Effect of Equality Constraints to Unconstrained Large Margin Distribution Machines

Shigeo Abe(✉)

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

Abstract. Unconstrained large margin distribution machines (ULDMs) maximize the margin mean and minimize the margin variance without constraints. In this paper, we first reformulate ULDMs as a special case of least squares (LS) LDMs, which are a least squares version of LDMs. By setting a hyperparameter to control the trade-off between the generalization ability and the training error to zero, LS LDMs reduce to ULDMs. In the computer experiments, we include the zero value of the hyperparameter as a candidate value for model selection. According to the experiments using two-class problems, in most cases LS LDMs reduce to ULDMs and their generalization abilities are comparable. Therefore, ULDMs are sufficient to realize high generalization abilities without equality constraints.

1 Introduction

In a classification problem, margins between data and the separating hyperplane play an important role. Here, margin is defined as the distance between a data point and the separating hyperplane and it is nonnegative when correctly classified, and negative, when misclassified. In the support vector machine (SVM) [1,2], the minimum margin is maximized.

Because the SVM does not assume a specific data distribution, the obtained separating hyperplane is optimal under the assumption that the data obey an unknown but fixed distribution. Therefore, if prior knowledge is available, it can improve the generalization ability.

The central idea of SVMs, maximizing the minimum margin, has been applied to improving generalization performance of other classifiers. However, for AdaBoost, instead of the minimum margin, directly controlling the margin distribution has been known to improve the generalization ability [3,4].

Among several classifiers to control the margin distribution [5–12], in [6], the margin mean for the training data is maximized without constraints. This approach is extended in [11]: the bias and slope of the separating hyperplanes are optimized and then equality constraints are introduced. This introduction results in the least squares SVM. According to the computer experiments, without equality constraints, the generalization ability is inferior to that of the SVM.

L. Pancioni et al. (Eds.): ANNPR 2018, LNAI 11081, pp. 41–53, 2018.
https://doi.org/10.1007/978-3-319-99978-4_3

In [9,10], in addition to maximizing the margin mean, the margin variance is minimized and the classifier is called large margin distribution machine (LDM). The advantage of the LDM is that the generalization ability is better than or comparable to that of the SVM, but one of the disadvantages is that two hyperparameters are added to the SVM. This will lengthen model selection. To solve this problem, in [12], an unconstrained LDM (ULDM) is developed, where the number of hyperparameters is the same as that of the SVM.

In this paper, we reformulate the ULDM as a special case of the least squares LDM (LS LDM). As in [12], we formulate the LS LDM as maximizing the margin mean and minimizing the margin variance, in addition to minimizing the square norm of the coefficient vector of the hyperplane and the square sum of slack variables. As in the LS SVM, we impose the equality constraints for training data. Because the hyperparameters are necessary for the square sum of slack variables and the margin variance, one hyperparameter is added to the LS SVM. Eliminating the square sum of slack variables in the objective function and the equality constraints, we obtain the ULDM.

By computer experiments we perform model selection of the LS LDM including the parameter value of zero for the slack variables, which results in the ULDM. Checking the number that the parameter value of zero is taken, we judge whether the equality constraints are necessary for improving the generalization ability.

In Sect. 2, we summarize the LS SVM. And in Sect. 3, we explain the LDM and then discuss its variants: the LS LDM and ULDM. In Sect. 4, we evaluate the effect of equality constraints to the ULDM using two-class problems.

2 Least Squares Support Vector Machines

Let the decision function in the feature space be

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

where $\boldsymbol{\phi}(\mathbf{x})$ maps the m-dimensional input vector $\mathbf{x}$ into the l-dimensional feature space, $\mathbf{w}$ is the l-dimensional coefficient vector, $\top$ denotes the transpose of a vector, and b is the bias term.

Let the M training input-output pairs be $\{\mathbf{x}_i, y_i\}$ $(i = 1, \ldots, M)$, where $\mathbf{x}_i$ are training inputs and y_i are the associated labels and $y_i = 1$ for Class 1 and -1 for Class 2.

