Improving Generalization Abilities of Maximal Average Margin Classifiers

Abe, Shigeo

(Citation)
Lecture Notes in Computer Science,9896:29-41

(Issue Date)
2016

(Resource Type)
journal article

(Version)
Accepted Manuscript

Improving Generalization Abilities of Maximal Average Margin Classifiers

Shigeo Abe

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

Abstract. Maximal average margin classifiers (MAMCs) maximize the average margin without constraints. Although training is fast, the generalization abilities are usually inferior to support vector machines (SVMs). To improve the generalization abilities of MAMCs, in this paper, we propose optimizing slopes and bias terms of separating hyperplanes after the coefficient vectors of the hyperplanes are obtained. The bias term is optimized so that the number of misclassifications is minimized. To optimized the slope, we introduce a weight to the average of mapped training data for one class and optimize the weight by cross-validation. To improve the generalization ability further, we propose equally constrained MAMCs and show that they reduce to least squares SVMs. Using two-class problems, we show that the generalization ability of the unconstrained MAMCs are inferior to those of the constrained MAMCs and SVMs.

1 Introduction

Since the introduction of support vector machines (SVMs) [1, 2] various variants have been developed to improve the generalization ability. Because SVMs do not assume a specific data distribution, a priori knowledge on the data distribution can improve the generalization ability. The Mahalanobis distance, instead of the Euclidean distance is useful for this purpose. One approach reformulates SVMs so that the margin is measured by the Mahalanobis distance [3–7], and another approach uses Mahalanobis kernels, which calculate the kernel value according to the Mahalanobis distance [8–13].

In SVMs, the minimum margin is maximized. But in AdaBoost [14], the margin distribution, instead of the minimum margin, has been known to be important in improving the generalization ability [15, 16].

Several approaches have been proposed to control the margin distribution in SVM-like classifiers [17–22]. In [18], a maximum average margin classifier (MAMC) is proposed, in which instead of maximizing the minimum margin, the margin mean for the training data is maximized without slack variables. In [21, 22], in addition to maximizing the margin mean, the margin variance is minimized and the classifier is called large margin distribution machine (LDM).

According to the computer experiments in [21], the generalization ability of MAMCs is inferior to SVMs and LDMs.

In this paper, we clarify why MAMCs perform poorly for some classification problems and propose two methods to improve the generalization ability. Because the MAMC does not include constraints associated with training data, the determined bias term depends only on the difference between the numbers of training data for the two classes. To solve this problem, after the weight vector is obtained by the MAMC, we optimize the bias term so that the classification error is minimized. Then to improve the generalization ability further, we introduce a weight parameter to the average vector of one class and determine the parameter value by cross-validation. This results in optimizing the slope of the separating hyperplane. To improve the generalization ability further, we define the equality-constrained MAMC (EMAMC), which is shown to be equivalent to the least squares (LS) SVM. Using two-class problems, we show that the generalization ability of the unconstrained MAMCs with the optimized bias term and slopes are inferior to that of the EMAMC.

In Section 2, we explain the architecture of the MAMC and clarify the problems of MAMC. Then, we propose bias term optimization and slope optimization and develop the EMAMC. In Section 3, we compare the generalization abilities of the MAMC with those of the proposed MAMC with optimized bias terms and slopes, the EMAMC, and the SVM.

2 Maximum Average Margin Classifiers

2.1 Architecture

In the following we explain the maximum average margin classifiers (MAMCs) according to [18].

We consider a classification problem with M training input-output pairs $\{\mathbf{x}_i, y_i\}$ $(i = 1, \ldots, M)$, where $\mathbf{x}_i$ are m-dimensional training inputs and belong to Class 1 or 2 and the associated labels are $y_i = 1$ for Class 1 and -1 for Class 2. We map the m-dimensional input vector $\mathbf{x}$ into the l-dimensional feature space using the nonlinear vector function $\boldsymbol{\phi}(\mathbf{x})$. In the feature space, we determine the decision function that separates Class 1 data from Class 2 data:

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

where $\mathbf{w}$ is the l-dimensional vector, $\top$ denotes the transpose of a vector (matrix), and b is the bias term.

