Do Minimal Complexity Least Squares Support Vector Machines Work?

Shigeo Abe(✉)

Kobe University, Rokkodai, Nada, Kobe, Japan
abe@kobe-u.ac.jp
http://www2.kobe-u.ac.jp/ abe

Abstract. The minimal complexity support vector machine is a fusion of the support vector machine (SVM) and the minimal complexity machine (MCM), and results in maximizing the minimum margin and minimizing the maximum margin. It works to improve the generalization ability of the L1 SVM (standard SVM) and LP (Linear Programming) SVM. In this paper, we discuss whether it also works for the LS (Least Squares) SVM. The minimal complexity LS SVM (MLS SVM) is trained by minimizing the sum of squared margin errors and minimizing the maximum margin. This results in solving a set of linear equations and a quadratic program, alternatingly. According to the computer experiments for two-class and multiclass problems, the MLS SVM does not outperform the LS SVM for the test data although it does for the cross-validation data.

1 Introduction

A classifier is designed to achieve high generalization ability for unknown data by maximizing class separability. The support vector machine (SVM) [1,2] realizes this by maximizing the minimum margin, where a margin of a data sample is defined as its distance from the separating hyperplane. Although the SVM works relatively well for a wide range of applications, there is still a room for improvement. Therefore, in addition to maximizing the minimum margin, controlling the margin distribution is considered. One approach controls the low order statistics [3–8]. In [5], a large margin distribution machine (LDM) was proposed, in which the average margin is maximized and the margin variance is minimized. Because the LDM includes an additional hyperparameter compared to the SVM, in [6,7], the unconstrained LDM (ULDM) was proposed, which has the same number of hyperparameters as the SVM. The least squares SVM (LS SVM) [3,4] is consider to be based on low order statistics because it minimizes the sum of squared margin errors.

Another approach [9–15] minimizes the VC (Vapnik-Chervonenkis) dimension [1]. In [9], the minimal complexity machine (MCM) that minimizes the VC dimension was proposed, which is reduced to minimizing the sum of margin errors and minimizing the maximum margin. According to the analysis in [10],

N. El Gayar et al. (Eds.): ANNPR 2022, LNAI 13739, pp. 53–64, 2023.
https://doi.org/10.1007/978-3-031-20650-4_5

however, the solution of the MCM was shown to be non-unique and unbounded. These disadvantages can be solved by introducing the regularization term into the MCM, which is a fusion of the LP (Linear Programming) SVM and the MCM called MLP SVM. The soft upper-bound minimal complexity LP SVM (SLP SVM) [14] is a soft upper-bound version of the MLP SVM. The ML1 SVM [11,12] is the fusion of the MCM and the standard SVM (L1 SVM) and the SL1 SVM [15] is a soft upper-bound version of the ML1 SVM. According to the computer experiments, in general, the fusions, i.e., minimization of the maximum margin in the SVMs, improved the generalization ability of the base classifiers, and the $\mathrm{ML1_v}$ SVM, which is a variant of the ML1 SVM performed best.

In this paper, we discuss whether the idea of minimizing the VC dimension, i.e., minimizing the maximum margin, also works for the LS SVM, which controls the margin distribution by the second order statistics. We formulate the minimal complexity LS SVM (MLS SVM) by minimizing the maximum margin as well as maximizing the minimum margin in the LS SVM framework. We derive the dual form of the MLS SVM and the training method that trains the MLS SVM alternatingly by matrix inversion and by the SMO (Sequential Minimal Optimization) based Newton's method [16]. By computer experiments, we show whether the MLS SVM performs better than the LS SVM.

In Sect. 2, we discuss the architecture of the MLS SVM and derive its dual problem. Then we discuss the training method of the MLS SVM. And in Sect. 3, we compare generalization performance of the MLS SVM with the LS SVM and other SVM-based classifiers using two-class and multiclass problems.

2 Minimal Complexity Least Squares Support Vector Machines

In this section we discuss the architecture of the MLS SVM, the KKT conditions, and a training method.

2.1 Architecture

For a two-class problem, we consider the following decision function:

$$D(\mathbf{x}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) + b, \tag{1}$$

where $\mathbf{w}$ is the l-dimensional vector, b is the bias term, and $\boldsymbol{\phi}(\mathbf{x})$ is the l-dimensional vector that maps m-dimensional vector $\mathbf{x}$ into the feature space. If $D(\mathbf{x}) > 0$, $\mathbf{x}$ is classified into Class 1 and if $D(\mathbf{x}) < 0$, Class 2.

We introduce the idea of minimizing the VC dimension into the LS SVM: we minimize the maximum margin as well as maximizing the minimum margin.

