Parameter Optimization of RBF Kernel SVM from miniCV

Li-Chia Yeh[1] **and Chung-Chin Lu**[1]
[1]Department of Electrical Engineering, National Tsing Hua University, Hsinchu City, Hsinchu, Taiwan

Abstract— *Support vector machine (SVM) is wildly used in classification. With the sequential minimal optimization (SMO), an iterative algorithm, SVM is able to be executed in most computers. However, the testing accuracy strongly depends on the selection of model parameters. Even though cross validation is employed in parameter optimization, it still needs enough domain knowledge to hint a range for search. In this research, we propose a light weighted intuitive core validation (miniCV). This miniCV is state-of-art in deeply counting on the distribution of data in the feature space. More specifically, the essential information in miniCV comes from the iterations of SMO. The key concept in the research is to let the data generated during training perfect the trained model. Combining a devised SVM, the test result via RBF kernel on MNIST outperforms the ones with other parameter optimization method in RBF kernel SVM.*

Keywords: supervised learning, support vector machine, sequential minimal optimization, clustering, RBF kernel, spectral

1. Introduction

Support vector machine (SVM) was proposed by Vapnik [1] that binary classification can be implemented by forming a separating hyperplane with maximized margin from the hyperplane to the instances with different labeling. This hyperplane is optimized via solving a quadratic optimization problem with some preset parameters either for linear SVMs or for kernel-based SVMs. The test accuracy is diverged under different parameter setting. Finding a better parameter combination always conducts a lower error rate.

When n-fold cross validation is utilized to search for a better parameter combination, an SVM needs to be executed completely at least n times. If the training dataset is large or cross-validation is executed based on a dense parameter grid, it is a time-consuming task. Therefore, several researchers developed algorithms for optimizing the parameter(s) in (kernel-based) SVMs. Among these parameter optimization algorithms, particle swarm optimization (PSO) is a popular method in the past decade [2], [3], [4], [5]. However, most of the PSO related parameter optimization algorithms are under a parallel computing architecture. It is very hardware dependent. In [6], a BAT algorithm is proposed to optimize the SVM performance, which converts the instances into histogram like figures first and runs classification. Both methods need to transform the data to get a better SVM model.

Unlike previous research works for optimizing SVM parameters, this paper focuses on a method in perfecting the SVM model directly via variables that are already generated in the SVM training process. The rest of this paper is organized as follows. The kernel-based SVM and SMO-SVM are briefly introduced inSection 2. In order to see the impact of parameter setting on the performance of (kernel-based) SVMs, several parameter-related experiments are designed in Section 3. The results of the experiments are discussed in Section 4. With the findings in Section 4, a state-of-art light weighted cross validation for SMO-SVM is proposed in Section 5. With this new way of cross validation, the training procedure of SMO-SVM is slightly adjusted and the performance of the proposed procedure is demonstrate via classifying the MNIST dataset in Section 5.2. Finally, the findings in this paper are summarized in Section 6, together with future works to do.

2. Support Vector Machine

Support vector machine (SVM) is a binary classifier in supervised learning. Let $\mathscr{X} \in \mathbb{R}^n$ be an input space and $\mathscr{Y} = \{1, -1\}$ a label space. Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \in (\mathscr{X} \times \mathscr{Y})^m$ be a sample of size m. Frequently, a positive definite symmetric kernel $k(\cdot, \cdot)$ is utilized to map $S|_{\mathscr{X}} \triangleq \{x_1, x_2, \ldots, x_m\}$ into a finite dimensional reproducing kernel Hilbert space (RKHS) $\mathbb{H}_S$. Denote $\Phi_x \triangleq k(\cdot, x) \in \mathbb{H}_S$. Then a positive semi-definite $m \times m$ matrix is formed as $\boldsymbol{K} \triangleq (K_{rr'})_{r,r'=1}^m = (k(x_r, x'_r))_{r,r'=1}^m = (\langle \Phi_{x_{r'}}, \Phi_{x_r} \rangle_{\mathbb{H}_S})_{r,r'=1}^m$. $\boldsymbol{K}$ is symmetric and called the kernel matrix. The primal function for classifying a dataset in $\mathbb{H}_S$ is formed as

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & L(\boldsymbol{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{w}\|_{\mathbb{H}_S}^2 + C\sum_{r=1}^m \xi_r \\ \text{subject to:} \quad & 1 - \xi_r - y_r(\langle \boldsymbol{w}, \Phi_{x_r} \rangle_{\mathbb{H}_S} + b) \leqslant 0, \\ & -\xi_r \leqslant 0, \ r = [1, m], \end{aligned}$$

where ξ_r are slack variables and C is a regulator to leverage the SVM model and the training error in preventing the overfitting problem. Moreover by nonlinear optimization with Lagrangian multipliers, the primal problem is transformed to a dual problem involving only the Lagrangian multipliers $0 \leqslant \lambda_r \leqslant C$, $r = [1, m]$. In the meantime, Karush-Kuhn-Tucher (KKT) necessary conditions are deduced and needed to be satisfied in order to solve the optimization problem. Based on the value of label y_r and the value range $\mathcal{R}(\lambda_r)$ of λ_r, the instances in the training set S

Table 1
KKT CONDITION IN EACH GROUP.