The margin of $\mathbf{x}_i$, δ_i, is defined as the distance from the separating hyperplane $f(\mathbf{x}) = 0$, and is given by

$$\delta_i = y_i\, f(\mathbf{x}_i)/\|\mathbf{w}\|. \tag{2}$$

If $\delta\, \|\mathbf{w}\| = 1$, where δ is the minimum margin among δ_i $(i = 1, \ldots, M)$, maximizing δ is equivalent to minimizing $\|\mathbf{w}\|$. To make δ_i larger than or equal to 1, $\mathbf{x}_i$ need to satisfy $y_i\, f(\mathbf{x}_i) \geq 1$. Then allowing misclassification, the LS

SVM is formulated in the primal form as follows:

$$\text{minimize} \quad Q(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} + \frac{C}{2}\sum_{i=1}^{M} \xi_i^2 \tag{3}$$

$$\text{subject to} \quad y_i\, f(\mathbf{x}_i) = 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, M, \tag{4}$$

where $Q(\mathbf{w}, b, \boldsymbol{\xi})$ is the objective function, C is the margin parameter that controls the trade-off between the training error and the generalization ability, ξ_i are the slack variables for $\mathbf{x}_i$, and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)^\top$. If we change ξ_i^2 to ξ_i, and $C/2$ to C in (3), and the equality constraints in (4) to inequality constraints, we obtain the L1 SVM.

Solving the equation in (4) for ξ_i and substituting it to the objective function in (3), we obtain the unconstrained optimization problem:

$$\text{minimize} \quad Q(\mathbf{w}, b) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} + \frac{C}{2}\sum_{i=1}^{M}(1 - y_i\, f(\mathbf{x}_i))^2. \tag{5}$$

The solution of the LS SVM can be obtained by solving a set of linear equations and generalization performance is known to be comparable to the L1 SVM [2], but unlike the L1 SVM, the solution is not sparse.

In the following we use the LS SVM to derive an LS LDM, which is a variant of the LDM, and also use to compare performance of the ULDM.

3 Large Margin Distribution Machines and Their Variants

In this section, first we briefly summarize the LDM. Then, we define the LS LDM and ULDM in a way slightly different from [12].

3.1 Large Margin Distribution Machines

The LDM [9] maximizes the margin mean and minimizes the margin variance.

The margin mean $\bar{\delta}$ and margin variance $\hat{\delta}$ are given, respectively, by

$$\bar{\delta} = \frac{1}{M}\sum_{i=1}^{M} \delta_i, \tag{6}$$

$$\hat{\delta} = \frac{1}{M}\sum_{i=1}^{M} \left(\delta_i - \bar{\delta}\right)^2 = \frac{1}{M}\sum_{i=1}^{M} \delta_i^2 - \bar{\delta}^2. \tag{7}$$

Here, instead of (2), we consider the margin as

$$\delta_i = y_i\, f(\mathbf{x}_i). \tag{8}$$

Similar to the L1 SVM, the LDM is formulated as follows:

$$\text{minimize} \quad Q(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} - \lambda_1 \bar{\delta} + \frac{1}{2}\lambda_2 \hat{\delta} + C \sum_{i=1}^{M} \xi_i \tag{9}$$

$$\text{subject to} \quad y_i\, f(\mathbf{x}_i) \geq 1 - \xi_i \qquad \text{for} \quad i = 1, \ldots, M, \tag{10}$$

where λ_1 and λ_2 are parameters to control maximization of the margin mean and minimization of the margin variance, respectively. In the objective function, the second and the third terms are added to the L1 SVM.

Because the LDM uses all the training data to calculate the margin mean and the margin variance, the solution is dense. Furthermore, because four parameter values (including one kernel parameter value), instead of two, need to be determined by model selection, model selection requires more time than the L1 SVM does.

3.2 Least Squares Large Margin Distribution Machines

The LS LDM that maximizes the margin mean and minimizes the margin variance is given by replacing the slack sum in (9) with the square sum and the inequality constraints in (10) with the equality constraints as follows:

$$\text{minimize} \quad Q(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} - \lambda_1 \bar{\delta} + \frac{1}{2}\lambda_2 \hat{\delta} + \frac{C}{2} \sum_{i=1}^{M} \xi_i^2 \tag{11}$$

$$\text{subject to} \quad y_i\, f(\mathbf{x}_i) = 1 - \xi_i \qquad \text{for} \quad i = 1, \ldots, M. \tag{12}$$

Solving the equation in (12) for ξ_i and substituting it to the objective function in (11) yield

$$\begin{aligned} \text{minimize} \quad Q(\mathbf{w}, b) &= \frac{1}{2}\mathbf{w}^\top \mathbf{w} - \lambda_1 \bar{\delta} + \frac{1}{2}\lambda_2 \hat{\delta} + \frac{C}{2} \sum_{i=1}^{M} (1 - y_i\, f(\mathbf{x}_i))^2 \\ &= \frac{1}{2}\mathbf{w}^\top \mathbf{w} - \lambda_1 \bar{\delta} + \frac{1}{2}\lambda_2 \hat{\delta} + \frac{C}{2} \sum_{i=1}^{M} (\delta_i - 1)^2. \end{aligned} \tag{13}$$

In the above objective function, the last term, which is the variance of margins around the minimum margin works similarly to the third term, which is the variance of margin around the margin mean, $\hat{\delta}$.