The minimal complexity LS SVM (MLS SVM) is formulated as follows:

$$\min \quad \frac{1}{2}\,\mathbf{w}^\top\mathbf{w} + \frac{C}{2}\sum_{i=1}^{M}\xi_i^2 + C_h\,(h^+ + h^-) \tag{2}$$

$$\text{s.t.} \quad y_i\,(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x}_i) + b) = 1 - \xi_i \quad \text{for} \quad i = 1,\ldots,M, \tag{3}$$

$$h_i \geq y_i\,(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x}_i) + b) \quad \text{for} \quad i = 1,\ldots,M, \tag{4}$$

$$h^+ \geq 1, \quad h^- \geq 1, \tag{5}$$

where $(\mathbf{x}_i, y_i)$ $(i = 1,\ldots,M)$ are M training input-output pairs, $y_i = 1$ if $\mathbf{x}_i$ belong to Class 1, and $y_i = -1$ if Class 2, ξ_i are the slack variables for $\mathbf{x}_i$, C is the margin parameter, h^+ and h^- are the upper bounds for the Classes 1 and 2, respectively, and $h_i = h^+$ for $y_i = 1$ and $h_i = h^-$ for $y_i = -1$. Here, if $\xi_i \geq 1$, $\mathbf{x}_i$ is misclassified and otherwise, $\mathbf{x}_i$ is correctly classified. Unlike L1 or L2 SVMs, ξ_i can be negative. The first term in the objective function is the reciprocal of the squared margin divided by 2, the second term is to control the number of misclassifications, and C controls the tradeoff between the first and second terms. The third term works to minimize the maximum margin. Parameter C_h controls the upper bounds h^+ and h^-.

If we delete (4), (5), and the third term in (2), we obtain the LS SVM. And if in the LS SVM we replace the equality constraints in (3) into the inequality constraints ($\geq$) and the square sum of slack variables in (2) into the linear sum multiplied by 2, we obtain the L1 SVM, which is a standard SVM.

In the following, we derive the dual problem of the above optimization problem.

Introducing the Lagrange multipliers α_i, α_{M+i} (≥ 0), η^+ (≥ 0), and η^- (≥ 0) into (2) to (5), we obtain the unconstrained objective function:

$$\begin{aligned}
&Q(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, h^+, h^-, \eta^+, \eta^-)\\
&= \frac{1}{2}\,\mathbf{w}^\top\mathbf{w} + \frac{C}{2}\sum_{i=1}^{M}\xi_i^2 + C_h\,(h^+ + h^-) - \sum_{i=1}^{M}\alpha_i\,(y_i(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x}_i) + b) - 1 + \xi_i),\\
&-\sum_{i=1}^{M}\alpha_{M+i}\,(h_i - y_i\,(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x}_i) + b)) - \eta^+\,(h^+ - 1) - \eta^-\,(h^- - 1)
\end{aligned} \tag{6}$$

where $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_M,\alpha_{M+1},\ldots,\alpha_{2M})^\top$, and $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_M)^\top$.

Taking the partial derivatives of (6) with respect to $\mathbf{w}$, b, $\boldsymbol{\xi}$, h^+, and h^- and equating them to zero, together with the equality constraints (3), we obtain the optimal conditions as follows:

$$\mathbf{w} = \sum_{i=1}^{M} y_i \, (\alpha_i - \alpha_{M+i}) \, \boldsymbol{\phi}(\mathbf{x}_i), \tag{7}$$

$$\sum_{i=1}^{M} y_i \, (\alpha_i - \alpha_{M+i}) = 0, \tag{8}$$

$$\alpha_i = C \, \xi_i \quad \text{for} \quad i = 1, \ldots, M, \tag{9}$$

$$C_h = \sum_{i=1, y_i=1}^{M} \alpha_{M+i} + \eta^+, \quad C_h = \sum_{i=1, y_i=-1}^{M} \alpha_{M+i} + \eta^-, \tag{10}$$

$$y_i \, (\mathbf{w}^\top \, \boldsymbol{\phi}(\mathbf{x}_i) + b) - 1 + \xi_i = 0 \quad \text{for} \quad i = 1, \ldots, M, \tag{11}$$

$$\alpha_{M+i} \, (h_i - y_i \, (\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b)) = 0, \quad \alpha_{M+i} \geq 0 \quad \text{for} \quad i = 1, \ldots, M, \tag{12}$$

$$\eta^+ \, (h^+ - 1) = 0, \quad \eta^+ \geq 0, \quad \eta^- \, (h^- - 1) = 0, \quad \eta^- \geq 0. \tag{13}$$

From (9), unlike L1 or L2 SVMs, α_i can be negative.