Group	$\mathcal{R}(\lambda_r)$	y_r	KKT condition
I_1	[0,0]	1	$F_r + b \geqslant 0$
I_2	[0,0]	-1	$F_r + b \leqslant 0$
I_3	$[C, C]$	1	$F_r + b \leqslant 0$
I_4	$[C, C]$	-1	$F_r + b \geqslant 0$
$I_{0,+}$	$(0, C)$	1	$F_r + b = 0$
$I_{0,-}$	$(0, C)$	-1	$F_r + b = 0$

are partitioned into 6 groups, $I_{0,-}, I_{0,+}, I_1, I_2, I_3, I_4$. Define $F_r \triangleq \langle \boldsymbol{w}, \Phi_{x_r} \rangle_{\mathbb{H}_S} - y_r$. The 6 groups and their corresponding KKT conditions are listed in Table 1. Platt [7] and Keerthi [8] suggested to implement the SVM via its dual form with sequential minimization optimization (SMO). SMO is an iterative algorithm in updating two KKT violating Lagrange multipliers λ_i, λ_j in each iteration through optimizing the simplified dual problem

$$\begin{aligned} \min_{\lambda_j^{new}} \quad & \tilde{\Theta}(\lambda_j^{new}) = \frac{1}{2}(\lambda_j^{new})^2 \eta_{ij} - \lambda_j^{new} \lambda_j^{old} \eta_{ij} \\ & \qquad - y_j \lambda_j^{new} (F_i^{old} - F_j^{old}) \\ \text{subject to:} \quad & 0 \leqslant \lambda_j^{new}, \tilde{R}_{ij} - s_{ij} \lambda_j^{new} \leqslant C, \end{aligned}$$

where $\eta_{ij} = K_{ii} + K_{jj} - 2K_{ij}, s_{ij} = y_i y_j$, and $\tilde{R}_{ij} = \lambda_i^{old} + s_{ij} \lambda_j^{old}$. We call this SVM implemented with SMO algorithm as SMO-SVM in this paper. In order to indicate the sequential iteration relation, a superscript (k), $k = 0, 1, \ldots$ is attached to variables for identifying the current values in the kth iteration. After the $(k-1)$th iteration, the SMO-SVM keeps going whenever there exists a pair (i, j) of instances such that $i \in I_{0,+} \cup I_{0,-} \cup I_2 \cup I_3$, $j \in I_{0,+} \cup I_{0,-} \cup I_1 \cup I_4$, and $\Delta F_{ij}^{(k-1)} \triangleq F_i^{(k-1)} - F_j^{(k-1)} > \tau$, where a tolerance τ is set to loosen the SMO stopping criterion to $\Delta F_{ij}^{(k-1)} \leqslant \tau$. In this case, the pair (i, j) is called as a τ-violating pair [9] and we set $(i^{(k)}, j^{(k)}) = (i, j)$ and run the kth iteration of the SMO-SVM. By the gradient descent method, the un-constrained dual problem $\tilde{\Theta}(\lambda_j^{new})$ is optimized with $t \triangleq \lambda_{j^{(k)}}^{(k)} - \lambda_{j^{(k)}}^{(k-1)} = y_j \frac{\Delta F^{(k-1)}}{\eta_{i^{(k)} j^{(k)}}}$ whenever $\eta_{ij} > 0$, where $\Delta F^{(k-1)} \triangleq F_{i^{(k)}}^{(k-1)} - F_{j^{(k)}}^{(k-1)}$. Due to the boundary constraints, t sometimes is clipped. Let t' be the finial updating quantity in each iteration, where $|t'| \leqslant |t|$. Then $\lambda_{i^{(k)}}^{(k)} = \lambda_{i^{(k)}}^{(k-1)} - s_{i^{(k)} j^{(k)}} t'^{(k)}$, $\lambda_{j^{(k)}}^{(k)} = \lambda_{j^{(k)}}^{(k-1)} + t'^{(k)}$ and

$$F_r^{(k)} \quad = \quad F_r^{(k-1)} - y_{j^{(k)}} t'^{(k)} (K_{i^{(k)} r} - K_{j^{(k)} r}).$$

are updated in the kth iteration.

Next we describe the SMO-SVM algorithm used in this paper. For a fixed tolerance $0 < \tau < 1$,

a. Initialization:
- *a*.1. Set $\boldsymbol{\lambda}^{(0)} = \mathbf{0}, \boldsymbol{F}^{(0)} = -\boldsymbol{y}$
- *a*.2. Set C and τ and other necessary parameters for a kernel-based SVM.
- *a*.3. Set $I_1, I_2, I_3, I_4, I_{0+}, I_{0-}$ based on Table 1.

b. Optimization: keep iterating and updating $\boldsymbol{\lambda}^{(k)}, \boldsymbol{F}^{(k)}$ until $\Delta F_{ij}^{(k-1)} \leqslant \tau$ for all $i \in I_{0,+} \cup I_{0,-} \cup I_2 \cup I_3$ and $j \in I_{0,+} \cup I_{0,-} \cup I_1 \cup I_4$.

3. Experiments on Parameters

In this section, we will conduct experiments on the setting of parameters in the RBF kernel SMO-SVM.

3.1 Regulator C

The support vectors in a kernel-based SVM are $\{\Phi_{x_r}\ r \in I_{0,-} \cup I_{0,+} \cup I_3 \cup I_4\}$. The SMO-SVM modifies the Lagrange multipliers $\boldsymbol{\lambda}$ iteratively from $\mathbf{0}$ to obtain the best decision hyperplane. If an instance Φ_{x_r} is weighted with a large λ_r in forming the best decision hyperplane with $\boldsymbol{w} = \sum_{r=1}^{m} \lambda_r y_r \Phi_{x_r}$, a number of non-clipped updating steps t are needed to move λ_r from 0 toward the boundary C. The value of $\Delta F^{(k-1)}$ is observed to distribute over a narrow range of values so that t is basically determined by the kernel matrix only. If C is large, the number of iterations must become large. We are interested in locating a just large enough regulator C with the best test performance. Therefore, we will set the regulator C as an independent variable and the rest as the control ones to see the variation in the number of iterations, the number of support vectors (nSV), the size of $I_3 \cup I_4$ and the error rate w.r.t. different precision γ.

