Minimal Complexity Support Vector Machines

Shigeo Abe(✉)

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

Abstract. Minimal complexity machines (MCMs) minimize the VC (Vapnik-Chervonenkis) dimension to obtain high generalization abilities. However, because the regularization term is not included in the objective function, the solution is not unique. In this paper, to solve this problem, we propose fusing the MCM and the standard support vector machine (L1 SVM). This is realized by minimizing the upper bound on the decision function for the training data in the L1 SVM. We call the machine Minimum complexity L1 SVM (ML1 SVM). We compare the ML1 SVM with other types of SVMs including the L1 SVM using several benchmark data sets and show that the ML1 SVM performs comparable to or better than the L1 SVM.

1 Introduction

In the support vector machine (SVM) [1,2], training data are mapped into the high dimensional feature space, and in that space, the separating hyperplane is determined so that the nearest training data of both classes are maximally separated. Here, the distance between a data sample and the separating hyperplane is called margin.

Motivated by the success of SVMs in real world applications, many SVM-like classifiers have been developed to improve the generalization ability. The ideas of extensions lie in incorporating the data distribution (or margin distribution) to the classifiers.

To cope with this, one approach proposes kernels based on the Mahalanobis distance [3,4]. Another approach reformulates the SVM so that the margin is measured by the Mahalanobis distance [5,6].

Yet another approach controls the overall margins instead of the minimum margin. In [7], a large margin distribution machine (LDM) is proposed, in which the average margin is maximized and the margin variance is minimized. Although the generalization ability is better than that of the SVM, the number of hyperparameters is larger than that of the SVM. To cope with this problem, in [8], the unconstrained LDM (ULDM) is proposed, which has the equal number of hyperparameters and which has the generalization ability comparable to that of the LDM and the SVM.

F.-P. Schilling and T. Stadelmann (Eds.): ANNPR 2020, LNAI 12294, pp. 89–101, 2020.
https://doi.org/10.1007/978-3-030-58309-5_7

The generalization ability of the SVM can be analyzed by the VC (Vapnik-Chervonenkis) dimension [1] and the maximum generalization ability is achieved by minimizing the radius-margin ratio, where the radius is the minimum radius of the hypersphere that encloses all the training data in the feature space.

If the center of the hypersphere is assumed to be at the origin, the radius of the hypersphere can be minimized for a given feature space as discussed in [9]. The minimal complexity machine (MCM) is derived based on this assumption. In the MCM, the VC dimension is minimized by minimizing the upper bound of the soft-margin constraints for the decision function. Because the regularization term is not included, the MCM is trained by linear programming. The generalization performance of the MCM is shown to be superior to that of the SVM, but according to our analysis [10], the solution is non-unique and the generalization ability is not better than that of the SVM. The problem of non-uniqueness is shown to be solved by adding the regularization term in the objective function of the MCM, which is a fusion of the MCM and the linear programming SVM (LP SVM) called MLP SVM.

In this paper we propose fusing the MCM with the standard SVM, i.e., L1 SVM, to improve the generalization ability of the L1 SVM. We call the fused architecture minimal complexity L1 SVM (ML1 SVM). The ML1 SVM is obtained by adding the upper bound on the decision function and the upper bound minimization term in the objective function of the L1 SVM. We derive the dual form of the ML1 SVM with one set of variables associated with the soft-margin constraints and the other set, upper-bound constraints. We then decompose the dual ML1 SVM into two subproblems: one for the soft-margin constraints, which is similar to the dual L1 SVM, and the other for the upper-bound constraints. These subproblems neither include the bias term nor the upper bound. Thus, for a convergence check, we derive the exact KKT (Karush-Kuhn-Tucker) conditions that do not include the bias term and the upper bound. The second subproblem is different from the first subproblem in that it includes the inequality constraint on the sum of dual variables. To remove this, we change the inequality constraint into two equality constraints and called this architecture $\mathrm{ML1_v}$ SVM.

In Sect. 2, we summarize the architectures of L1 SVM and the MCM. In Sect. 3, we discuss the architectures of the ML1 SVM and $\mathrm{ML1_v}$ SVM. In Sect. 4, we compare the generalization ability of the proposed classifiers with other SVM-like classifiers using two-class and multiclass problems.

2 L1 Support Vector Machines and Minimal Complexity Machines

In this section, we briefly explain the architectures of the L1 SVM and the MCM [9]. Then we discuss the problem of non-unique solutions of the MCM and one approach to solving the problem [10].