Now substituting (6), (7), and (8) into the objective function of (13) and deleting the constant term, we obtain

$$\begin{aligned} Q(\mathbf{w}, b) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} &+ \frac{\lambda_2}{2\,M}\left(1 + \frac{M\,C}{\lambda_2}\right) \sum_{i=1}^{M} f^2(\mathbf{x}_i) - \frac{\lambda_2}{2}\left(\frac{1}{M}\sum_{i=1}^{M} y_i\, f(\mathbf{x}_i)\right)^2 \\ &- \left(\frac{\lambda_1}{M} + C\right) \sum_{i=1}^{M} y_i\, f(\mathbf{x}_i). \end{aligned} \tag{14}$$

The first three terms in the above objective function are quadratic and the last term is linear with respect to $\mathbf{w}$ and b. Therefore, the coefficient of the linear term is a scaling factor of the decision function obtained by minimizing (14) with respect to $\mathbf{w}$ and b. Dividing (14) by λ_2 and eliminating the coefficient of the last term, we obtain

$$\begin{aligned} Q(\mathbf{w}, b) = & \frac{1}{2\,C_{\mathrm{m}}} \mathbf{w}^\top \mathbf{w} + \frac{1 + C_{\mathrm{e}}}{2\,M} \sum_{i=1}^{M} f^2(\mathbf{x}_i) \\ & - \frac{1}{2} \left(\frac{1}{M} \sum_{i=1}^{M} y_i\, f(\mathbf{x}_i) \right)^2 - \sum_{i=1}^{M} y_i\, f(\mathbf{x}_i). \end{aligned} \tag{15}$$

Here, $C_{\mathrm{m}} = \lambda_2$ and $C_{\mathrm{e}} = M\,C/\lambda_2$.

According to the above formulation of the LS LDM, the parameter λ_1 in (13) does not work for controlling the margin mean. Therefore, the three hyperparameters in (11) and (12) are reduced to two.

3.3 Unconstrained Large Margin Distribution Machines

Deleting the square sum of the slack variables in (11) and equality constraints in (12), we consider the unconstrained LDM (ULDM) as follows:

$$\begin{aligned} \text{minimize} \quad Q(\mathbf{w}, b) & = \frac{1}{2\,C_{\mathrm{m}}} \mathbf{w}^\top \mathbf{w} - M\,\bar{\delta} + \frac{1}{2} \hat{\delta} \\ & = \frac{1}{2\,C_{\mathrm{m}}} \mathbf{w}^\top \mathbf{w} + \frac{1}{2\,M} \sum_{i=1}^{M} f^2(\mathbf{x}_i) - \frac{1}{2} \left(\frac{1}{M} \sum_{i=1}^{M} y_i\, f(\mathbf{x}_i) \right)^2 \\ & \quad - \sum_{i=1}^{M} y_i\, f(\mathbf{x}_i). \end{aligned} \tag{16}$$

Here, we multiply $\bar{\delta}$ with M so that the coefficient of the linear term is 1.

Comparing (15) and (16), the ULDM is obtained by setting $C_{\mathrm{e}} = 0\,(C = 0)$.

Because the LS LDM includes the ULDM, we derive the optimality conditions for (15) in the empirical feature space [2]. Let $\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ be a subset of $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, where $N \leq M$ and let $\{\boldsymbol{\phi}(\mathbf{z}_1), \ldots, \boldsymbol{\phi}(\mathbf{z}_N)\}$ span the empirical feature space. Then the mapping function that maps the input space into the empirical feature space is expressed by

$$\mathbf{h}(\mathbf{x}) = (K(\mathbf{x}, \mathbf{z}_1), \ldots, K(\mathbf{x}, \mathbf{z}_N))^\top, \tag{17}$$

where $K(\mathbf{x}, \mathbf{z}_j) = \boldsymbol{\phi}^\top(\mathbf{x})\, \boldsymbol{\phi}(\mathbf{z}_j)$. Then the decision function (1) is expressed by

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{h}(\mathbf{x}) + b. \tag{18}$$

For a linear kernel with $m < N$, to improve sparsity, we use the Euclidean coordinates: $\mathbf{z}_1 = \{1, 0, \ldots, 0\}, \cdots, \mathbf{z}_m = \{0, \cdots, 0, 1\}$, and use the identity mapping: $\mathbf{h}(\mathbf{x}) = \mathbf{x}$.