3.2 Precision γ in the RBF Kernel SVM

The experiment in this subsection aims to pin out the best RBF kernel among the various precision γ. Each RBF kernel SVM is preset with a proper C based on the experiment results in Subsection 3.1. The best RBF kernel is determined via the test error rate, nSV, the value of an intermediate variable defined in Subsection 3.4, and the Fisher discriminant on $\boldsymbol{F}$, where

$$\text{Fisher discriminant} = \frac{(\text{mean}(F_+) - \text{mean}(F_-))^2}{\text{var}(F_+) + \text{var}(F_-)}$$

with $F_+ \triangleq \{F_i, i \in [1, m] | y_i = 1\}$ and $F_- \triangleq \{F_i, i \in [1, m] | y_i = -1\}$. This is a linear discriminant analysis (LDA) on $\boldsymbol{F}$. Since the grouping characteristics of any (kernel-based) SVM are determined by the kernel matrix $\boldsymbol{K}$, we apply the spectral analysis on the kernel matrix for each distinct precision γ. The spectral analysis is implemented via the RBF kernel PCA and the fully connected RatioCut spectral clustering [10].

Let $\mu_{p,\gamma}^{(0)} \geqslant \mu_{p,\gamma}^{(1)} \geqslant \cdots$ be ordered eigenvalues of a kernel matrix under the RBF kernel PCA w.r.t. a precision γ. The superscript represents the order of eigenvalues. Define the ith eigengap of the RBF kernel PCA as $\delta_{p,\gamma,i} = \mu_{p,\gamma}^{(i)} - \mu_{p,\gamma}^{(i+1)}, i = 0, 1, \ldots$. Similarly, let $\mu_{s,\gamma}^{(0)} \leqslant \mu_{s,\gamma}^{(1)} \leqslant \cdots$ be ordered eigenvalues of the fully connected RatioCut spectral clustering w.r.t. a precision γ. Because $\mu_{s,\gamma}^{(0)} = 0, \ \forall \gamma$, only

$\mu^{(1)}_{s,\gamma}, \mu^{(2)}_{s,\gamma}, \ldots$ are considered. Define the ith eigengap of the fully connected RatioCut spectral clustering as $\delta_{s,\gamma,i} = \mu^{(i)}_{s,\gamma} - \mu^{(i-1)}_{s,\gamma}, i = 2, 3, \ldots$. The largest eigengap for the RBF kernel PCA or for the fully connected RatioCut spectral clustering is defined as $\delta'_{p,\gamma} \triangleq \max_{i \geqslant 0} \delta_{p,\gamma,i}$, $\delta'_{s,\gamma} \triangleq \max_{i \geqslant 2} \delta_{s,\gamma,i}$ respectively. Moreover, since the index of the largest eigengap of a spectral clustering leads to the best number of clusters in $\mathbb{H}_S$, we will evaluate the value of $\delta'_{s,\gamma}$ along with $\arg\max_{i=2,3,\ldots} \delta_{s,\gamma,i}$ in selecting the best precision γ for the RBF kernel SVM.

3.3 Tolerance τ in the RBF Kernel SVM

We sweep the value of tolerance τ from 1 to near 0 in each independent test to confirm the hypothesis that the test performance becomes invariant even though the tolerance is not small.

3.4 An Intermediate Variable in the RBF Kernel SVM

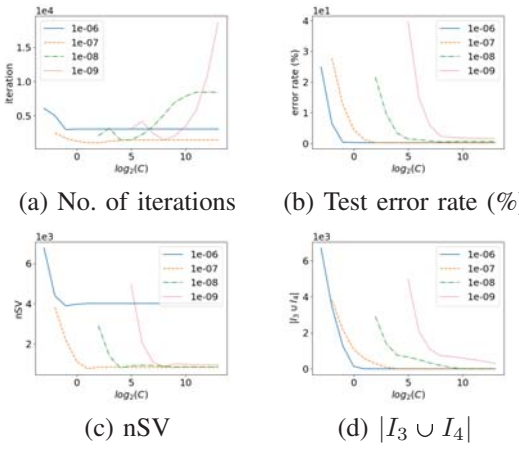
Since a kernel matrix defines the characteristics of an SVM model and a non-clipped updating step t strongly depends on it, we collect the η_{ij} value of each τ-violating pair (i, j) in first n iterations and examine the relation of these values to the test performance. Let $\mathfrak{E}_n = \{\eta_{i^{(1)}j^{(1)}}, \eta_{i^{(2)}j^{(2)}}, \ldots, \eta_{i^{(n)}j^{(n)}}\}$ represent the sequence of all η_{ij} in first n iterations. The variance $\mathrm{var}(\mathfrak{E}_n)$ of $\mathfrak{E}_n$ is chosen to study the relation between the precision γ and the error rate.

4. Results of Experiments

The instances labeled with 2 and 3 from the training and test datasets of MNIST [11] are used in performing the experiments in the last section. The size of the training dataset is 12089 and of the test dataset is 2042. We name both datasets together as MNIST-23.