Now, we derive the dual problem. Substituting (7) and (8) into (6), we obtain the objective function with respect to $\boldsymbol{\alpha}$, η^+, and η^-. Thus, we obtain the following dual problem:

$$\begin{aligned} \max \quad & Q(\boldsymbol{\alpha}, \eta^+, \eta^-) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} y_i \, (\alpha_i - \alpha_{M+i}) \\ & \quad \times y_j \, (\alpha_j - \alpha_{M+j}) K(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{2C} \sum_{i}^{M} \alpha_i^2 + \eta^+ + \eta^-, \end{aligned} \tag{14}$$

$$\text{s.t.} \quad \sum_{i=1}^{M} y_i \, (\alpha_i - \alpha_{M+i}) = 0, \tag{15}$$

$$C_h \geq \sum_{i=1, y_i=1}^{M} \alpha_{M+i}, \quad C_h \geq \sum_{i=1, y_i=-1}^{M} \alpha_{M+i}, \tag{16}$$

$$\alpha_{M+i} \geq 0 \quad \text{for} \quad i = 1, \ldots, M, \tag{17}$$

where $K(\mathbf{x}, \mathbf{x}')$ is the kernel and $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}^\top(\mathbf{x}) \, \boldsymbol{\phi}(\mathbf{x}')$. Similar to the SVM, defining $K(\mathbf{x}, \mathbf{x}')$, we can avoid the explicit treatment of variables in the feature space.

In the above optimization problem, if we delete $(\alpha_{M+1}, \ldots, \alpha_{2M})$, η^+, η^-, and their related terms, we obtain the LS SVM.

Similar to the ML1$_\mathrm{v}$ SVM [12], we assume that $\eta^+ = \eta^- = 0$. This means that $h^+ \geq 1$ and $h^- \geq 1$. Then the optimization problem reduces to

$$\max \quad Q(\boldsymbol{\alpha}) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} (\alpha_i - \alpha_{M+i})\,(\alpha_j - \alpha_{M+j})\, y_i\, y_j\, K(\mathbf{x}_i, \mathbf{x}_j)$$

$$- \frac{1}{2C} \sum_{i=1}^{M} \alpha_i^2, \tag{18}$$

$$\text{s.t.} \quad \sum_{i=1}^{M} y_i\, \alpha_i = 0, \tag{19}$$

$$C_h = \sum_{i=1, y_i=1}^{M} \alpha_{M+i} = \sum_{i=1, y_i=-1}^{M} \alpha_{M+i}, \tag{20}$$

$$\alpha_{M+i} \geq 0 \quad \text{for} \quad i = 1, \ldots, M. \tag{21}$$

Notice that because of (20), (15) reduces to (19).

We decompose the above optimization problem into two subprograms:

1. Subproblem 1 Solving the problem for $\alpha_1, \ldots, \alpha_M$ and b fixing $\alpha_{M+1}, \ldots, \alpha_{2M}$:

$$\max \quad Q(\boldsymbol{\alpha}^0) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} (\alpha_i - \alpha_{M+i})\,(\alpha_j - \alpha_{M+j})\, y_i\, y_j\, K(\mathbf{x}_i, \mathbf{x}_j)$$

$$- \frac{1}{2C} \sum_{i=1}^{M} \alpha_i^2, \tag{22}$$

$$\text{s.t.} \quad \sum_{i=1}^{M} y_i\, \alpha_i = 0, \tag{23}$$

where $\boldsymbol{\alpha}^0 = (\alpha_1, \ldots, \alpha_M)^\top$.

2. Subproblem 2 Solving the problem for $\alpha_{M+1}, \ldots, \alpha_{2M}$ fixing $\boldsymbol{\alpha}^0$ and b:

$$\max \quad Q(\boldsymbol{\alpha}^M) = -\frac{1}{2} \sum_{i,j=1}^{M} (\alpha_i - \alpha_{M+i})\,(\alpha_j - \alpha_{M+j})\, y_i\, y_j\, K(\mathbf{x}_i, \mathbf{x}_j) \tag{24}$$

$$\text{s.t.} \quad C_h = \sum_{i=1, y_i=1}^{M} \alpha_{M+i} = \sum_{i=1, y_i=-1}^{M} \alpha_{M+i}, \tag{25}$$

$$\alpha_{M+i} \geq 0 \quad \text{for} \quad i = 1, \ldots, M, \tag{26}$$

where $\boldsymbol{\alpha}^M = (\alpha_{M+1}, \ldots, \alpha_{2M})^\top$.