2.1 L1 Support Vector Machines

Let the M training data and their labels be $\{\mathbf{x}_i, y_i\}(i = 1, \ldots, M)$, where $\mathbf{x}_i$ is an n-dimensional input vector and $y_i = 1$ for Class 1 and -1 for Class 2. The input space is mapped into the l-dimensional feature space by the mapping function $\boldsymbol{\phi}(\mathbf{x})$ and in the feature space the following separating hyperplane is constructed:

$$\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) + b = 0, \tag{1}$$

where $\mathbf{w}$ is the l-dimensional constant vector and b is the bias term.

The primal form of the L1 SVM is given by

$$\min Q(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{M} \xi_i \tag{2}$$

$$\text{s.t. } y_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b\right) + \xi_i \geq 1, \ \xi_i \geq 0, \ i = 1, \ldots, M, \tag{3}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)^\top$, ξ_i is the slack variable for $\mathbf{x}_i$, and $C\,(> 0)$ is the margin parameter that determines the trade-off between the maximization of the margin and minimization of the classification error. Inequalities (3) are called soft-margin constraints.

2.2 Minimal Complexity Machines

The VC dimension is a measure for estimating the generalization ability of a classifier and lowering the VC dimension leads to realizing a higher generalization ability. For an SVM-like classifier with the minimum margin $\delta_{\min}$, the VC dimension D is bounded by [1]

$$D \leq 1 + \min\left(R^2/\delta_{\min}^2, l\right), \tag{4}$$

where R is the radius of the smallest hypersphere that encloses all the training data.

In training the L1 SVM, both R and l are not changed. In the LS SVM, where ξ_i are replaced with ξ_i^2 in (2) and the inequality constraints, with equality constraints in (3), although both R and l are not changed by training, the second term in the objective function works to minimize the square sums of $y_i\, f(\mathbf{x}_i) - 1$. Therefore, like the LDM and ULDM, this term works to condense the margin distribution in the direction orthogonal to the separating hyperplane.

The MCM that minimizes the VC-dimension, i.e., $R/\delta_{\min}$ in (4) is

$$\min Q(\boldsymbol{\alpha}, h, \boldsymbol{\xi}, b) = h + C \sum_{i=1}^{M} \xi_i \tag{5}$$

$$\text{s.t. } h \geq y_i \left(\sum_{j=1}^{M} \alpha_j K_{ij} + b \right) + \xi_i \geq 1, \ i = 1, \ldots M, \tag{6}$$

where h is the upper bound of the soft-margin constraints and $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}^\top(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_j)$. Here, the mapping function $\boldsymbol{\phi}(\mathbf{x})$ in (1) is [11]

$$\boldsymbol{\phi}(\mathbf{x}) = (K_{11}, \ldots, K_{1M})^\top, \tag{7}$$

and $\mathbf{w} = \boldsymbol{\alpha}$. The MCM can be solved by linear programming.

Because the upper bound h in (6) is minimized in (5), the separating hyperplane is determined so that the maximum distance between the training data and the separating hyperplane is minimized.

The MCM, however, does not explicitly include the term related to the margin maximization. This makes the solution non-unique and unbounded.

To make the solution unique, in [10] the MCM and the LP SVM are fused and the resulting classifier is called minimal complexity LP SVM (MLP SVM):

$$\min Q(\boldsymbol{\alpha}, h, \boldsymbol{\xi}, b) = C_h\, h + \sum_{i=1}^{M} (C_\alpha\, |\alpha_i| + C\, \xi_i) \tag{8}$$

$$\text{s.t. } h \geq y_i \left(\sum_{j=1}^{M} \alpha_j\, K_{ij} + b \right) + \xi_i \geq 1,\ i = 1, \ldots M, \tag{9}$$

where C_h is the positive parameter and $C_\alpha = 1$. Deleting $C_h\, h$ in (8) and upper bound h in (9), we obtain the LP SVM. Setting $C_h = 1$ and $C_\alpha = 0$ in (8), we obtain the MCM.

3 Minimal Complexity L1 Support Vector Machines

In this section, we discuss the architecture and optimality conditions of the proposed classifiers.