We derive the optimality condition of the LS LDM given by (15), using (18):

$$\begin{aligned}\frac{\partial Q(\mathbf{w},b)}{\partial \mathbf{w}} &= \left(\frac{1}{C_{\mathrm{m}}} I_N + (1+C_{\mathrm{e}})\overline{K^2} - {\overline{K^y}}^\top \overline{K^y}\right)\mathbf{w} \\ &\quad + \left((1+C_{\mathrm{e}})\bar{K}^\top - \bar{y}\,{\overline{K^y}}^\top\right) b - {\overline{K^y}}^\top = 0, \qquad (19)\\ \frac{\partial Q(\mathbf{w},b)}{\partial b} &= \left((1+C_{\mathrm{e}})\,\bar{K} - \bar{y}\,\overline{K^y}\right)\mathbf{w} + \left(1+C_{\mathrm{e}} - \bar{y}^2\right) b - \bar{y} = 0, \qquad (20)\end{aligned}$$

where I_N is the $N \times N$ unit matrix,

$$\begin{aligned}&\overline{K^2} = \frac{1}{M}\sum_{i=1}^{M} K_i^\top K_i, \qquad \bar{K} = \frac{1}{M}\sum_{i=1}^{M} K_i, \quad \overline{K^y} = \frac{1}{M}\sum_{i=1}^{M} y_i K_i, \quad \bar{y} = \frac{1}{M}\sum_{i=1}^{M} y_i,\\ &K_i = (K_{i1}, \ldots, K_{iN}) = \mathbf{h}^\top(\mathbf{x}_i),\\ &K_{ij} = K(\mathbf{x}_i, \mathbf{z}_j) \quad \text{for } i = 1, \ldots, M,\ j = 1, \ldots, N. \qquad (21)\end{aligned}$$

In a matrix form, (19) and (20) are given by

$$\begin{aligned}&\begin{pmatrix} \frac{1}{C_{\mathrm{m}}} I_N + (1+C_{\mathrm{e}})\,\overline{K^2} - {\overline{K^y}}^\top \overline{K^y} & (1+C_{\mathrm{e}})\,\bar{K}^\top - \bar{y}\,{\overline{K^y}}^\top \\ (1+C_{\mathrm{e}})\,\bar{K} - \bar{y}\,\overline{K^y} & 1 + C_{\mathrm{e}} - \bar{y}^2 \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ b \end{pmatrix}\\ &= \begin{pmatrix} {\overline{K^y}}^\top \\ \bar{y} \end{pmatrix} \qquad (22)\end{aligned}$$

If $C = 0$, (22) reduces to the ULDM. The difference between (22) with $C = 0$ and the ULDM in [12] is that $1/C_{\mathrm{m}}$ is used in (22) instead of C_{m}.

Because the coefficient matrix of (22) is positive definite, we can solve (22) for $\mathbf{w}$ and b by the coordinate descent method [13] as well as by matrix inversion.

In model selection, we need to determine the values of C_{m}, C in C_{e}, and γ in the kernel. To speed up model selection, as well as grid search of three values, we consider line search: after determining the values of C_{m} and γ with $C = 0$ by grid search, we determine the C value fixing the values of C_{m} and γ with the determined values.

4 Performance Evaluation

We compare performance of the ULDM with that of the LS LDM to clarify whether the equality constraints in the LS LDM are necessary. We also compare the ULDM with the LS SVM and the L1 SVM. Because of the space limitation, we only use two-class problems.

4.1 Conditions for Experiment

Because the coefficient matrix of (22) is positive definite, (22) can be solved by the coordinate descent method [9]. But to avoid the imprecise accuracy caused by the improper convergence, we train the ULDM and LS LDM by matrix inversion. We also train the LS SVM given by (3) and (4) by matrix inversion. For the L1 SVM, we use SMO-NM [14], which fuses SMO (Sequential minimal optimization) and NM (Newton's method).

We use the radial basis function (RBF) kernels: $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2/m)$, where m is the number of inputs for normalization and γ is used to control a spread of a radius. We carry out model selection by fivefold cross-validation. To speed up cross-validation for the LS LDM, which has three hyperparameters including γ for the RBF kernel, we use line search in addition to grid search of the optimal values of C, C_{m} and γ. In line search, after determining the values of C_{m} and γ by grid search, we determine the optimal value of C by cross-validation. Therefore C_{m} and γ for the ULDM give the same values for the LS LDM by line search.