We must notice that as the value of C_h approaches zero, the MLS SVM reduces to the LS SVM. Therefore, for a sufficiently small value of C_h, the MLS SVM and LS SVM behave similarly.

We consider solving the above subproblems alternatingly.

Here, because of (25), if we modify α_{M+i}, another α_{M+j} belonging to the same class must be modified. Therefore, $\boldsymbol{\alpha}^M$ can be updated per class.

2.2 Solving Subproblem 1

Variables $(\alpha_1, \ldots, \alpha_M)$ and b can be solved for using (7), (9), (11), and (23) by matrix inversion. Substituting (7) and (9) into (11) and expressing it and (23) in matrix form, we obtain

$$\begin{pmatrix} \varOmega & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}' \\ b \end{pmatrix} = \begin{pmatrix} \mathbf{d}_1 \\ 0 \end{pmatrix}, \tag{27}$$

or

$$\varOmega \boldsymbol{\alpha}' + \mathbf{1} b = \mathbf{d}_1, \tag{28}$$
$$\mathbf{1}^\top \boldsymbol{\alpha}' = 0, \tag{29}$$

where $\mathbf{1}$ is the M-dimensional vector and

$$\boldsymbol{\alpha}' = (y_1\, \alpha_1, \ldots, y_M\, \alpha_M)^\top \tag{30}$$
$$\varOmega_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) + \frac{\delta_{ij}}{C}, \tag{31}$$
$$\mathbf{d}_1 = (d_{11}, \ldots, d_{1M})^\top, \tag{32}$$
$$d_{1i} = y_i + \sum_{j=1}^{M} y_j\, \alpha_{M+j}\, K(\mathbf{x}_i, \mathbf{x}_j), \tag{33}$$
$$\mathbf{1} = (1, \ldots, 1)^\top, \tag{34}$$

where $\delta_{ij} = 1$ for $i = j$, and $\delta_{ij} = 0$ for $i \neq j$.

If $\boldsymbol{\alpha}^M = \mathbf{0}$, (27) reduces to solving the LS SVM.

Subproblem 1 is solved by solving (27) for $\boldsymbol{\alpha}^0$ and b as follows. Because of $1/C\,(> 0)$ in the diagonal elements of $\varOmega$, $\varOmega$ is positive definite. Therefore,

$$\boldsymbol{\alpha}' = \varOmega^{-1}(\mathbf{d}_1 - \mathbf{1}\, b). \tag{35}$$

Substituting (35) into (29), we obtain

$$b = (\mathbf{1}^\top \varOmega^{-1} \mathbf{1})^{-1} \mathbf{1}^\top \varOmega^{-1} \mathbf{d}_1. \tag{36}$$

Thus, substituting (36) into (35), we obtain $\boldsymbol{\alpha}'$.

2.3 Solving Subproblem 2

Subproblem 2 needs to be solved iteratively. We derive the KKT (Karush-Kuhn-Tucker) conditions for Subproblem 2 for the convergence check. Because of the space limitation, we skip the detailed training method based on the SMO (Sequential Minimal Optimization) combined with Newton's method [16].

For Subprogram 2, training is converged if the KKT optimality condition (12) is satisfied. Substituting (7) and (9) into (12), we obtain the following KKT conditions:

$$\alpha_{M+i}\ (h_i + y_i\, F_i - y_i\, b) = 0 \quad \text{for} \quad i = 1, \ldots, M, \tag{37}$$

where

$$F_i = -\sum_{j=1}^{M} y_j (\alpha_j - \alpha_{M+j}) K(\mathbf{x}_i, \mathbf{x}_j). \tag{38}$$

Here the value of b is determined in Subprogram 1.

KKT Conditions. We can classify the conditions of (37) into the following two cases:

1. $\alpha_{M+i} = 0$. From $h_i \geq -y_i\, F_i + y_i\, b$,

$$F_i \geq b - h^+ \ \text{ for } \ y_i = 1, \ \ b + h^- \geq F_i \ \text{ for } \ y_i = -1. \tag{39}$$

2. $C_h \geq \alpha_{M+i} > 0$. From $h_i = -y_i\, F_i + y_i\, b$,

$$b - h^+ = F_i \ \text{ for } \ y_i = 1, \ \ b + h^- = F_i \ \text{ for } \ y_i = -1. \tag{40}$$