3.1 Architecture

Similar to the MLP SVM, here we propose fusing the MCM given by (5) and (6) and the L1 SVM given by (2) and (3):

$$\min Q(\mathbf{w}, b, h, \boldsymbol{\xi}) = C_h\, h + \frac{1}{2}\, \|\mathbf{w}\|^2 + C \sum_{i=1}^{M} \xi_i \tag{10}$$

$$\text{s.t. } y_i\, (\mathbf{w}^\top\, \boldsymbol{\phi}(\mathbf{x}_i) + b) + \xi_i \geq 1,\ \xi_i \geq 0, \tag{11}$$

$$h \geq y_i\, (\mathbf{w}^\top\, \boldsymbol{\phi}(\mathbf{x}_i) + b),\ i = 1, \ldots, M, \tag{12}$$

$$h \geq 1, \tag{13}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)^\top$, C_h is the positive parameter to control the volume that the training data occupy, and h is the upper bound of the constraints. The upper bound defined by (6) is redefined by (12) and (13), which exclude ξ_i. This makes the KKT conditions for the upper bound simpler. We call (12) the upper

bound constraints and the above classifier minimum complexity L1 SVM (ML1 SVM).

In the following, we derive the dual problem of the ML1 SVM. Introducing the nonnegative Lagrange multipliers α_i, β_i, and η, we obtain

$$Q(\mathbf{w}, b, h, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \eta) = C_h\, h + \frac{1}{2}\,\|\mathbf{w}\|^2 - \sum_{i=1}^{M} \alpha_i \,\left(y_i\,(\mathbf{w}^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b) - 1 + \xi_i\right)$$

$$+ C \sum_{i=1}^{M} \xi_i - \sum_{i=1}^{M} \beta_i\,\xi_i - \sum_{i=1}^{M} \alpha_{M+i}\,\left(h - y_i\,(\mathbf{w}^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b)\right) - (h-1)\,\eta, \quad (14)$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{2M})^\top$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_M)^\top$.

For the optimal solution, the following KKT conditions are satisfied:

$$\frac{\partial Q(\mathbf{w}, b, h, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \eta)}{\partial \mathbf{w}} = \mathbf{0}, \quad \frac{\partial Q(\mathbf{w}, b, h, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \eta)}{\partial h} = 0, \quad (15)$$

$$\frac{\partial Q(\mathbf{w}, b, h, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \eta)}{\partial b} = 0, \quad \frac{\partial Q(\mathbf{w}, b, h, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \eta)}{\partial \boldsymbol{\xi}} = \mathbf{0}, \quad (16)$$

$$\alpha_i\,(y_i\,(\mathbf{w}^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b) - 1 + \xi_i) = 0,\ \alpha_i \geq 0, \quad (17)$$

$$\alpha_{M+i}\,\left(h - y_i\,(\mathbf{w}^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b)\right) = 0,\ \alpha_{M+i} \geq 0, \quad (18)$$

$$\beta_i\,\xi_i = 0, \quad \beta_i \geq 0, \quad \xi_i \geq 0,\ i = 1, \ldots, M, \quad (19)$$

$$(h-1)\,\eta = 0,\ h \geq 1,\ \eta \geq 0, \quad (20)$$

where $\mathbf{0}$ is the zero vector whose elements are zero. Equations (17) to (20) are called KKT complementarity conditions.

Using (14), we reduce (15) and (16), respectively, to

$$\mathbf{w} = \sum_{i=1}^{M} (\alpha_i - \alpha_{M+i})\,y_i\,\boldsymbol{\phi}(\mathbf{x}_i), \quad \sum_{i=1}^{M} \alpha_{M+i} = C_h - \eta, \quad (21)$$

$$\sum_{i=1}^{M} (\alpha_i - \alpha_{M+i})\,y_i = 0, \quad \alpha_i + \beta_i = C, \quad i = 1, \ldots, M. \quad (22)$$

Substituting (21) and (22) into (14), we obtain the following dual problem:

$$\max\ Q(\boldsymbol{\alpha}) = \sum_{i=1}^{M} (\alpha_i - \alpha_{M+i}) - \frac{1}{2} \sum_{i,j=1}^{M} (\alpha_i - \alpha_{M+i})$$

$$\times (\alpha_j - \alpha_{M+j})\,y_i\,y_j\,K(\mathbf{x}_i, \mathbf{x}_j) \quad (23)$$

$$\text{s.t.} \sum_{i=1}^{M} y_i\,(\alpha_i - \alpha_{M+i}) = 0, \quad (24)$$

$$C_h \geq \sum_{i=1}^{M} \alpha_{M+i}, \quad (25)$$

$$C\ \geq \alpha_i \geq 0,\ C_h \geq \alpha_{M+i} \geq 0,\ i = 1, \ldots, M. \quad (26)$$

If we delete variables α_{M+i} and C_h from the above optimization problem, we obtain the dual problem of the original L1 SVM.

For the solution of (23) to (26), positive α_i and α_{M+j} are support vectors.