The margin of $\mathbf{x}_i$, δ_i, which is the distance from the hyperplane, is given by

$$\delta_i = y_i\, (\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b)/\|\mathbf{w}\|. \tag{2}$$

Under the assumption of

$$\mathbf{w}^\top\, \mathbf{w} = 1, \tag{3}$$

(2) becomes

$$\delta_i = y_i\, (\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b). \tag{4}$$

With $b = 0$, the MAMC, which maximizes the average margin, is defined by

$$\text{maximize} \quad Q(\mathbf{w}) = \frac{1}{M} \sum_{i=1}^{M} y_i \, \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) \tag{5}$$

$$\text{subject to} \quad \mathbf{w}^\top \mathbf{w} = 1. \tag{6}$$

Introducing the Lagrange multiplier $\lambda \, (> 0)$, we obtain the unconstrained optimization problem:

$$\text{maximize} \quad Q(\mathbf{w}) = \frac{1}{M} \sum_{i=1}^{M} y_i \, \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) - \frac{\lambda}{2} (\mathbf{w}^\top \mathbf{w} - 1). \tag{7}$$

Taking the derivative of Q with respect to $\mathbf{w}$, we obtain the optimal $\mathbf{w}$:

$$\lambda \, \mathbf{w} = \frac{1}{M} \sum_{i=1}^{M} y_i \, \boldsymbol{\phi}(\mathbf{x}_i). \tag{8}$$

In [18], λ is determined using (6) and (8), but λ can take on any positive value because that does not change the decision boundary. Therefore, in the following we set $\lambda = 1$.

In calculating the decision function given by (1), we use kernels $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}^\top(\mathbf{x}) \, \boldsymbol{\phi}(\mathbf{x})$ to avoid treating the variables in the feature space explicitly.

The resulting decision function $f(\mathbf{x})$ with $b = 0$ is given by

$$f(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} y_i \, K(\mathbf{x}, \mathbf{x}_i). \tag{9}$$

Among several kernels, radial basis function (RBF) kernels are widely used and thus in the following study we use RBF kernels:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2 / m), \tag{10}$$

where m is the number of inputs for normalization and γ is to control a spread of a radius.

2.2 Problems with MAMCs

The MAMC is derived without a bias term, i.e., $b = 0$. To include the bias term we change (7) to

$$\text{maximize} \quad Q(\mathbf{w}, b) = \frac{1}{M} \sum_{i=1}^{M} y_i \, (\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b) - \frac{1}{2} (\mathbf{w}^\top \mathbf{w} + b^2). \tag{11}$$

Here, we replace λ with 1 and delete the constant term. Then, Q is maximized when

$$\mathbf{w} = \frac{1}{M} \sum_{i=1}^{M} y_i \, \boldsymbol{\phi}(\mathbf{x}_i), \tag{12}$$

$$b = \frac{1}{M} \sum_{i=1}^{M} y_i. \tag{13}$$

From (13), b is determined by the deference of the numbers of the data belonging to Classes 1 and 2, not by the distributions of the data belonging to the two classes. And if the numbers are the same, $b = 0$, irrespective of $\mathbf{x}_i \, (i = 1, \ldots, M)$. This occurs because the coefficient of b becomes zero in (11); the value of b does not affect optimality of the solution.

This means that the constraints are lacking for determining the optimal value of b. Similar to SVMs, the addition of inequality or equality constraints for the training data may solve the problem, which will be discussed later.

2.3 Bias Term Optimization

In this section we propose two-stage training; in the first stage, we determine the coefficient vector $\mathbf{w}$ by (12), and in the second stage, we optimize the value of b by

$$\text{minimize} \quad E_{\mathrm{R}} = \sum_{i=1}^{M} I(\xi_i) \tag{14}$$

$$\begin{aligned} \text{subject to} \quad & y_i \, (\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b) \geq \rho - \xi_i \\ & \rho > 0, \quad \xi_i \geq 0, \end{aligned} \tag{15}$$

where E_{R} is the number of misclassifications, ρ is a positive constant, $\xi_i \, (\geq 0)$ is a slack variable, $I(\xi_i) = 0$ for $\xi_i = 0$ and $I(\xi_i) = 1$ for $\xi_i > 0$. If there are multiple b values that minimize (14), we break the tie by

$$\text{minimize} \quad E_{\mathrm{S}} = \sum_{i=1}^{M} \xi_i, \tag{16}$$

where E_{S} is the sum of slack variables.