We select the γ value from {0.01, 0.1, 0.5, 1, 5, 10, 15, 20, 50, 100, 200}, for the C value from {0.1, 1, 10, 50, 100, 500, 1000, 2000}, and for the C_{m} value from $\{0.1, 1, 10, 100, 1000, 10^4, 10^6, 10^8\}$. In the LS LDM, we also include 0 as a candidate of the C value. Then if 0 is selected, the LS LDM reduces to the ULDM.

We measure the average CPU time per data set including model selection by fivefold cross-validation, training a classifier, and classifying the test data by the trained classifier. We used a personal computer with 3.4 GHz CPU and 16 GB memory.

4.2 Results for Two-Class Problems

Table 1 lists the numbers of inputs, training data, test data, and data set pairs of two-class problems [15]. Each data set pair consists of the training data set and the test data set. Using the training data set, we determine parameter values by cross-validation, train classifiers with the determined parameter values and evaluate the performance using the test data set. Then we calculate the average accuracy and the standard deviation for all the test data sets.

Table 2 lists the parameter values determined by cross-validation. In the first row, (l) and (g) show that the three hyperparameters of the LS LDM are determined by linear search and grid search, respectively. Because each classification problem consists of 100 or 20 training and test data pairs, we show the most frequently selected parameter values. For the LS LDM, most selected value for C is 0. Thus, in the table, we show the number that $C \neq 0$ is selected in the parentheses.

As we discussed before, the C_{m} and γ values for the ULDM and the LS LDM (l) are the same. Therefore, if the number that $C \neq 0$ is selected is 0, the LS LDM (l) reduces to ULDM for all the training data sets. This happens for seven problems. Except for the german problem, the C value of zero is selected frequently. For the LS LDM (g) also, the C value of zero is frequently selected. Therefore, LS LDM (g) reduces to ULDM frequently. These results indicate that the equality constraints are not important in the LS LDM.

Table 1. Benchmark data 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

The γ values for the three classifiers are very similar and so are the C values for the LS and L1 SVMs.

In the following we show the distributions of C, C_{e}, and γ values for the german data, in which $C = 0$ is least frequently selected for the LS LDM.

Table 3 shows the C value distributions for the german data. The distributions for the LS LDM by line search and by grid search are very similar. The values of C smaller than or equal to 1 are selected 93 times and 90 times for the LS LDM (l) and LS LDM (g), respectively. Therefore, C does not affect much to the generalization ability. The distributions for the LS SVM and L1 SVM are similar and although the value of 1 is frequently selected, the larger values are also selected. This means that the value of C affect directly on the generalization ability.

Table 4 shows the distributions of C_{m} values for the ULDM and LS LDM (g). The both distributions are similar. The distribution for the LS LDM (l) is the same as that for the ULDM.

Table 5 lists the γ value distributions for the german data. The γ values larger than 20 are not selected for the four classifiers. The distributions of the ULDM (LS LDM (l)) and LS LDM (g) are similar although smaller values are selected for the ULDM (LS LDM (l)). The distributions of the LS SVM and L1 SVM are similar and tend to gather towards smaller values than those of the ULDM (LS LDM (l)) and LS LDM (g).

Table 6 shows the average accuracies and their standard deviations of the five classifiers with RBF kernels. Among the five classifiers 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

Table 2. Most-frequently-selected parameter values for the two-class problems. The numeral in the parentheses shows the number that $C \neq 0$ is selected.

Data	ULDM	LS LDM (l)	LS LDM (g)	LS SVM	L1 SVM
	C_{m}, γ	$C_{\mathrm{m}}, \gamma\,(C)$	$C_{\mathrm{m}}, \gamma\,(C)$	C, γ	C, γ
Banana	10^4, 50	10^4, 50 (1)	10^4, 100 (1)	10, 50	1, 20
B. cancer	10, 0.01	10, 0.01 (17)	10, 10 (30)	1, 5	1, 0.5
Diabetes	100, 5	100, 5 (10)	100, 5 (22)	1, 0.5	500, 0.1
Flare-solar	10, 0.01	10, 0.01 (0)	10, 1 (0)	10, 0.01	50, 0.01
German	100, 10	100, 10 (31)	100, 10 (38)	1, 0.1	1, 0.1
Heart	100, 0.01	100, 0.01 (0)	10^4, 0.5 (1)	10, 0.01	100, 0.01
Image	10^8, 15	10^8, 15 (1)	10^8, 20 (1)	50, 50	50, 100
Ringnorm	10, 50	10, 50 (0)	10, 100 (0)	0.1, 50	1, 50
Splice	10^4, 10	10^4, 10 (0)	10^6, 10 (0)	10, 10	10, 10
Thyroid	10, 100	10, 100 (0)	10, 200 (6)	1, 100	50, 5
Titanic	10^4, 0.01	10^4, 0.01 (0)	10, 1 (3)	10, 0.01	50, 0.01
Twonorm	1000, 0.01	1000, 0.01 (0)	100, 5 (1)	50, 0.01	1, 0.01
Waveform	100, 50	100, 50 (10)	100, 50 (21)	1, 20	1, 15