We consider decomposing the above problem into two subproblems: 1) optimizing $\alpha_i\,(i = 1, \ldots, M)$ and 2) optimizing $\alpha_{M+i}\,(i = 1, \ldots, M)$. To make this possible, we eliminate the interference between α_i and α_{M+i} in (24) by

$$\sum_{i=1}^{M} y_i\,\alpha_i = 0, \quad \sum_{i=1}^{M} y_i\,\alpha_{M+i} = 0. \tag{27}$$

Then the optimization problem given by (23) to (26) is decomposed into the following two subproblems:

Subproblem 1: Optimization of α_i

$$\max\ Q(\boldsymbol{\alpha}^0) = \sum_{i=1}^{M}(\alpha_i - \alpha_{M+i}) - \frac{1}{2}\sum_{i,j=1}^{M}(\alpha_i - \alpha_{M+i}) \times(\alpha_j - \alpha_{M+j})\,y_i\,y_j\,K(\mathbf{x}_i, \mathbf{x}_j) \tag{28}$$

$$\text{s.t.} \sum_{i=1}^{M} y_i\,\alpha_i = 0, \tag{29}$$

$$C \geq \alpha_i \geq 0 \quad \text{for} \quad i = 1, \ldots, M, \tag{30}$$

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

Subproblem 2: Optimization of α_{M+i}

$$\max\ Q(\boldsymbol{\alpha}^M) = \sum_{i=1}^{M}(\alpha_i - \alpha_{M+i}) - \frac{1}{2}\sum_{i,j=1}^{M}(\alpha_i - \alpha_{M+i}) \times(\alpha_j - \alpha_{M+j})\,y_i\,y_j\,K(\mathbf{x}_i, \mathbf{x}_j) \tag{31}$$

$$\text{s.t.} \sum_{i=1}^{M} y_i\,\alpha_{M+i} = 0, \tag{32}$$

$$C_h \geq \sum_{i=1}^{M} \alpha_{M+i}, \tag{33}$$

$$C_h > \alpha_{M+i} \geq 0,\ i = 1, \ldots, M, \tag{34}$$

where $\boldsymbol{\alpha}^M = (\alpha_{M+1}, \ldots, \alpha_{2M})^\top$. Here we must notice that $\alpha_{M+i} \neq C_h$. If $\alpha_{M+i} = C_h$, from (32), at least

$$\sum_{j=1,\ldots,M,\ y_j \neq y_i} \alpha_{M+j} = C_h \tag{35}$$

is satisfied. This contradicts (33).

We solve Subproblems 1 and 2 alternatingly until the solution converges. Subproblem 1 is very similar to the L1 SVM and can be solved by the SMO (Sequential minimal optimization) combined with Newton's method [12]. Subproblem 2, which includes the constraint (33) can also be solved by a slight modification of the SMO combined with Newton's method.

3.2 KKT Conditions

To check the convergence of Subproblems 1 and 2, we use the KKT complementarity conditions (17) to (20). However, variables h and b, which are included in the KKT conditions, are excluded from the dual problem. Therefore, as with the L1 SVM [13], to make an accurate convergence test, the exact KKT conditions that do not include h and b need to be derived.

We rewrite (17) as follows:

$$\alpha_i \left(y_i \, b - y_i \, F_i + \xi_i \right) = 0, \; i = 1, \ldots, M, \tag{36}$$

where

$$F_i = y_i - \sum_{j=1}^{M} y_j \left(\alpha_j - \alpha_{M+j} \right) K(\mathbf{x}_i, \mathbf{x}_j). \tag{37}$$

We can classify the conditions of (36) into the following three cases:

1. $\alpha_i = 0$. Because $y_i \, b - y_i \, F_i + \xi_i \geq 0$ and $\xi_i = 0$, $y_i \, b \geq y_i \, F_i$, namely, $b \geq F_i$ if $y_i = 1$; $b \leq F_i$ if $y_i = -1$.
2. $C > \alpha_i > 0$. Because $\beta_i > 0$, $\xi_i = 0$ is satisfied. Therefore, $b = F_i$.
3. $\alpha_i = C$. Because $\beta_i = 0$, $\xi_i \geq 0$ is satisfied. Therefore, $y_i \, b \leq y_i \, F_i$, namely, $b \leq F_i$ if $y_i = 1$; $b \geq F_i$ if $y_i = -1$.