First we consider the case where the classification problem is separable in the feature space. Suppose that

$$\max_{\substack{j=1,\ldots,M \\ y_j=-1}} \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_j) < \min_{\substack{i=1,\ldots,M \\ y_i=1}} \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) < 0 \tag{17}$$

is satisfied, where training data belonging to Class 2 are correctly classified but some of the training data belonging to Class 1 are misclassified. Because of the first inequality in (17), by setting a proper value to b, all the training data are correctly classified.

Let

$$j = \arg_j \max_{\substack{j=1,\dots,M\\ y_j=-1}} \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_j), \quad i = \arg_i \min_{\substack{i=1,\dots,M\\ y_i=1}} \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i). \tag{18}$$

Then, from (15), to make $\mathbf{x}_i$ and $\mathbf{x}_j$ be correctly classified with margin ρ,

$$\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b = \rho, \quad -(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_j) + b) = \rho \tag{19}$$

must be satisfied. Thus,

$$b = -\frac{1}{2}(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_j)), \quad \rho = \frac{1}{2}(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_j)). \tag{20}$$

The above equations are also valid when

$$0 < \max_{\substack{j=1,\dots,M\\ y_j=-1}} \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_j) < \min_{\substack{i=1,\dots,M\\ y_i=1}} \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i), \tag{21}$$

where some of the training data for Class 2 are misclassified.

It is clear that (20) satisfies $E_{\mathrm{R}} = E_{\mathrm{S}} = 0$ and that ρ is the maximum.

Now consider the inseparable case. Let the misclassified training data for Class 1 be $\mathbf{x}_{i_k}$ $(k = 1, \dots, p)$ and

$$\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_{i_1}) \le \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_{i_2}) \le \cdots \le \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_{i_p}) \le 0. \tag{22}$$

Likewise, let the misclassified training data for Class 2 be $\mathbf{x}_{j_k}$ $(k = 1, \dots, n)$ and

$$0 \le \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_{j_1}) \le \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_{j_2}) \le \cdots \le \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_{j_n}). \tag{23}$$

Similar to the separable case, it is clear that the optimal b occurs at (20) with $i = i_k$ $(k \in \{1, \dots, p\})$ and j being given by

$$j = \arg_j \max_{\substack{\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_j) < \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_{i_k})\\ y_j=-1, j=1,\dots,M}} \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_j) \tag{24}$$

or with $j = j_k$ $(k \in \{1, \dots, n\})$ and i being given by

$$i = \arg_i \max_{\substack{\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_{j_k}) < \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i)\\ y_i=1, i=1,\dots,M}} \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i). \tag{25}$$

Let $E_{\mathrm{R}}(i,j)$ and $E_{\mathrm{S}}(i,j)$ denote that E_{R} and E_{S} are evaluated with b determined using $\mathbf{x}_i$ and $\mathbf{x}_j$ by (20), where $i = i_k$ $(k \in \{1, \dots, p\})$ and j is given by (24) or $j = j_k$ $(k \in \{1, \dots, n\})$ and i is given by (25). For each pair of i and j, we calculate $E_{\mathrm{R}}(i,j)$ and select the value of b that minimizes $E_{\mathrm{R}}(i,j)$. If there are multiple pairs of i and j, we select the value of b that minimizes $E_{\mathrm{S}}(i,j)$.

2.4 Extension of MAMCs

Characteristics of Solutions Rewriting (8) with $\lambda = 1$,

$$\mathbf{w} = \frac{M_+}{M}\bar{\boldsymbol{\phi}}_+ - \frac{M_-}{M}\bar{\boldsymbol{\phi}}_- \qquad (26)$$

where

$$\bar{\boldsymbol{\phi}}_+ = \frac{1}{M_+}\sum_{\substack{i=1\\ y_i=1}}^{M} \boldsymbol{\phi}(\mathbf{x}_i), \quad \bar{\boldsymbol{\phi}}_- = \frac{1}{M_-}\sum_{\substack{i=1\\ y_i=-1}}^{M} \boldsymbol{\phi}(\mathbf{x}_i), \qquad (27)$$

and $\bar{\boldsymbol{\phi}}_+$ and $\bar{\boldsymbol{\phi}}_-$ are the averages of the mapped training data belonging to Classes 1 and 2, respectively, and M_+ and M_- are the numbers of training data belonging to Classes 1 and 2, respectively.