Then the KKT conditions for (37) are simplified as follows:

$$\begin{aligned} &\bar{F}_i^{\ +} \geq b - h^+ \geq \tilde{F}_i^{\ +} \ \text{ for } \ y_i = 1, \\ &\bar{F}_i^{\ -} \geq b + h^- \geq \tilde{F}_i^{\ -} \ \text{ for } \ y_i = -1, \quad \text{for} \quad i = 1, \ldots, M, \end{aligned} \tag{41}$$

where

$$\bar{F}_i^{\ +} = F_i \quad \text{if} \quad \alpha_{M+i} \geq 0, \quad \tilde{F}_i^{\ +} = F_i \quad \text{if} \quad \alpha_{M+i} > 0, \tag{42}$$

$$\bar{F}_i^{\ -} = F_i \quad \text{if} \quad \alpha_{M+i} > 0, \quad \tilde{F}_i^{\ -} = F_i \quad \text{if} \quad \alpha_{M+i} \geq 0. \tag{43}$$

To detect the violating variables, we define b^s_{up} and b^s_{low} as follows:

$$b^s_{\text{up}} = \min_i \bar{F}_i^{\ s}, \quad b^s_{\text{low}} = \max_i \tilde{F}_i^{\ s}, \tag{44}$$

where $s = +, -$, $b^+ = b - h^+$, and $b^- = b + h^-$.

If the KKT conditions are satisfied,

$$b^s_{\text{up}} \geq b^s_{\text{low}}. \tag{45}$$

To speed up training we consider that training is converged if

$$\max_{s=+,-} b^s_{\text{low}} - b^s_{\text{up}} \leq \tau, \tag{46}$$

where τ is a small positive parameter.

And the upper bounds are estimated to be

$$h^+_{\text{e}} = b - \frac{1}{2}(b^+_{\text{up}} + b^+_{\text{low}}), \ \ h^-_{\text{e}} = -b + \frac{1}{2}(b^-_{\text{up}} + b^-_{\text{low}}). \tag{47}$$

2.4 Training Procedure

In the following we show the training procedure of the MLS SVM.

1. (Solving Subprogram 1) Solve (27) for $\boldsymbol{\alpha}^0$ and b fixing $\boldsymbol{\alpha}^M$ with the solution obtained in Step 2. Initially we set $\boldsymbol{\alpha}^M = \mathbf{0}$.
2. (Solving Subprogram 2) Solve (24)–(26) for $\boldsymbol{\alpha}^M$ fixing $\boldsymbol{\alpha}^0$ and b with the solution obtained in Step 1. Initially, we set one α_{M+i} in each class to C_h.
3. (Convergence check) If (46) is satisfied, finish training. Otherwise go to Step 1.

The objective function $Q(\boldsymbol{\alpha})$ is monotonic during training: In Step 1, the objective function is maximized with the fixed $\boldsymbol{\alpha}^M$. Therefore the objective function is non-decreasing after $\boldsymbol{\alpha}^0$ and b are corrected. In Step 2, the objective function is maximized with the fixed $\boldsymbol{\alpha}^0$ and b. Therefore, the objective function is also non-decreasing after $\boldsymbol{\alpha}^M$ is corrected. In Step 2, so long as (45) is not satisfied, the objective function is increased by correcting $\boldsymbol{\alpha}^M$. Therefore, the training stops within finite steps.

The hyperparameter values of γ, C, and C_h are determined by cross-validation. To make the accuracy improvement over the LS SVM clear, in the following performance evaluation, we determined the values of γ and C, with $C_h = 0$, i.e., using the LS SVM. After they were determined, we determined the C_h value of MLS SVM. By this method, we can make the accuracy of the MLS SVM at least by cross-validation not lower than that of the LS SVM, if the smallest value of C_h in cross-validation is sufficiently small.

3 Performance Evaluation

We evaluated whether the idea of minimizing the VC-dimension, i.e., minimizing the maximum margin works to improve the generalization ability of the LS SVM. As classifiers we used the MLS SVM, LS SVM, ML1$_\mathrm{v}$ SVM, which is a variant of ML1 SVM, L1 SVM, and ULDM. As a variant of the MLS SVM, we used an early stopping MLS SVM, MLS$_\mathrm{e}$ SVM, which terminates training when the Subprogram 2 converges after matrix inversion for Subprogram 1 is carried out. This was to check whether early stopping improves the generalization ability when overfitting occurs.