4.1 Regulator C

Fig. 1 demonstrates the test results of RBF kernel SVMs with precision $\gamma = 10^{-6}, 10^{-7}, \ldots, 10^{-9}$ and a fixed preset tolerance $\tau = 0.2$. Fig. 1 reveals that there exists a $C_{\gamma,0}$ for an RBF kernel SVM with a precision γ such that when $C > C_{\gamma,0}$, the deviation of the test error rate in Fig. 1-(b) and the nSV in Fig. 1-(c) is almost none. However, when C becomes large, the number of iterations grows and even grows exponentially, as C is sufficiently large, as shown in Fig. 1-(a). Furthermore, when C is large enough, $|I_3 \cup I_4|$ is approximately zero as shown in Fig. 1-(d). Therefore, when training an SVM model, we only need to ensure that the regulator C is just large enough. Thus in Subsections 4.2-4.4, we select a regulator C which is just large enough for doing experiment. The chosen C for the kernel parameter $\gamma = 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}$ of the RBF kernel SVM are $2^{1.753}, 2^{4.17}, 2^{8.1}, 2^{10.41}$ respectively.



(a) No. of iterations (b) Test error rate (%)

(c) nSV (d) $|I_3 \cup I_4|$

Fig. 1
THE INFLUENCE OF REGULATOR C IN THE RBF KERNEL SVM ON MNIST-23.

4.2 Precision γ in the RBF Kernel SVM

In order to get a detailed trend on the testing error rate and the intermediate variable $\mathrm{var}(\mathfrak{E}_n)$ defined in Subsection 3.4, the experiments are conducted with $\gamma = 6.0, 6.1, \ldots, 10.9, 11.0$ and $\tau = 0.2$ as shown in Fig. 2. Fig. 2-(c) and 2-(d) together show the trend that when Fisher discriminant of $\boldsymbol{F}$ is higher, the less support vectors are required. The fewest nSV occurs at $\gamma = 10^{-7.2}$ marked with a square in Fig. 2-(c); the highest Fisher discriminant locates at $\gamma = 10^{-7.1}$ also marked with a green square in Fig. 2-(d). Since these two green squares locate at different γ, we also label their corresponding nSV and Fisher discriminant value with $((\cdot))$ in Fig. 2-(c) and 2-(d).

From Fig. 2, if the precision γ is selected around 10^{-7}, not only the test error rate is among the lowest, the value of Fisher discriminant is also among the highest. This high Fisher discriminant value implies that the separateness between F_+ and F_- is large so that the test error rate becomes low. Also the large separateness of $\boldsymbol{F}$ confirms that only a few support vectors are needed in forming the decision hyperplane. The curve of the test error rate in Fig. 2- (a) shows that the number of mis-classified data points in the test dataset is either 4 or 5 when the range of γ is in between 10^{-6} and $10^{-7.4}$. However 10^{-6} is not a better choice than 10^{-7}. We will explain the reason in the rest of this subsection via spectral analysis.

We study the RBF kernel matrix of the MNIST-23 training dataset with the precision $\gamma \in \{10^{-2}, 10^{-3}, \ldots, 10^{-9}\}$. As mentioning in Subsection 3.2, both the RBF kernel PCA and the fully connected RatioCut spectral clustering are employed in spectral analysis.

The results from the PCA spectral analysis in Fig. 3 show

that $\forall i \in [0, m-1]$,

$$\begin{cases} \mu_{p,\gamma}^{(i)} = 1, & \text{if } \gamma \in \{10^{-2}, 10^{-3}, 10^{-4}\}, \\ 1 \leqslant \mu_{p,\gamma}^{(i)} \leqslant 1.02, & \text{if } \gamma = 10^{-5}, \end{cases}$$

which indicates that there is hardly any significant gap between successive eigenvalues for $\gamma \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. However, from Fig. 3, there are significant gaps between successive eigenvalues for $\gamma \in \{10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}\}$, especially for the initial eigengaps, we have

$$\delta_{p,10^{-7},0} \gg \delta_{p,10^{-7},1} > \delta_{p,10^{-8},0} > \delta_{p,10^{-6},0} > \delta_{p,10^{-9},0}.$$

That implies that when $\gamma = 10^{-7}$, there exists at least one direction to express the most variance among data points, which turns out to guarantee the best classifying or clustering performance. In the fully connected RatioCut spectral clustering, the second lowest to the 11th lowest eigenvalues in its Laplacian are all zeros for a RBF kernel matrix with $\gamma \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. The fully connected spectral clustering reveals two facts in dealing with the kernel matrix from the clustering aspect. First, the kernel matrix with $\gamma \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ is impossible to have a stable clustering results under the fully connected RatioCut on MNIST-23. Secondly, since we have from Fig. 4,

$$\delta_{s,10^{-7},2} > \delta_{s,10^{-8},2} \gg \delta_{s,10^{-9},2} \gg \delta_{s,10^{-6},3} > \delta_{s,10^{-6},2},$$

the priority from high to low in selecting γ for spectral clustering is

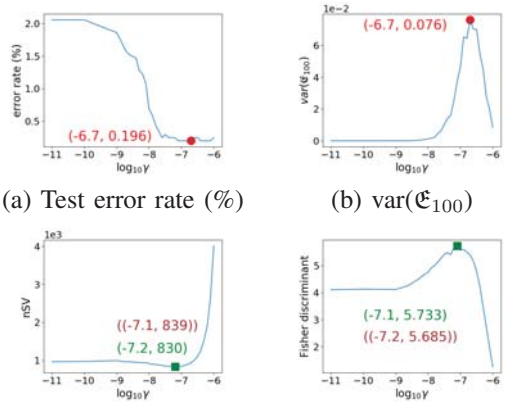
$$10^{-7}, 10^{-8}, 10^{-9}, 10^{-6}.$$

We locate $\gamma = 10^{-6}$ to the lowest priority because the eigengaps are small and the inequality $\delta_{s,10^{-6},3} > \delta_{s,10^{-6},2}$ implies that 3 is the better choice in deciding the number of clusters than 2 in spectral clustering. From the spectral analysis, we conclude that the precision $\gamma = 10^{-7}$ is the best choice in the set $\{10^{-2}, 10^{-3}, \ldots, 10^{-9}\}$ for the RBF kernel SVM on MNIST-23.

