
by *K*-fold cross-validation ($K = 5$). Results are saved in Table 3.

In each run, the training accuracy and the test accuracy are estimated

Table 3: Experimental results of the three Dataset

Dataset	Method		Test Accuracy	Training Accuracy
Ionosphere	Default		68.5714	77.1429
	The proposed Method	$\omega = \omega_1$	94.2857	94.4857
		$\omega = \omega_2$	87.1429	94.8857
Fisheriris	Default		93.3333	95.3333
	The proposed Method	$\omega = \omega_1$	93.333	96
		$\omega = \omega_2$	100	96
Pineapple Slices	Default		50	53.6
	The proposed Method	$\omega = \omega_1$	88	88.6857
		$\omega = \omega_2$	88	89.08

From Table 3, we can see that in terms of classification accuracy, the training accuracy is usually higher than the test accuracy for two approaches. This means that the two approaches perform well on empirical risk minimization in SVM. We have to check generalization ability of the two approaches from the perspective of test accuracy. The first approach using default values of C and σ works worst between the two approaches. That is, the first approach usually has low test accuracy. Efforts made by the second 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 σ is close to the default value of σ, *e.g.* the Fisheriris dataset. Otherwise, the test accuracy of the first approach suffers severely, and even tends to over-fitting, such as the ionosphere dataset. Therefore, it is strongly suggested to avoid performing the Gaussian SVM with a default value of σ.
The proposed method has good generalization abilities to reach high and robust test accuracy. However, in the pineapple slices dataset, the approach works a little bit worse but still better than default method. It is mainly because that class separability is under estimated with a small training size.

V. 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 σ from the viewpoint of class separability in the kernel space. The optimal σ is defined as the one with the maximum class separability. An analytical solution of σ 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: $\boldsymbol{\omega}1 = [0.5, 0.5]^T$; $\boldsymbol{\omega}2 = [W'/(W'+B'), B'/(W'+B')]^T$. Experimental results on the three real-world datasets demonstrate that the proposed method is very fast and robust.

VI. ACKNOWLEDGMENT

Much thank goes to my two supervisor for their constructive comments and helpful suggestions.

REFERENCES

[1] W. Wang, Z. Xu, W. Lu, and X. Zhang, "Determination of the spread parameter in the Gaussian kernel for classification and regression," Neurocomputing, Vol. 55, 2003, pp. 643-663.
[2] Z. Xu, M. Dai, and D. Meng, "Fast and efficient strategies for model selection of Gaussian support vector machine," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 39, 2009, pp. 1292-1307.
[3] J. Qu and M. J. Zuo, "Support vector machine based data processing algorithm for wear degree classification of slurry pump systems," Measurement, Vol. 43, 2010, pp. 781-791.
[4] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," Mechanical Systems and Signal Processing, Vol. 21, 2007, pp. 2560-2574.
[5] J. Qu, Z. Liu, M. J. Zuo, and H.-Z. Huang, "Feature selection for damage degree classification of planetary gearboxes using support vector machine," Journal of Mechanical Engineering Science, Vol. 225, 2011, pp. 2250-2264.
[6] P. Konar and P. Chattopadhyay, "Bearing fault detection of induction motor using wavelet and support vector machines (SVMs)," Applied Soft Computing, Vol. 11, 2011, pp. 4203-4211.
[7] A. Widodo, E. Y. Kimb, J.-D. Sonc, B.-S. Yang, A. C. C. Tan, D.-S. Gu, B.-K.Choi, and J. Mathew, "Fault diagnosis of low speed bearing based on relevance vector machine and support vector machine," Expert Systems with Applications, Vol. 36, 2009, pp. 7252-7261.

[8] M. Zhao, C. Fu, L. Ji, K. Tang, and M. Zhou, "Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes," Expert Systems with Applications, Vol. 38, 2011, pp. 5197-5204.

[9] S.-W. Lin, Z.-J. Lee, S.-C. Chen, and T.-Y. Tseng, "Parameter determination of support vector machine and feature selection using simulated annealing approach," Applied Soft Computing, Vol. 8, 2008, pp. 1505-1512.

[10] J. Peng and S. Wang, "Parameter selection of support vector machine based on chaotic particle swarm optimization algorithm," in Proceedings of World Congress on Intelligent Control and Automation, 2010, pp. 1654-1657.

[11] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," Machine Leaning, Vol. 46, 2002, pp. 131-159.

[12] C.-H. Li, C.-T. Lin, B.-C. Kuo, and H. S. Chu, "An automatic method for selecting the parameter of the RBF kernel function to support vector machines," in Proceedings of IEEE International Conference on Geoscience and Remote Sensing Symposium, 2010, pp. 836-839.

[13] E. K. P. Chong and S. H. Żak, An Introduction to Optimization, 3rd ed., John Wiley & Sons Inc., NY, 2008.

[14] E. Alpaydin, Introduction to Machine Learning, the MIT Press, MA, 2004.

[15] S. Maddipati, R. Nandigam, S. Kim and V. Venkatasubramanian, "Learning patterns in combinatorial-protein libraries by Support Vector Machines", Comput. Chem. Eng., 35, 1143–1151, 2011.

[16] J.Kamau, "Pineapple slices classification using a hybrid feature extraction technique and multiclass SVM, case study: Del Monte Kenya ltd, cannery," JKUAT Msc.Thesis, 2021

[17] Z. Liu, M. J. Zuo, and H. Xu, "A Gaussian radial basis function based feature selection algorithm," in Proceedings of IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2011, pp. 1-4.

[18] K. Krishnamoorthy, Handbook of Statistical Distributions with Applications, Chapman & Hall/CRC, London, 2006.