To make comparison fair we determined the values of the hyperparameters by fivefold cross-validation of the training data, trained the classifiers with the selected hyperparameter values, and tested the accuracies for the test data. (Because of the computational cost we did not use nested (double) cross-validation.) We used two-class and multiclass problems used in [15]. The two-class problems have 100 or 20 pairs of training and test data sets and the multiclass problems, one, each. In cross-validation, the candidate values for γ and C were the same as those discussed in [15]. Those for C_h in the MLS SVM and MLS$_\mathrm{e}$ SVM were $\{0.001, 0.01, 0.1, 1, 10, 50, 100, 500\}$ instead of $\{0.1, 1, 10, 50, 100, 500, 1000, 2000\}$ in the ML1$_\mathrm{v}$ SVM. This was to avoid deteriorating the cross-validation accuracy in determining the C_h value. In addition, a tie was broken by selecting a smallest value except for $\mathrm{MLS_e}$ SVM; for the $\mathrm{MLS_e}$ SVM, a largest value was selected so that minimizing the maximum margin worked.

Table 1 shows the average accuracies for the 13 two-class problems. In the first column, in I/Tr/Te, I shows the number of inputs, Tr, the number of training data, and Te, the number of test data. For each problem, the best average accuracy is in bold and the worst, underlined. The average accuracy is accompanied by the standard deviation. The plus sign attached to the accuracy shows that the MLS SVM is statistically better than the associated classifier. Likewise, the minus sign, worse than the associated classifier. The "Average" row shows the average accuracy of the associated classifier for the 13 problems and B/S/W denotes that the associated classifier shows the best accuracy B times, the second best, S times, and the worst accuracy, W times. The "W/T/L" denotes that the MLS SVM is statistically better than the associated classifier W times, comparable to, T times, and worse than, L times.

From the Average measure, the ULDM performed best and the $\mathrm{MLS_e}$ SVM, the worst. And both the MLS SVM and $\mathrm{MLS_e}$ SVM were inferior to the LS SVM. From the B/S/W measure, also the ULDM was the best and the LS SVM was the second best. From the W/T/L measure, the MLS SVM was better than the $\mathrm{MLS_e}$ SVM and comparable or almost comparable to the LS SVM, $\mathrm{ML1_v}$ SVM, and ULDM. Although the MLS SVM was statistically comparable to or better than other classifiers, from the Average measure, it was inferior to the LS SVM. To investigate, why this happened, we compared the average accuracy obtained by cross-validation, which is shown in Table 2. From the table, the MLS SVM showed the best average accuracies for all the problems. This shows that the idea of minimizing the maximum margin worked for the MLS SVM at least for the cross-validation accuracies. But from Table 1, the MLS SVM was better than or equal to the LS SVM for only three problems: the diabetes, flare-solar, and splice problems. This shows that in most cases overfitting occurred for the MLS SVM. On the other hand, the $\mathrm{MLS_e}$ SVM was inferior to the LS SVM except for the test data accuracy of the titanic problem. Thus, in most cases, inferior performance was caused by underfitting.

Table 4 shows the accuracies of the test data for the multiclass problems. The original MNIST data set has 6000 data points per class and it is difficult to train the low order statistic-based classifiers by matrix inversion. Therefore, to reduce the cross-validation time, we switched the roles of training and test data sets for the MNIST problem and denote it as MNIST (r). From the Average measure, the $\mathrm{ML1_v}$ SVM performed best, the ULDM the second best, and the MLS SVM and $\mathrm{MLS_e}$ SVM, worst. From the B/S/W measure, the $\mathrm{MLS_e}$ SVM was the best and the MLS SVM the worst. For the $\mathrm{MLS_e}$ SVM, the accuracy for the thyroid problem was the worst. Comparing the $\mathrm{MLS_e}$ SVM and LS SVM, the $\mathrm{MLS_e}$ SVM performed better than the LS SVM six times, but the MLS SVM, only once. Therefore, the $\mathrm{MLS_e}$ SVM performed better than the LS SVM but MLS SVM did not.