At the end, recall that the intermediate variable var$(\mathfrak{E}_n)$ is introduced in Subsection 3.4. Set $n = 100$. Fig. 2 reveals that the higher var$(\mathfrak{E}_{100})$ in Fig. 2-(b), the better test performance in Fig. 2-(a). The highest var$(\mathfrak{E}_{100})$ and the lowest error rate (%) are marked with red dots in Fig. 2-(b) and 2-(a) respectively. We will discuss more in Subsection 4.4.

4.3 Tolerance τ in the RBF Kernel SVM

Fig. 5 shows that when τ is small enough, the test performance becomes almost converged. Furthermore, if we choose $\gamma = 10^{-7}$ for the RBF kernel SVM, the test error rate (%) is even stabilized with larger tolerance τ. The sweeping results are illustrated in Fig. 5-(b).



(a) Test error rate (%) (b) var($\mathfrak{E}_{100}$)

(c) nSV (d) Fisher discriminant

Fig. 2

THE INFLUENCE OF PRECISION γ IN THE RBF KERNEL SVM ON MNIST-23, WHERE $\tau = 0.2$.

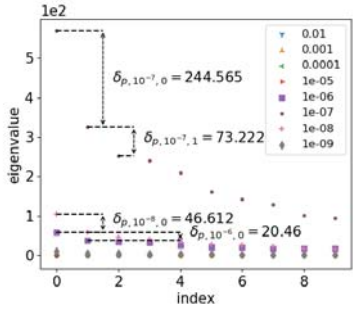
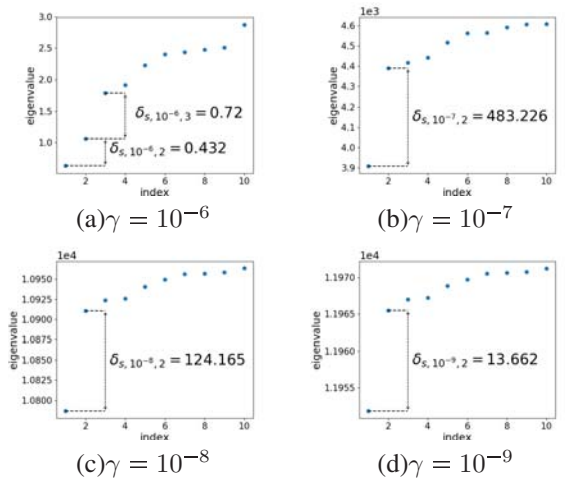


Fig. 3

THE 10 HIGHEST PCA EIGENVALUES OF THE RBF KERNEL MATRIX WITH DIFFERENT γ.

4.4 Role of var$(\mathfrak{E}_n)$ in the RBF Kernel SVM

Fig. 2 provides a connection between test error rate and var$(\mathfrak{E}_{100})$ in the RBF kernel SVM on MNIST-23. Their relation is formulated in Fig. 6 via curve fitting. Because η_{ij} in the RBF kernel SVM is equal to $2(1 - \exp^{-\gamma\|x_i - x_j\|^2})$, we apply $y = a + b\ln(x)$ as the curve fitting model. Note that every dot in Fig. 6 represents a testing result under the RBF kernel SVM corresponding to a distinct precision $\gamma \in \{6.0, 6.1, \ldots, 10.9, 11.0\}$. The results in Fig. 6 confirms that var$(\mathfrak{E}_n)$ reflects the performance of the kernel matrix in the RBF kernel SVM. Therefore, the higher var$(\mathfrak{E}_{100})$ is, the more suitable γ value of the RBF kernel is. Furthermore, since n is quite small, it leads us to devise a mini core validation to screen out the best parameter for the RBF kernel SVM. From the results in Subsection 4.1, the regulator C can also be determined in such a mini core validation before the fully run RBF kernel SVM is applied.

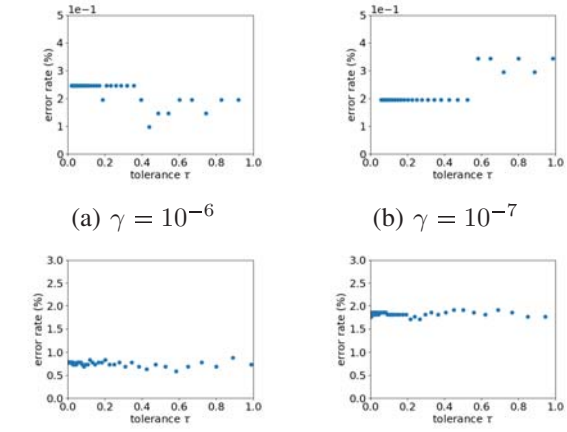


(a)$\gamma = 10^{-6}$ (b)$\gamma = 10^{-7}$

(c)$\gamma = 10^{-8}$ (d)$\gamma = 10^{-9}$

Fig. 4

THE EIGENVALUES $\mu^{(i)}_{s,\gamma}, i = [1, 10]$ OF THE FULLY CONNECTED RATIOCUT SPECTRAL CLUSTERING WITH DIFFERENT γ.



(a) $\gamma = 10^{-6}$ (b) $\gamma = 10^{-7}$

(c) $\gamma = 10^{-8}$ (d) $\gamma = 10^{-9}$

Fig. 5

TEST ERROR RATE (%) V.S. τ W.R.T. γ. THE UPPER LIMIT OF y-AXIS IS 0.5 FOR (A)(B); 3 FOR (C)(D).