If $M_+ = M_-$, $\mathbf{w}$ is the vector which is from $\bar{\boldsymbol{\phi}}_-/2$ to $\bar{\boldsymbol{\phi}}_+/2$. Therefore the decision function is orthogonal to the vector. If $M_+ \neq M_-$, the decision function is orthogonal to $\bar{\boldsymbol{\phi}}_+ - (M_-/M_+)\,\bar{\boldsymbol{\phi}}_-$

Slope Optimization To control the decision function, we introduce a positive hyperparameter C_{m} as follows:

$$\mathbf{w} = \frac{M_+}{M}\bar{\boldsymbol{\phi}}_+ - \frac{C_{\mathrm{m}}\,M_-}{M}\bar{\boldsymbol{\phi}}_-, \qquad (28)$$

where C_{m} works to lengthen or shorten the length of vector $\bar{\boldsymbol{\phi}}_-$ according to whether $C_{\mathrm{m}} > 1$ or $0 < C_{\mathrm{m}} < 1$. Therefore, by changing the value of C_{m}, the slope of the decision function is changed.

Then the decision function becomes

$$f(\mathbf{x}) = \frac{1}{M}\sum_{i=1, y_i=1}^{M} K(\mathbf{x}, \mathbf{x}_i) - \frac{C_{\mathrm{m}}}{M}\sum_{i=1, y_i=-1}^{M} K(\mathbf{x}, \mathbf{x}_i) + b. \qquad (29)$$

We determine the value of C_{m} by cross-validation.

In k-fold cross-validation, we divide the training data set into k almost-equal-size subsets and train the classifier using the $k-1$ subsets and test the trained classifier using the remaining subset. We iterate this procedure k times for different combinations and calculate the classification error .

Calculation of the classification error for a given C_{m} value is as follows:

1. Calculate (29) with $b = 0$ using the $k-1$ subsets.
2. Calculate the bias term using the method discussed in Section 2.3.
3. Calculate the classification error for the remaining subset using the decision function generated in Steps 1 and 2.

Repeat the above procedure for the k different combinations and calculate the classification error for the decision function.

For a given set of C_{m} values, we calculate the classification errors and select the value of C_{m} with the minimum classification error.

2.5 Equality-Constrained MAMCs

To improve the generalization ability of MAMCs further, we consider equality-constrained MAMCs (EMAMCs) as follows:

$$\text{maximize} \quad Q(\mathbf{w}, b) = -\frac{1}{2}\mathbf{w}^\top \mathbf{w} + \frac{C_\mathrm{a}}{M}\sum_{i=1}^{M} y_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b\right) - \frac{C}{2}\sum_{i=1}^{M} \xi_i^2 \tag{30}$$

$$\text{subject to} \quad y_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b\right) = 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, M, \tag{31}$$

where C_a and C are parameters to control the trade-off between the generalization ability and the classification error for the training data, and ξ_i are the slack variables for $\mathbf{x}_i$.

We solve (30) and (31) in the empirical feature space [2] and define

$$\boldsymbol{\phi}(\mathbf{x}) = (K(\mathbf{x}, \mathbf{x}_1), \ldots, K(\mathbf{x}, \mathbf{x}_M))^\top. \tag{32}$$

Solving (31) for ξ_i and substituting it into (30), we obtain the unconstrained optimization problem:

$$\begin{aligned}\text{maximize} \quad Q(\mathbf{w}, b) = & -\frac{1}{2}\mathbf{w}^\top \mathbf{w} + \frac{C_\mathrm{a}}{M}\sum_{i=1}^{M} y_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b\right) \\ & -\frac{C}{2}\sum_{i=1}^{M} (1 - y_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b\right))^2. \end{aligned} \tag{33}$$

If we delete the second term (the average margin) in the above equation, the optimization problem result in the least squares (LS) SVM defined in the empirical feature space [2].