Then the KKT conditions for (36) are simplified as follows:

$$\bar{F}_i \geq b \geq \tilde{F}_i, \; i = 1, \ldots, M, \tag{38}$$

where

$$\tilde{F}_i = F_i \quad \text{if} \quad (y_i = 1, \alpha_i = 0), \quad C > \alpha_i > 0 \quad \text{or} \quad (y_i = -1, \alpha_i = C), \tag{39}$$

$$\bar{F}_i = F_i \quad \text{if} \quad (y_i = -1, \alpha_i = 0), \quad C > \alpha_i > 0 \quad \text{or} \quad (y_i = 1, \alpha_i = C). \tag{40}$$

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

$$b_{\text{low}} = \max_i \tilde{F}_i, \; b_{\text{up}} \; = \min_i \bar{F}_i. \tag{41}$$

If the KKT conditions are satisfied,

$$b_{\text{up}} \geq b_{\text{low}}. \tag{42}$$

The bias term is estimated to be

$$b_{\text{e}} = \frac{1}{2}(b_{\text{up}} + b_{\text{low}}), \tag{43}$$

where b_{e} is the estimate of the bias term using (17).

Likewise, using (37), (18) becomes

$$\alpha_{M+i}\,(h + y_i\,F_i - y_i\,b - 1) = 0,\ i = 1, \ldots, M. \tag{44}$$

Then the conditions for (18) are rewritten as follows:

1. $\alpha_{M+i} = 0$. From $h + y_i\,F_i - y_i\,b - 1 \geq 0$, we have $y_i\,b - h \leq y_i\,F_i - 1$, namely, $b - h \leq F_i - 1$ if $y_i = 1$; $b + h \geq F_i + 1$ if $y_i = -1$.
2. $C_h > \alpha_{M+i} > 0$. $y_i\,b - h = y_i\,F_i - 1$, namely, $b - h = F_i - 1$ if $y_i = 1$; $b + h = F_i + 1$ if $y_i = -1$.

The KKT conditions for (18) are simplified as follows:

$$\begin{aligned} &\text{if } y_i = -1,\ \bar{F}_i^{\,-} + 1 \geq b^- \geq \tilde{F}_i^{\,-} + 1, \\ &\text{if } y_i = 1,\ \bar{F}_i^{\,+} - 1 \geq b^+ \geq \tilde{F}_i^{\,+} - 1,\ i = 1, \ldots, M, \end{aligned} \tag{45}$$

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

$$\tilde{F}_i^{\,-} = F_i + 1 \quad \text{if} \quad y_i = -1, \tag{46}$$

$$\bar{F}_i^{\,-} = F_i + 1 \quad \text{if} \quad y_i = -1, C_h > \alpha_{M+i} > 0, \tag{47}$$

$$\tilde{F}_i^{\,+} = F_i - 1 \quad \text{if} \quad y_i = 1, C_h > \alpha_{M+i} > 0, \tag{48}$$

$$\bar{F}_i^{\,+} = F_i - 1 \quad \text{if} \quad y_i = 1. \tag{49}$$

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

$$\begin{aligned} b^-_{\text{low}} &= \max_i \tilde{F}_i^{\,-}, \quad b^+_{\text{low}} = \max_i \tilde{F}_i^{\,+}, \\ b^-_{\text{up}} &= \min_i \bar{F}_i^{\,-}, \quad b^+_{\text{up}} = \min_i \bar{F}_i^{\,+}. \end{aligned} \tag{50}$$

In general, the distributions of Classes 1 and 2 data are different. Therefore, the upper bounds of h for Classes 1 and 2 are different. This means that either of b^-_{up} ($\bar{F}_i^{\,-}$) and b^+_{low} ($\tilde{F}_i^{\,+}$) may not exist. But because of (32), both classes have at least one positive α_{M+i} each, and because of (44), the values of h for both classes can be different. This happens because we separate (24) into two equations as in (27). Then, if the KKT conditions are satisfied, both of the following inequalities hold

$$b^-_{\text{up}} \geq b^-_{\text{low}}, \quad b^+_{\text{up}} \geq b^+_{\text{low}}. \tag{51}$$

From the first inequality, the estimate of h, h^-_{e} for Class 2, is given by

$$h^-_{\text{e}} = -b_{\text{e}} + \frac{1}{2}(b^-_{\text{up}} + b^-_{\text{low}}). \tag{52}$$

From the second inequality, the estimate of h, h^+_{e} for Class 1, is given by

$$h^+_{\text{e}} = b_{\text{e}} - \frac{1}{2}(b^+_{\text{up}} + b^+_{\text{low}}). \tag{53}$$

3.3 Variant of Minimal Complexity Support Vector Machines

Subproblem 2 of the ML1 SVM is different from Subproblem 1 in that the former includes the inequality constraint given by (33). This makes the solution process makes more complicated. In this section, we consider making the solution process similar to that of Subproblem 1.