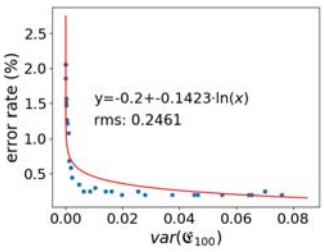


Fig. 6

TEST ERROR RATE (%) V.S. VAR($\mathfrak{E}_{100}$). ($\tau = 0.2$.)

A devised SVM algorithm will be proposed in Subsection 5.1, which includes the mini core validation and an SMO-SVM with adjustable tolerance τ.

5. Devised Algorithm and Its Performance

Based on the findings in Section 4, a devised SMO-SVM will be proposed in Subsection 5.1. The purpose of the devised SMO-SVM is to determine the best regulator C, tolerance τ and kernel parameters from the dataset itself, instead of the preset values. The performance of the proposed algorithm is demonstrated in Subsection 5.2.

5.1 Devised Algorithm

SMO-SVM partitions all the data points into two groups, I_1 and I_2, initially and then classifies the data points into $I_{0-}, I_{0+}, I_1, I_2, I_3, I_4$ iteration by iteration.

Instead of running cross validation for full SMO-SVM every time, we propose a mini core validation along with a devised SMO-SVM. This mini core validation is designed to find a proper regulator C for linear SMO-SVM or the better regulator C and kernel parameters for kernel-based SVM. We name this mini core validation as *miniCV*. This miniCV is an SMO-SVM except the $\tau-$violating pair is selected from $I_2 \times I_1$ and classified only into $I_{0,-} \times I_{0,+}$ in the first n iterations right after any initialization. In addition, the precision γ is determined via var($\mathfrak{E}_n$).

We divide miniCV into two parts. Determine kernel parameters first and, in the second part, evaluate the the regulator C and a tolerance under these kernel parameters. For RBF kernel SVM, miniCV has inputs L, H for the searching range $[L, H]$ of $-\log_{10}\gamma$. The miniCV keeps scanning for the highest var($\mathfrak{E}_n$) among different γ. In the present research, we apply greedy algorithm. Let z be an even integer in $[4, 10]$. With a preset $\epsilon_\gamma \in (0, 1)$, the stopping criterion for our greedy algorithm is $(H-L) < \epsilon_\gamma$. Denote $\rho_l = L + l \cdot \frac{H-L}{z}$, $l = [0, z]$. In a greedy process, each SMO-SVM with $\gamma_l = 10^{-\rho_l}$ iterates only n times and evaluate var($\mathfrak{E}_{n,\gamma_l}$). Let $p = \arg\max_l \text{var}(\mathfrak{E}_{n,\gamma_l})$. Then $[L^{new}, H^{new}]$ for next greedy process is updated as below, with $H^{new} > L^{new} > 10^{-10}$ for all H^{new}, L^{new}.

$$\begin{cases} [\rho_p - (H-L),\ \rho_p + (H-L)], & \text{if } p \in \{0, z\}, \\ [\rho_p - \frac{H-L}{z},\ \rho_p + \frac{H-L}{z}], & \text{otherwise .} \end{cases}$$

When the greedy algorithm stops, a precision γ_T with highest var($\mathfrak{E}_n$) is located. Since var($\mathfrak{E}_n$) is independent of C, this greedy algorithm in miniCV can be executed with any preset C. In the second part of miniCV, because the $\tau-$violating pair in miniCV is only classified into $I_{0,-} \times I_{0,+}$, the value of C must be well-defined to ensure all updating steps t in the second stage of miniCV is not clipped. Therefore, we run SMO-SVM under the precision γ_T for n effective iterations. An effective iteration is counted when

there is no more updated regulator C by Algorithm 1. At the end of the miniCV, a loosened tolerance τ_{pre} is estimated via Algorithm 2. The preset tolerance τ_{org} in miniCV is pretty small. If the dataset is small and separable, the miniCV ends at the n'th iteration with $F_{i^{(n')}}^{(n'-1)} - F_{j^{(n')}}^{(n'-1)} \leqslant \tau_{org}$ where $n' \leqslant n$. Let α be the time complexity of one SMO-SVM iteration. Then the time complexity in our miniCV is $O(n \cdot \alpha \cdot z \cdot \log_z \frac{H}{\epsilon_\gamma}) = O(n \cdot \alpha)$. When the training set is big, the complexity of miniCV becomes $O(\alpha)$.

Applying C, τ_{pre} and γ_T estimated in miniCV, a devised SMO-SVM is executed to train a model. This devised SMO-SVM is divided into two stages, pre-classification and normal stage. The elements in $\tau-$violating pair is classified from $I_2 \times I_1$ to $I_{0,-} \times I_{0,+}$ with the loosened tolerance τ_{pre} in pre-classification stage. At the end of the pre-classification stage, a tighten tolerance τ_{norm} is determined based on the distribution of $\{F_r, r \in I_{0,+} \cup I_{0,-}\}$ such as *mean* and *quantile*. Algorithm 3 details how the tightened τ_{norm} is assigned. Both miniCV and the preclassification stage are initialized as the initialization of the SMO-SVM algorithm in Section 2. When the following two conditions is satisfied in the pre-classification stage, this stage will be terminated and then τ_{norm} is estimated to start the normal stage.

- $F_i^{old} - F_j^{old} \leqslant \tau_{pre}$.
- $|t| \geqslant C$.