Taking the partial derivative of (33) with respect to $\mathbf{w}$ and b and setting the results to zero, we obtain the optimality conditions:

$$\left(1 + C\sum_{i=1}^{M} \boldsymbol{\phi}(\mathbf{x}_i)\, \boldsymbol{\phi}^\top(\mathbf{x}_i)\right)\mathbf{w} + C\sum_{i=1}^{M} y_i\, \boldsymbol{\phi}(\mathbf{x}_i)\, b = \left(\frac{C_\mathrm{a}}{M} + C\right)\sum_{i=1}^{M} y_i\, \boldsymbol{\phi}(\mathbf{x}_i), \tag{34}$$

$$C\sum_{i=1}^{M} \boldsymbol{\phi}(\mathbf{x}_i)\,\mathbf{w} + C\, M\, b = \left(\frac{C_\mathrm{a}}{M} + C\right)\sum_{i=1}^{M} y_i. \tag{35}$$

The above optimality conditions can be solved for $\mathbf{w}$ and b by matrix inversion. The coefficient $(C_\mathrm{a}/M + C)$ can be deleted because it is a scaling factor and does not change the decision boundary. Then, because C_a is not included in the left-hand sides of (34) and (35), the value of C_a does not influence the location of the decision boundary. This means that the second term in (33) can be safely deleted.

In addition, if we delete the $\mathbf{w}^\top \mathbf{w}$ term from (33), all the terms in the left-hand sides of (34) and (35) include C, thus C can be deleted; C does not work to control the trade-off. Therefore, the $\mathbf{w}^\top \mathbf{w}$ term is essential.

Accordingly, dividing (34) and (35) by C and deleting the constant term $(C_{\mathrm{a}}/C\,M+1)$ from the right-hand sides of (34) and (35), we obtain

$$\left(\frac{1}{C}+\sum_{i=1}^{M}\boldsymbol{\phi}(\mathbf{x}_i)\,\boldsymbol{\phi}^{\top}(\mathbf{x}_i)\right)\mathbf{w}+\sum_{i=1}^{M}y_i\,\boldsymbol{\phi}(\mathbf{x}_i)\,b=\sum_{i=1}^{M}y_i\,\boldsymbol{\phi}(\mathbf{x}_i), \tag{36}$$

$$\sum_{i=1}^{M}\boldsymbol{\phi}(\mathbf{x}_i)\,\mathbf{w}+M\,b=\sum_{i=1}^{M}y_i. \tag{37}$$

The above formulation is exactly the same as the LS SVM defined in the empirical feature space. Therefore, the EMAMC results in the LS SVM.

3 Performance Evaluation

3.1 Experimental Conditions

We compared the proposed MAMC including the EMAMC (LS SVM) with the plain MAMC and the L1 SVM using two-class data sets [23]. The L1 SVM we used is as follows:

$$\text{maximize} \quad Q(\boldsymbol{\alpha})=\sum_{i=1}^{M}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{M}\alpha_i\alpha_j\,y_i\,y_jK(\mathbf{x}_i,\mathbf{x}_j) \tag{38}$$

$$\text{subject to} \quad \sum_{i=1}^{M}y_i\,\alpha_i=0, \quad 0\le\alpha_i\le C \quad \text{for} \quad i=1,...,M, \tag{39}$$

where α_i are Lagrange multipliers associated with $\mathbf{x}_i$ and $C\,(>0)$ is a margin parameter that controls the trade-off between the classification error of the training data and the generalization ability.

Table 1 lists the numbers of inputs, training data, test data, and data set pairs of two class problems. Each data set pair consists of the training data set and the test data set. We trained classifiers using the training data set and evaluated the performance using the test data set. Then we calculated the average accuracy and the standard deviation for all the data set pairs. We determined the parameter values by fivefold cross-validation. We selected the γ value of the RBF kernels from $\{0.01, 0.1, 0.5, 1, 5, 10, 15, 20, 50, 100, 200, 300, 400, 500, 600, 700\}$. For the C_{m}, we selected from $\{0.05, 0.1, 0.2, \ldots, 0.9, 1.0, 1.1111, \ldots, 20\}$. For the EMAMC and L1 SVM, we selected the γ value from 0.01 to 200 and the C value, from $\{0.1, 1, 10, 50, 100, 500, 1000, 2000\}$. We trained the L1 SVM using SMO-NM [24], which fuses SMO (Sequential minimal optimization) and NM (Newton's method).