Solving Subproblem 2 results in obtaining h_{e}^{+} and h_{e}^{-}. We consider assigning separate variables h^{+} and h^{-} for Classes 1 and 2 instead of a single variable h. Then the complementarity conditions for h^{+} and h^{-} are

$$\begin{aligned} &(h^{+}-1)\,\eta^{+}=0,\ h^{+}\geq 1,\ \eta^{+}\geq 0,\ (h^{-}-1)\,\eta^{-}=0,\\ &h^{-}\geq 1,\ \eta^{-}\geq 0, \end{aligned} \tag{54}$$

where η^{+} and η^{-} are the Lagrange multipliers associated with h^{+} and h^{-}, respectively. To simplify Subproblem 2, we assume that $\eta^{+}=\eta^{-}=0$. This makes the equations corresponding to (33) equality constraints. Then the optimization problem given by (31) to (34) becomes

$$\max\ Q(\boldsymbol{\alpha}^{M})=\sum_{i=1}^{M}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{M}(\alpha_i-\alpha_{M+i})\,(\alpha_j-\alpha_{M+j})\,y_i\,y_j\,K(\mathbf{x}_i,\mathbf{x}_j) \tag{55}$$

$$\text{s.t.}\ \sum_{y_i=1,i=1}^{M}\alpha_{M+i}=C_h,\ \sum_{y_i=-1,i=1}^{M}\alpha_{M+i}=C_h, \tag{56}$$

$$C\geq\alpha_i\geq 0,\ C_h\geq\alpha_{M+i}\geq 0,\ i=1,\ldots,M. \tag{57}$$

Here, (32) is not necessary because of (56). We call the above architecture $\mathrm{ML1_v}$ SVM.

For the solution of the ML1 SVM, the same solution is obtained by the $\mathrm{ML1_v}$ SVM with the C_h value given by

$$C_h=\sum_{i=1,\ldots,M,y_i=1}\alpha_{M+i}=\sum_{i=1,\ldots,M,y_i=-1}\alpha_{M+i}. \tag{58}$$

However, the reverse is not true, namely, the solution of the $\mathrm{ML1_v}$ SVM may not be obtained by the ML1 SVM. As the C_h value becomes large, the value of η becomes positive for the ML1 SVM, but for the $\mathrm{ML1_v}$ SVM, the values of α_{M+i} are forced to become larger. But as the following computer experiments show, the performance difference is small.

4 Computer Experiments

In this section, we compare the generalization performance of the $\mathrm{ML1_v}$ SVM and ML1 SVM with the L1 SVM, MLP SVM [10], LS SVM, and ULDM [8] using two-class and multiclass problems.

4.1 Comparison Conditions

We determined the hyperparameter values using the training data by fivefold cross-validation, trained the classifier with the determined hyperparameter values, and evaluate the accuracy for the test data.

We trained the $\mathrm{ML1_v}$ SVM, ML1 SVM, and L1 SVM by SMO combined with Newton's method [12]. We trained the MLP SVM by the simplex method and the LS SVM and ULDM by matrix inversion.

Because RBF kernels perform well for most pattern classification applications, we used RBF kernels: $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2/m)$, where γ is the parameter to control the spread of the radius, and m is the number of inputs.

In cross-validation, we selected the γ values from $\{0.01, 0.1, 0.5, 1, 5, 10, 15, 20, 50, 100, 200\}$ and the C and C_h values from $\{0.1, 1, 10, 50, 100, 500, 1000, 2000\}$. For the ULDM, C value was selected from $\{10^{-12}, 10^{-10}, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1\}$.

For the L1 SVM, LS SVM, and ULDM, we determined the γ and C values by grid search. For the $\mathrm{ML1_v}$ SVM, ML1 SVM, and MLP SVM, to shorten computation time, first we determined the γ and C values with $C_h = 1$ ($C_h = 0.1$ for the MLP SVM) by grid search and then we determined the C_h value by line search fixing the γ and C values with the determined values.

After model selection, we trained the classifier with the determined hyperparameter values and calculated the accuracy for the test data. For two-class problems we calculated the average accuracies and their standard deviations, and performed Welch's t test with the confidence level of 5%.

4.2 Two-Class Problems

Table 1 lists accuracies for the two-class problems. In the first column, I/Tr/Te denotes the numbers of input variables, training data, and test data. Except for the image and splice problems, each problem has 100 training and test data pairs. For the image and splice problems, 20 pairs.