The first condition is the termination of pre-classification stage. The second condition is set because all $\tau-$violating pairs in pre-classification stage are selected from $I_2 \times I_1$ and updated into $I_{0,-} \times I_{0,+}$.

The normal stage inherits all the training results from the pre-classification stage, except it is an SMO-SVM in Section 2 under the tolerance τ_{norm}. Note the $\tau-$violating pair in normal stage is from $(I_{0,-} \cup I_{0,+} \cup I_2 \cup I_3) \times (I_{0,-} \cup I_{0,+} \cup I_1 \cup I_4)$.

Unlike ν-fold cross validation, we do not partition training set. The whole training set is utilized to run over miniCV in case some important points for hyperplane is missing. Since the characteristics of a dataset in $\mathbb{H}_S$ is determined by its kernel matrix, the number n in miniCV is small comparing to large dataset. This means miniCV is designed for locating the proper kernel parameter from few kernel related variables from the first n iterations. However, because the size of $\{F_r, r \in [1, m] | i = r, i \in I_{0,+} \cup I_{0,-}\}$ is only $2n$, it is not enough to determine a tolerance to complete the SMO-SVM. Therefore, we split SMO-SVM process into two stages,pre-classification and normal stages. Algorithm 3 in Pre-classification stage determines a proper tolerance τ_{norm} as to complete the SMO-SVM in the normal stage.

Summarize the target of each stage in the devised SMO-SVM as below.

- mini-CV: Determine the regulator C for linear SVM or the regulator C and kernel parameter for kernel-based SVM. Evaluate a loosened tolerance τ_{pre} for pre-classification stage.
- pre-classification stage: Determine a tightened tolerance τ_{norm} for normal stage.
- normal stage: Complete the training of SMO-SVM.

Note we denote the qth quantile of a series $\mathfrak{A}$ as Q($\mathfrak{A}$,q), for all $q \in [0, 1]$.

Algorithm 1 MiniCV Stage

```
function FINDC(|t|, C,maxStep) *
    maxStep← max{|t|, maxStep}
    if maxStep> C or C > 12∗maxStep then
        C ←maxStep∗6; Reinitailize SVM with new C
    else
        return maxStep
```

Algorithm 2 MiniCV Stage

```
function SETTAUPRE(F_{I_{0+}}, F_{I_{0−}}, τ_org) *
    Q_{2,+}=median(F_{I_{0+}}); Q_{2,−}=median(F_{I_{0−}});
    Q_{3,+} = Q(F_{I_{0+}}, 0.75); Q_{1,+} = Q(F_{I_{0+}}, 0.25);
    Q_{3,−} = Q(F_{I_{0−}}, 0.75); Q_{1,−} = Q(F_{I_{0−}}, 0.25)
    if Q_{2,+} < Q_{2,−} < Q_{2,+} + τ_org/2 then:
        Max = max(F_{I_{0−}}); Min = min(F_{I_{0+}});
        τ_pre = max{τ_org, (Max − Min) ∗ 2}
    else if Q_{2,−} ⩾ Q_{2,+} + τ_org/2 then:
        τ_pre = (Q_{2,−} − Q_{2,+})
    else
        τ_pre = τ_org
    return τ_pre
```

5.2 Performance

The performance of the proposed algorithm is illustrated by the testing error rate of all 10 digits in MNIST [11]. Each instance is a 28×28 pixels with one handwritten digit picture in gray scale. The size of the training and test dataset is 60000 and 10000 respectively.

We apply the one-versus-all method to do multi-classification on MNIST. If a target class in a one-versus-all task is a, only the instances labeled with a is tagged with $y = 1$, the others are assigned with $y = -1$. The miniCV for multi-classification is separated into two parts. The first half only decides best γ_a for each task. $\boldsymbol{\gamma} \triangleq \{\gamma_0, \ldots, \gamma_9\}$ is a set of recommended γ_a once all these 10 one-versus-all tasks complete its first half miniCV. n is set to 50 due to the large dataset. Then, determine a common precision $\hat{\gamma}$ for multi-classification by the mean estimated from Q($\boldsymbol{\gamma}$,25%) to Q($\boldsymbol{\gamma}$,75%). Then run the second half of miniCV for these 10 one-versus-all tasks to estimate $\hat{C}_a$ and $\hat{\tau}_{pre,a}$, if the target digit is a. Tolerance $\hat{\tau}_{norm,a}$ is estimated at the end of pre-classification. MNIST without deskewing is tested. Table 2 lists the binary classification test result for eaach one-versus-all task. The second and third columns are the

Algorithm 3 Pre-Classification Stage

function SETTAU($F_{I_{0+}}$,$F_{I_{0-}}$, τ_{pre}) *
  $Q_{2,+}$=median($F_{I_{0+}}$); $Q_{2,-}$=median($F_{I_{0-}}$);
  $Q_{3,+} = \mathrm{Q}(F_{I_{0+}}, 0.75)$; $Q_{1,+} = \mathrm{Q}(F_{I_{0+}}, 0.25)$;
  $Q_{3,-} = \mathrm{Q}(F_{I_{0-}}, 0.75)$; $Q_{1,-} = \mathrm{Q}(F_{I_{0-}}, 0.25)$;
  if $Q_{2,+} < Q_{2,-} < Q_{2,+} + \tau_{pre}/2$ **then**:
    $\tau_{norm} = (Q_{2,-} - Q_{2,+})$
  else if $Q_{2,-} \geqslant Q_{2,+} + \tau_{pre}/2$ **then**:
    if $Q_{1,-} > Q_{3,+}$ **then**
      $\tau_{norm} = (Q_{1,-} - Q_{3,+})$
    else
      $\tau_{norm} = (Q_{2,-} - Q_{2,+})/4$
  else
    Mean_+ =mean($F_{I_{0+}}$); Mean_- =mean($F_{I_{0-}}$);
    if $\mathrm{Mean}_- \geqslant \mathrm{Mean}_+$ **then**:
      τ_{norm} =Mean_- - Mean_+
    else
      $\tau_{norm} = \min\{Q_{3,-} - Q_{2,-}, Q_{2,-} - Q_{1,-},$
        $Q_{3,+} - Q_{2,+}, Q_{2,+} - Q_{1,+}\}$
  return τ_{norm}