3.2 Results

Table 2 shows the average accuracies and their standard deviations of the six classifiers with RBF kernels. In the table, MAMC is given by (12) and (13)

Table 1. Benchmark data sets for two-class problems

Data	Inputs	Train	Test	Sets
Banana	2	400	4,900	100
Breast cancer	9	200	77	100
Diabetes	8	468	300	100
Flare-solar	9	666	400	100
German	20	700	300	100
Heart	13	170	100	100
Image	18	1,300	1,010	20
Ringnorm	20	400	7,000	100
Splice	60	1,000	2,175	20
Thyroid	5	140	75	100
Titanic	3	150	2,051	100
Twonorm	20	400	7,000	100
Waveform	21	400	4,600	100

and the γ value is optimized by cross-validation. In MAMC$_{\rm b}$, the bias term is optimized as discussed in Section 2.3. In MAMC$_{\rm bs}$, after γ value is optimized with $C_{\rm m} = 1$, the C_m value is optimized. We call this strategy line search in contrast to grid search. In MAMC$_{\rm bsg}$, the γ and $C_{\rm m}$ values are optimized by grid search.

Among the six classifiers including the L1 SVM the best average accuracy is shown in bold and the worst average accuracy is underlined. The "Average" row shows the average accuracy of the 13 average accuracies and the two numerals in the parentheses show the numbers of the best and worst accuracies in the order. We performed Welch's t test with the confidence intervals of 95%. The "W/T/L" row shows the results; W, T, and L denote the numbers that the MAMC$_{\rm bs}$ shows statistically better than, the same as, and worse than the remaining five classifiers, respectively.

From the "Average" row, the EMAMC performed best in the average accuracy and the L1 SVM the second best. The difference between MAMC$_{\rm bsg}$ and MAMC$_{\rm bs}$ is small. The MAMC is the worst. From the "W/T/L" row, the accuracies of the MAMC$_{\rm bs}$ and the MAMC$_{\rm bsg}$ are statistically comparable and the accuracy of the MAMC$_{\rm bs}$ is slightly better than that of the the MAMC$_{\rm b}$ but always better than that of the MAMC. The accuracy of the MAMC$_{\rm bs}$ is worse than that of the EMAMC and L1 SVM.

In Section 2.2, we clarified that the bias term is not optimized by the original MAMC formulation. This is exemplified by the experiments. By optimizing the bias term as proposed in Section 2.3, the accuracy is improved drastically. The effect of slope optimization to the accuracy is small. However, by the bias term and slope optimization, the generalization ability is still below that of EMAMC or L1 SVM. This indicates that the equality or inequality constraints are essential in realizing the high generalization ability.

We measured the average CPU time per data set including time for model selection by fivefold cross-validation, training a classifier, and classifying the

Table 2. Accuracy comparison for the two-class problems

Data	MAMC$_{\text{bs}}$	MAMC$_{\text{bsg}}$	MAMC$_{\text{b}}$	MAMC	EMAMC	L1 SVM
Banana	88.46±0.85	88.49±0.86	88.51±0.73	$\underline{59.69}$±9.17	89.13±0.63	**89.17**±0.72
B. cancer	74.14±4.33	74.00±4.43	**74.27**±4.36	$\underline{71.19}$±4.53	73.57±4.55	73.03±4.51
Diabetes	73.03±2.26	72.19±2.49	72.66±2.32	$\underline{65.22}$±2.15	**76.67**±1.76	76.29±1.73
Flare-solar	66.10±2.00	65.64±2.05	66.35±2.03	$\underline{59.48}$±5.83	66.25±2.00	**66.99**±2.12
German	75.53±2.18	75.93±2.21	$\underline{69.51}$±2.28	70.18±1.95	**76.27**±2.04	75.95±2.24
Heart	81.00±3.58	80.99±3.24	81.32±3.38	$\underline{58.87}$±8.84	82.70±3.70	**82.82**±3.37
Image	93.59±1.22	94.11±1.24	92.90±1.12	$\underline{56.81}$±1.10	96.97±0.74	**97.16**±0.41
Ringnorm	**98.27**±0.27	98.25±0.30	**98.27**±0.25	$\underline{70.26}$±21.99	98.04±0.43	98.14±0.35
Splice	84.43±1.11	85.17±0.88	84.08±0.99	$\underline{54.57}$±8.76	**89.07**±0.59	88.89±0.91
Thyroid	95.15±2.24	95.41±2.30	95.11±2.17	$\underline{70.25}$±4.36	**95.43**±2.35	95.35±2.44
Titanic	77.68±0.84	**77.70**±1.03	77.64±0.84	$\underline{67.69}$±0.30	77.69±0.82	77.39±0.74
Twonorm	97.28±0.31	97.21±0.40	97.35±0.27	$\underline{79.25}$±16.20	**97.41**±0.23	97.38±0.26
Waveform	88.93±1.53	89.23±1.29	88.84±1.64	$\underline{67.07}$±0.19	**90.20**±0.50	89.76±0.66
Average	84.12 (1/0)	84.18 (2/0)	83.60 (2/1)	65.43 (0/11)	85.34 (6/0)	85.26 (4/0)
W/T/L	—	1/11/1	1/12/0	13/0/0	1/4/8	2/3/8