In the table, for each classifier and each classification problem, the average accuracy and the standard deviation are shown. For each problem the best average accuracy is shown in bold and the worst, underlined. The "+" and "−" symbols at the accuracy show that the $\mathrm{ML1_v}$ SVM is statistically better and worse than the classifier associated with the attached symbol, respectively. The "Average" row shows the average accuracy of the 13 problems for each classifier and "B/S/W" denotes the number of times that the associated classifier showed the best, the second best, and the worst accuracies. The "W/T/L" row denotes the number of times that the $\mathrm{ML1_v}$ SVM is statistically better than, comparable to, and worse than the associated classifier.

According to the "W/T/L" row, the $\mathrm{ML1_v}$ SVM is statistically better than the MLP SVM but is comparable to other classifiers. From the "Average" measure, the ULDM is the best and the $\mathrm{ML1_v}$ SVM, the second. However, the differences of the measures among the $\mathrm{ML1_v}$ SVM, ML1 SVM, and L1 SVM are

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

Problem I/Tr/Te	$ML1_v$ SVM	ML1 SVM	L1 SVM	MLP SVM	LS SVM	ULDM
Banana 2/400/4900	89.11 ± 0.70	**89.18** ± 0.70	89.17 ± 0.72	89.07 ± 0.73	89.17 ± 0.66	89.12 ± 0.69
Cancer 9/200/77	73.12 ± 4.43	73.03 ± 4.45	73.03 ± 4.51	72.81 ± 4.59	73.13 ± 4.68	**73.70** ± 4.42
Diabetes 8/468/300	76.33 ± 1.94	76.17 ± 2.25	76.29 ± 1.73	76.05 ± 1.74	76.19 ± 2.00	**76.51** ± 1.95
Flare-solar 9/666/400	**66.99** ± 2.16	66.98 ± 2.14	**66.99** ± 2.12	66.62 ± 3.10	$66.25^{+} \pm 1.98$	$66.28^{+} \pm 2.05$
German 20/700/300	75.97 ± 2.21	75.91 ± 2.03	75.95 ± 2.24	75.63 ± 2.57	76.10 ± 2.10	**76.12** ± 2.3
Heart 13/170/100	**82.96** ± 3.25	82.84 ± 3.26	82.82 ± 3.37	82.52 ± 3.27	82.49 ± 3.60	82.57 ± 3.64
Image 18/1300/1010	97.27 ± 0.46	97.29 ± 0.44	97.16 ± 0.41	$96.47^{+} \pm 0.87$	**97.52** ± 0.54	97.16 ± 0.68
Ringnorm 20/400/7000	97.97 ± 1.11	98.12 ± 0.36	98.14 ± 0.35	97.97 ± 0.37	**98.19** ± 0.33	98.16 ± 0.35
Splice 60/1000/2175	88.99 ± 0.83	89.05 ± 0.83	88.89 ± 0.91	$86.71^{+} \pm 1.27$	88.98 ± 0.70	**89.16** ± 0.53
Thyroid 5/140/75	**95.37** ± 2.50	95.32 ± 2.41	95.35 ± 2.44	95.12 ± 2.38	95.08 ± 2.55	95.15 ± 2.27
Titanic 3/150/2051	77.40 ± 0.79	77.37 ± 0.81	77.39 ± 0.74	77.41 ± 0.77	77.39 ± 0.83	**77.46** ± 0.91
Twonorm 20/400/7000	97.38 ± 0.25	97.36 ± 0.28	97.38 ± 0.26	$97.13^{+} \pm 0.29$	**97.43** ± 0.27	97.41 ± 0.26
Waveform 21/400/4600	89.67 ± 0.75	89.72 ± 0.73	89.76 ± 0.66	$89.39^{+} \pm 0.53$	$90.05^{-} \pm 0.59$	**90.18**$^{-} \pm 0.54$
Average	85.27	85.26	85.26	84.84	85.23	**85.31**
B/S/W	3/1/0	1/3/1	1/2/0	0/1/9	3/4/3	6/1/0
W/T/L	—	0/13/0	0/13/0	4/9/0	1/11/1	1/11/1

small. From the "B/S/W" measure, the ULDM is the best and the LS SVM is the second best.

4.3 Multiclass Problems

Table 2 shows the accuracies for the ten multiclass problems. The symbol "C" in the first column denotes the number of classes. Unlike the two-class problems, each multiclass problem has only one training and test data pair.

We used fuzzy pairwise (one-vs-one) classification for multiclass problems [2]. In the table, for each problem, the best accuracy is shown in bold, and the worst, underlined. For the MLP SVM, the accuracies for the thyroid, MNIST, and letter problems were not available.