Table 1. Average accuracies of the test data for the two-class problems

Problem I/Tr/Te	MLS	MLS$_\mathrm{e}$	LS	ML1$_\mathrm{v}$	L1	ULDM
Banana 2/400/4900	89.16 ± 0.68	89.02 ± 0.79	**89.17** ± 0.66	89.13 ± 0.70	**89.17** ± 0.74	89.12 ± 0.69
Cancer 9/200/77	72.99 ± 4.66	71.01^{+} ± 4.38	73.13 ± 4.68	73.14 ± 4.38	72.99 ± 4.49	**73.70** ± 4.42
Diabetes 8/468/300	76.21 ± 2.01	74.76^{+} ± 2.77	76.19 ± 2.00	76.36 ± 1.84	76.23 ± 1.80	**76.51** ± 1.95
Flare-solar 9/666/400	66.25 ± 1.98	63.62^{+} ± 2.65	66.25 ± 1.98	**66.99**$^{-}$ ± 2.16	**66.99**$^{-}$ ± 2.12	66.28 ± 2.05
German 20/700/300	76.00 ± 2.28	74.72^{+} ± 3.31	76.10 ± 2.10	75.88 ± 2.18	76.01 ± 2.12	**76.12** ± 2.30
Heart 13/170/100	82.43 ± 3.53	82.35 ± 3.61	82.49 ± 3.60	**82.89** ± 3.33	82.72 ± 3.40	82.57 ± 3.64
Image 18/1300/1010	97.50 ± 0.57	97.14^{+} ± 0.52	**97.52** ± 0.54	97.28 ± 0.46	97.16^{+} ± 0.41	97.16 ± 0.68
Ringnorm 20/400/7000	98.18 ± 0.35	97.29^{+} ± 1.56	**98.19** ± 0.33	98.01 ± 1.11	98.14 ± 0.34	98.16 ± 0.35
Splice 60/1000/2175	89.00 ± 0.71	88.93 ± 0.82	88.98 ± 0.70	88.99 ± 0.83	88.89 ± 0.84	**89.16** ± 0.53
Thyroid 5/140/75	95.04 ± 2.56	94.84 ± 2.60	95.08 ± 2.55	95.35 ± 2.48	**95.39** ± 2.43	95.15 ± 2.27
Titanic 3/150/2051	77.30 ± 1.27	77.42 ± 0.78	77.39 ± 0.83	77.42 ± 0.74	77.35 ± 0.80	**77.46** ± 0.91
Twonorm 20/400/7000	97.40 ± 0.28	97.05^{+} ± 0.60	**97.43** ± 0.27	97.37 ± 0.28	97.38 ± 0.27	97.41 ± 0.26
Waveform 21/400/4600	90.01 ± 0.58	89.32^{+} ± 1.15	90.05 ± 0.59	89.66^{+} ± 0.76	89.72^{+} ± 0.70	**90.18**$^{-}$ ± 0.54
Average (B/S/W)	85.19 (0/3/1)	84.42 (0/1/12)	85.23 (4/2/0)	85.27 (2/4/0)	85.24 (3/1/0)	**85.31** (6/1/0)
W/T/L	—	8/5/0	0/13/0	1/11/1	2/10/1	0/12/1

Table 2. Average accuracies by cross-validation for the two-class problems

Problem	MLS	MLS$_\mathrm{e}$	LS
Banana	**90.60**	90.21	90.50
Cancer	**76.03**	72.77	75.99
Diabetes	**78.19**	76.07	78.15
Flare-solar	**67.38**	63.59	67.36
German	**76.59**	74.36	76.58
Heart	**84.70**	83.99	84.59
Image	**97.39**	97.24	97.38
Ringnorm	**98.65**	97.53	98.60
Splice	**89.02**	89.01	88.94
Thyroid	**97.44**	97.04	97.37
Titanic	**79.49**	78.81	79.45
Twonorm	**98.06**	97.55	97.98
Waveform	**91.06**	90.05	91.00
Average	**86.51**	85.25	86.45
B/W	13/0	0/12	0/1

Table 3. Average accuracies by cross-validation for the multiclass problems

Problem	MLS	MLS$_\mathrm{e}$	LS
Numeral	**99.63**	99.51	**99.63**
Thyroid	**95.97**	95.02	**95.97**
Blood cell	**94.83**	94.54	**94.83**
Hiragana-50	99.67	**99.70**	99.67
Hiragana-13	**99.86**	99.83	**99.86**
Hiragana-105	**99.98**	**99.98**	**99.98**
Satimage	92.72	**92.76**	92.72
USPS	**98.46**	98.44	98.44
MNIST(r)	**97.59**	**97.59**	**97.59**
Letter	**97.81**	97.74	**97.81**
Average	**97.65**	97.51	**97.65**
B/W	8/2	4/6	7/3

Now examine the result from the cross-validation accuracies shown in Table 3. The accuracies of the MLS SVM were the same as those of the LS SVM except for the USPS problem. Therefore, from Table 4, the idea of minimizing the maximum margin did not contribute in improving the accuracies of the test data except for the blood cell problem. For the MLS$_\mathrm{e}$ SVM, the adverse effect of early stopping occurred for the thyroid problem: the worst accuracy of the test data in Table 4 was caused by underfitting as seen from Table 3. For the remaining problems the adverse effect was small or none.