test data by the trained classifier. We used a personal computer with 3.4GHz CPU and 16GB memory. Table 3 shows the results. From the table the MAMC is the fastest and the MAMC$_{\text{bs}}$ and MAMC$_{\text{b}}$ are comparable to the MAMC. Comparing the MAMC$_{\text{bs}}$ and the MAMC$_{\text{bsg}}$, the MAMC$_{\text{bsg}}$ requires more time because of the grid search. Because the classification performance is comparable, line search seems to be sufficient. The EMAMC, L1 SVM and MAMC$_{\text{bsg}}$ are in the slowest group.

3.3 Discussions

The advantage of the MAMC is its simplicity: The coefficient vector of the decision hyperplane is calculated by addition or subtraction of kernel values. The inferior generalization ability of the original MAMC is mitigated by bias and slope optimization, but the improvement is still not sufficient compared to the EMAMC and L1 SVM. Therefore, the introduction of the equality or inequality constraints are essential. But it leads to the LS SVM or L1 SVM and the simplicity of the MAMC is completely lost.

4 Conclusions

We discussed two ways to improve the generalization ability of the maximum average margin classifier (MAMC). One is to optimize the bias term after calculating the weight vector, and the other is to optimize the slope of the decision function by introducing the weight parameter to the average vector of one class. The parameter value is determined by cross validation. To improve the generalization ability further, we introduced the EMAMC, which is the equality constrained MAMC, but this is shown to be equivalent to the LS SVM defined in the empirical feature space.

Table 3. Training time comparison for the two-class problems (in seconds)

Data	MAMC$_{\rm bs}$	MAMC$_{\rm bsg}$	MAMC$_{\rm b}$	MAMC	EMAMC	L1 SVM
Banana	1.10	9.74	**0.59**	**0.59**	22.40	4.92
B. cancer	0.31	2.77	**0.13**	0.14	2.95	7.08
Diabetes	1.51	14.39	0.78	**0.73**	35.86	22.96
Flare-solar	3.24	29.89	1.61	**1.51**	111.43	218.67
German	4.42	40.03	2.07	**2.04**	148.65	776.53
Heart	0.24	2.13	0.12	**0.11**	2.05	1.75
Image	15.04	144.43	7.62	**7.18**	2290.87	56.7
Ringnorm	1.60	12.91	0.99	**0.96**	23.27	12.57
Splice	13.34	126.51	7.16	**6.87**	887.16	30.71
Thyroid	0.15	1.35	**0.07**	**0.07**	1.16	0.33
Titanic	0.17	1.31	**0.09**	**0.09**	1.32	21.25
Twonorm	1.67	12.92	0.99	**0.92**	22.51	10.46
Waveform	1.51	12.84	**0.88**	**0.88**	22.44	35.61
B/W	0/0	0/1	5/0	12/0	0/7	0/5

According to the experiments for two-class problems, we show that the generalization ability is improved by the bias term and slope optimization. However, the obtained generalization ability is inferior to the EMAMC and L1 SVM. Therefore, the unconstrained MAMC is not so powerful as the EMAMC and L1 SVM.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 25420438.