Table 2
TEST RESULT OF DEVISED RBF KERNEL SVM UNDER EACH ONE-VERSUS-ALL MNIST TEST. THE RIGHT PART OF "||" IS TESTED BY COMMON PRECISION $\hat{\gamma} = 10^{-6.4276}$

a	$\log_{10}\gamma_a$	var($\mathfrak{E}_{50,a}$)	$\hat{C}_a$	$\hat{\tau}_{pre,a}$	$\hat{\tau}_{norm,a}$	error(%)
0	-6.66	0.093	7.44	0.35	0.091	0.16
1	-6.25	0.221	8.03	0.5	0.134	0.13
2	-6.54	0.109	8.64	0.43	0.109	0.36
3	-6.61	0.101	9.77	0.48	0.041	0.27
4	-6.32	0.134	7.11	0.76	0.112	0.27
5	-6.47	0.129	8.11	0.59	0.005	0.31
6	-6.34	0.166	7.56	0.43	0.112	0.25
7	-6.24	0.170	8.07	0.66	0.097	0.45
8	-6.50	0.091	8.23	0.47	0.007	0.45
9	-6.36	0.131	8.39	0.57	0.126	0.57

γ_a selected in first half of miniCV and the corresponding var($\mathfrak{E}_{50,a}$). The fourth column and fifth column represent the regulator $\hat{C}_a$ and loosened tolerance $\tau_{pre,a}$ estimated in the second half of miniCV under $\hat{\gamma} = 10^{-6.4276}$. Note the $\hat{\tau}_{norm,a}$ in the sixth column of Table 2 is small enough, since the 10 curves of dual problem value are almost converged shown in Fig. 7. The overall test error rate of RBF kernel SVM is 1.27% for non-deskwing MNIST. This result is comparable to the test record (1.4%) via RBF kernel SVM on non-deskewing MNIST in [11]. The test result via our parameter searching in miniCV beats the one (26.9%) via PSO parameter optimization in [2] and (4.4%) via SVM-BAT in [6].

6. Conclusion and Future Work

From the test results, the intermediate variable var($\mathfrak{E}_n$) indeed points to the best precision γ in the RBF kernel SVM.

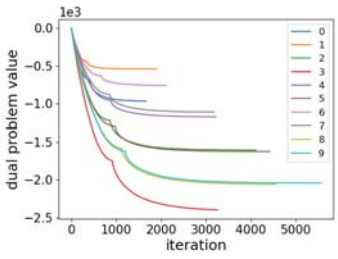


Fig. 7
DUAL PROBLEM VALUES OF THE DEVISED RBF KERNEL SMO-SVM ON NON-DESKEWING MNIST.

In additions, combining the miniCV, our devised SMO-SVM conducts a very competitive test result in classifying MNIST, compared with other implementations of the RBF kernel SVM. Also, the alterable tolerance adds the flexibility in terminating SMO-SVM earlier. In the future, we will do more research in the theory behind the relation in var($\mathfrak{E}_n$) and the RBF kernel SVM model selection. In the meantime, we would like to explore the training related variables for other kernel-based SVMs.

References

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Verlin, Heidelberg: Springer-Verlang, 1995.
[2] N. Omer Sahel Ba-Karait and S. M. Shamsuddin, "Handwritten digits recognition using particle swarm optimization," 05 2008, pp. 615–619.
[3] Y. Bao, Z. Hu, and T. Xiong, "A pso and pattern search based memetic algorithm for svms parameters optimization," *Neurocomputing*, vol. 117, pp. 98–106, 10 2013.
[4] H. Chen, B. Yen, and jing Wang, "Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy," *Applied Mathematics and Computation*, vol. 239, p. 180–197, 07 2014.
[5] L. Demidova, E. Nikulchev, and Y. Sokolova, "The svm classifier based on the modified particle swarm optimization," *International Journal of Advanced Computer Science and Applications*, vol. 7, pp. 16–24, 03 2016.
[6] M. T. Eva Tuba and D. Simian, "Handwritten digit recognition by support vector machine optimized by bat algorithm," in *WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision*, May 2016, pp. 369–376.
[7] J. C. Platt, "A fast algorithm for training support vector machines," *Advances in Kernel Methods-Support Vector Learning*, vol. 208, 07 1998.
[8] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to platt's smo algorithm for svm classifier design," *Neural Comput.*, vol. 13, no. 3, pp. 637–649, Mar. 2001. [Online]. Available: http://dx.doi.org/10.1162/089976601300014493
[9] S. Keerthi and E. Gilbert, "Convergence of a generalized smo algorithm for svm classifier design," *Machine Learning*, vol. 46, no. 1, pp. 351–360, Jan 2002. [Online]. Available: https://doi.org/10.1023/A:1012431217818
[10] U. von Luxburg, "A tutorial on spectral clustering," *CoRR*, vol. abs/0711.0189, 2007. [Online]. Available: http://arxiv.org/abs/0711.0189
[11] "The mnist database of handwritten digits," http://yann.lecun.com/exdb/mnist/, accessed: 2019-05-29.