Among the ten problems, the accuracies of the $ML1_v$ SVM and ML1 SVM were better than or equal to those of the L1 SVM for nine and seven problems,

Table 2. Accuracies of the test data for the multiclass problems

Problem I/C/Tr/Te	$ML1_v$ SVM	ML1 SVM	L1 SVM	MLP SVM	LS SVM	ULDM
Numeral 12/10/810/820 [2]	**99.76**	**99.76**	**99.76**	99.27	99.15	99.39
Thyroid 21/3/3772/3428 [14]	97.23	**97.26**	**97.26**	—	95.39	95.57
Blood cell 13/12/3097/3100 [2]	93.55	93.19	93.16	93.36	94.23	**94.61**
Hiragana-50 50/39/4610/4610 [2]	98.98	99.46	99.00	98.96	**99.48**	98.92
Hiragana-13 13/38/8375/8356 [2]	99.79	99.89	99.79	**99.90**	99.87	**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 [14]	91.85	91.85	91.90	91.10	91.95	**92.25**
USPS 256/10/7291/2007 [15]	95.42	**95.47**	95.27	95.17	**95.47**	95.42
MNIST 784/10/10000/60000 [16]	96.96	96.96	96.55	—	96.99	**97.03**
Letter 16/26/16000/4000 [14]	97.95	**98.03**	97.85	—	97.88	97.75
Average	97.15	**97.19**	97.05	—	97.04	97.08
B/S/W	2/1/1	5/1/0	3/0/3	2/0/2	3/3/2	5/0/2

respectively. In addition, the best average accuracy was obtained for the ML1 SVM and the second best, the $ML1_v$ SVM. This is very different from the two-class problems where the difference was very small.

5 Conclusions

In this paper, to solve the problem of the non-unique solution of the MCM, and to improve the generalization ability of the L1 SVM, we fused the MCM and the L1 SVM. We derived two dual subproblems: the first subproblem corresponds to the L1 SVM and the second subproblem corresponds to minimizing the upper bound. We further modified the second subproblem by converting the inequality constraint into two equality constraints. We call this architecture $ML1_v$ SVM and the original architecture, ML1 SVM.

According to computer experiments for two-class problems, the average accuracy of the $ML1_v$ SVM is statistically comparable to that of the ML1 SVM and L1 SVM. For multiclass problems, the $ML1_v$ SVM and ML1 SVM generalized better than the L1 SVM.

Acknowledgment. This work was supported by JSPS KAKENHI Grant Number 19K04441.

References

1. Vapnik, V.N.: Statistical Learning Theory. Wiley, Hobooken (1998)
2. Abe, S.: Support Vector Machines for Pattern Classification, 2nd edn. Springer, London (2010). https://doi.org/10.1007/978-1-84996-098-4
3. Abe, S.: Training of support vector machines with Mahalanobis kernels. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3697, pp. 571–576. Springer, Heidelberg (2005). https://doi.org/10.1007/11550907_90
4. Reitmaier, T., Sick, B.: The responsibility weighted Mahalanobis kernel for semi-supervised training of support vector machines for classification. Inf. Sci. **323**, 179–198 (2015)
5. Lanckriet, G.R.G., et al.: Learning the kernel matrix with semidefinite programming. J. Mach. Learn. Res. **5**, 27–72 (2004)
6. Peng, X., Xu, D.: Twin Mahalanobis distance-based support vector machines for pattern recognition. Inf. Sci. **200**, 22–37 (2012)
7. Zhang, T., Zhou, Z.-H.: Large margin distribution machine. In: Proceedings of the 12th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 313–322 (2014)
8. Abe, S.: Unconstrained large margin distribution machines. Pattern Recogn. Lett. **98**, 96–102 (2017)
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 the IJCNN 2019 (2019)
11. Abe, S.: Sparse least squares support vector training in the reduced empirical feature space. Pattern Anal. Appl. **10**(3), 203–214 (2007). https://doi.org/10.1007/s10044-007-0062-1

12. Abe, S.: Fusing sequential minimal optimization and Newton's method for support vector training. Int. J. Mach. Learn. Cybern. **7**(3), 345–364 (2016). https://doi.org/10.1007/s13042-014-0265-x
13. Keerthi, S.S., Gilbert, E.G.: Convergence of a generalized SMO algorithm for SVM classifier design. Mach. Learn. **46**(1–3), 351–360 (2002). https://doi.org/10.1023/A:1012431217818
14. http://www.ics.uci.edu/~mlearn/MLRepository.html
15. https://www.kaggle.com/bistaumanga/usps-dataset
16. http://yann.lecun.com/exdb/mnist/