References

1. V. N. Vapnik. Statistical Learning Theory. John Wiley & Sons, New York, NY, 1998.
2. S. Abe. Support Vector Machines for Pattern Classification. Springer-Verlag, London, UK, second edition, 2010.
3. G. R. G. Lanckriet, L. El Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. Journal of Machine Learning Research, 3:555–582, 2002.
4. K. Huang, H. Yang, I. King, and M. R. Lyu. Learning large margin classifiers locally and globally. In Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004), pages 1–8, 2006.
5. D. S. Yeung, D. Wang, W. W. Y. Ng, E. C. C. Tsang, and X. Wang. Structured large margin machines: sensitive to data distributions. Machine Learning, 68(2):171–200, 2007.
6. H. Xue, S. Chen, and Q. Yang. Structural regularized support vector machine: A framework for structural large margin classifier. IEEE Transactions on Neural Networks, 22(4):573–587, 2011.
7. X. Peng and D. Xu. Twin Mahalanobis distance-based support vector machines for pattern recognition. Information Sciences, 200:22–37, 2012.

8. S. Abe. Training of support vector machines with Mahalanobis kernels. In W. Duch, J. Kacprzyk, E. Oja, and S. Zadrożny, editors, Artificial Neural Networks: Formal Models and Their Applications (ICANN 2005)—Proceedings of Fifteenth International Conference, Part II, Warsaw, Poland, pages 571–576. Springer-Verlag, Berlin, Germany, 2005.
9. D. Wang, D. S. Yeung, and E. C. C. Tsang. Weighted Mahalanobis distance kernels for support vector machines. IEEE Transactions on Neural Networks, 18(5):1453–1462, 2007.
10. C. Shen, J. Kim, and L. Wang. Scalable large-margin Mahalanobis distance metric learning. IEEE Transactions on Neural Networks, 21(9):1524–1530, 2010.
11. X. Liang and Z. Ni. Hyperellipsoidal statistical classifications in a reproducing kernel Hilbert space. IEEE Transactions on Neural Networks, 22(6):968–975, 2011.
12. M. Fauvel, J. Chanussot, J.A. Benediktsson, and A. Villa. Parsimonious Mahalanobis kernel for the classification of high dimensional data. Pattern Recognition, 46(3):845–854, 2013.
13. T. Reitmaier and B. Sick. The responsibility weighted Mahalanobis kernel for semi-supervised training of support vector machines for classification. Information Sciences, 323:179–198, 2015.
14. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119–139, 1997.
15. L. Reyzin and R. E. Schapire. How boosting the margin can also boost classifier complexity. In Proceedings of the 23rd International Conference on Machine learning, pages 753–760. ACM, 2006.
16. W. Gao and Z.-H. Zhou. On the doubt about margin explanation of boosting. Artificial Intelligence, 203:1–18, 2013.
17. A. Garg and D. Roth. Margin distribution and learning. In Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), 2003, Washington, DC, USA, pages 210–217, 2003.
18. K. Pelckmans, J. Suykens, and B. D. Moor. A risk minimization principle for a class of parzen estimators. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1137–1144. Curran Associates, Inc., 2008.
19. F. Aiolli, G. Da San Martino, and A. Sperduti. A kernel method for the optimization of the margin distribution. In V. Kůrková, R. Neruda, and J. Koutnik, editors, Artificial Neural Networks (ICANN 2008)—Proceedings of the Eighteenth International Conference, Prague, Czech Republic, Part I, pages 305–314. Springer-Verlag, Berlin, Germany, 2008.
20. L. Zhang and W.-D. Zhou. Density-induced margin support vector machines. Pattern Recognition, 44(7):1448–1460, 2011.
21. Z.-H. Zhou T. Zhang. Large margin distribution machine. In Twentieth ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 313–322, 2014.
22. Z.-H. Zhou. Large margin distribution learning. In Artificial Neural Networks in Pattern Recognition, pages 1–11. Springer, 2014.
23. G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. Machine Learning, 42(3):287–320, 2001.
24. S. Abe. Fusing sequential minimal optimization and Newton's method for support vector training. International Journal of Machine Learning and Cybernetics (DOI: 10.1007/s13042-014-0265-x), 7:345–364, 2016.