Table 4. Accuracies of the test data for the multiclass problems

Problem I/C/Tr/Te	MLS	MLS_e	LS	ML1_v	L1	ULDM
Numeral 12/10/810/820 [2]	99.15	99.39	99.15	**99.76**	**99.76**	99.39
Thyroid 21/3/3772/3428 [17]	95.39	94.57	95.39	97.23	**97.26**	95.57
Blood cell 13/12/3097/3100 [2]	94.29	94.29	94.23	93.65	93.19	**94.61**
Hiragana-50 50/39/4610/4610 [2]	99.20	99.28	**99.48**	99.11	98.98	98.92
Hiragana-13 13/38/8375/8356 [2]	99.87	99.88	99.87	**99.92**	99.76	99.90
Hiragana-105 105/38/8375/8356 [2]	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Satimage 36/6/4435/2000 [17]	91.95	**92.30**	91.95	91.85	91.90	92.25
USPS 256/10/7291/2007 [18]	95.42	**95.52**	95.47	95.37	95.27	95.42
MNIST(r) 784/10/10000/60000 [19]	96.98	**97.03**	96.99	96.95	96.55	**97.03**
Letter 16/26/16000/4000 [17]	97.87	97.85	97.88	**98.03**	97.70	97.75
Average	97.01	97.01	97.04	**97.18**	97.04	97.08
B/S/W	1/1/1	4/2/1	2/2/1	4/1/1	3/0/5	3/2/1

For the experiment of the multiclass problems, we compared the accuracies of the classifiers because we had only one training data set and one test data set. It was possible to generate multiple training and test data sets from the original data. However, we avoided this because of long cross-validation time. To strengthen performance comparison, in the future, we would like to compare classifiers statistically using multiple training and test data sets.

4 Conclusions

In this paper we proposed the minimal complexity least squares support vector machine (MLS SVM), which is a fusion of the LS SVM and the minimal complexity machine (MCM). Unlike the ML1_v SVM, which is a fusion of the L1 SVM and the MCM, the MLS SVM did not show an improvement in the accuracy for the test data over the LS SVM although the MLS SVM showed an improvement for the cross-validation accuracy. However, early stopping of the MLS SVM training sometimes showed improvement over the LS SVM.

In the future, we would like to compare performance of classifiers statistically using multiple training and test data sets.

Acknowledgment. This work was supported by JSPS KAKENHI Grant Numbers 19K04441 and 22K04154.

References

1. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
2. Abe, S.: Support Vector Machines for Pattern Classification, 2nd edn. Springer, London (2010)
3. Suykens, J.A.K.: Least squares support vector machines for classification and nonlinear modelling. Neural Network World **10**(1–2), 29–47 (2000)
4. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific Publishing, Singapore (2002)
5. Zhang, T., Zhou, Z.-H.: Large margin distribution machine. In: Twentieth ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 313–322 (2014)
6. Abe, S.: Unconstrained large margin distribution machines. Pattern Recogn. Lett. **98**, 96–102 (2017)
7. Abe, S.: Effect of equality constraints to unconstrained large margin distribution machines. In: Pancioni, L., Schwenker, F., Trentin, E. (eds.) ANNPR 2018. LNCS (LNAI), vol. 11081, pp. 41–53. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99978-4_3
8. Zhang, T., Zhou, Z.: Optimal margin distribution machine. IEEE Trans. Knowl. Data Eng. **32**(6), 1143–1156 (2020)
9. Jayadeva: Learning a hyperplane classifier by minimizing an exact bound on the VC dimension. Neurocomputing **149**, 683–689 (2015)
10. Abe, S.: Analyzing minimal complexity machines. In: Proceedings of International Joint Conference on Neural Networks, pp. 1–8. Budapest, Hungary (2019)
11. Abe, S.: Minimal complexity support vector machines. In: Schilling, F.-P., Stadelmann, T. (eds.) ANNPR 2020. LNCS (LNAI), vol. 12294, pp. 89–101. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58309-5_7
12. Abe, S.: Minimal complexity support vector machines for pattern classification. Computers **9**, 88 (2020)
13. Jayadeva, Soman, S., Pant, H., Sharma, M.: QMCM: Minimizing Vapnik's bound on the VC dimension. Neurocomputing **399**, 352–360 (2020)
14. Abe, S.: Soft upper-bound minimal complexity LP SVMs. In: Proceedings of International Joint Conference on Neural Networks, pp. 1–7 (2021)
15. Abe, S.: Soft upper-bound support vector machines. In: Proceedings of International Joint Conference on Neural Networks, pp. 1–8 (2022)
16. Abe, S.: Fusing sequential minimal optimization and Newton's method for support vector training. Int. J. Mach. Learn. Cybern. **7**(3), 345–364 (2016)
17. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007). http://www.ics.uci.edu/~mlearn/MLRepository.html
18. USPS Dataset. https://www.kaggle.com/bistaumanga/usps-dataset
19. LeCun, Y., Cortes, C.: The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/