Table 3. Distribution of C values for the german data

C	LS LDM (l)	LS LDM (g)	LS SVM	L1 SVM
0.0	69	62	—	—
0.1	11	11	0	0
1	13	17	42	32
10	3	4	11	9
50	2	2	14	20
100	1	2	7	8
500	0	1	9	8
1000	0	0	5	7
2000	1	1	12	16

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 ULDM shows statistically better than, the same as, and worse than the LS LDM (l), LS LDM (g), LS SVM, and L1 SVM, respectively. Symbols "+" and "−" in the L1 SVM column show that the ULDM is statistically better and worse than the L1 SVM, respectively.

Ignore the difference of 0.01 for the average accuracies and the standard deviations. Then the results of the ULDM and those of the LS LDM (l) are

Table 4. Distribution of C_{m} values for the german data

C_{e}	ULDM	LS LDM (g)
0.1	7	5
1	44	50
10	18	21
100	12	13
10^3	6	5
10^4	2	4
10^6	5	2
10^8	6	0

Table 5. Distribution of γ values for the german data

γ value	ULDM	LS LDM (g)	LS SVM	L1 SVM
0.01	11	0	10	12
0.1	2	6	23	24
0.5	9	8	16	16
1	8	9	11	13
5	23	26	22	15
10	27	30	12	8
15	9	10	5	8
20	11	11	1	4
50	0	0	0	0
100	0	0	0	0
200	0	0	0	0

different only for the german problem. Whereas for the ULDM and LS LDM (g), only the ringnorm problem gives the same results.

From the table, from the standpoint of the average accuracy, the ULDM and LS LDM (l) performed best and the LS SVM, the worst. But from the standpoint of statistical analysis the ULDM is statistically comparable with the remaining four classifiers.

Therefore, because the LS LDM frequently reduces to the ULDM and the ULDM is comparable with the LS LDM, the LS LDM can be replaced with the ULDM.

Table 7 shows the average CPU time per data set for calculating the accuracies. The last row shows the numbers that each classifier shows best/worst execution time. In average, the LS SVM is the fastest and the LS LDM (g) the slowest because of the slow model selection by grid search of three hyperparameters. Because the ULDM and LS SVM are trained by solving the sets

Table 6. Accuracy comparison of the two-class problems for RBF kernels

Data	ULDM	LS LDM (l)	LS LDM (g)	LS SVM	L1 SVM
Banana	89.13±0.69	89.13±0.69	89.16±0.59	**89.17**±0.66	**89.17**±0.72
B. cancer	**73.73**±4.34	**73.73**±4.35	**73.73**±4.48	73.13±4.68	73.03±4.51
Diabetes	**76.52**±1.95	**76.52**±1.95	76.32±2.00	76.19±2.00	76.29±1.70
Flare-solar	66.33±2.02	66.33±2.02	66.18±1.94	66.25±1.98	$^{-}$**66.99**±2.12
German	76.14±2.30	76.10±2.30	**76.25**±2.17	76.10±2.10	75.95±2.24
Heart	82.61±3.61	82.61±3.61	82.33±3.77	82.49±3.60	**82.82**±3.37
Image	97.16±0.68	97.17±0.68	97.23±0.53	**97.52**±0.54	97.16±0.41
Ringnorm	98.16±0.35	98.16±0.35	98.17±0.34	**98.19**±0.33	98.14±0.35
Splice	89.13±0.60	89.13±0.60	**89.17**±0.55	88.98±0.70	88.89±0.91
Thyroid	95.28±2.28	95.28±2.28	95.25±2.42	95.08±2.55	**95.35**±2.44
Titanic	77.45±0.89	77.45±0.89	**77.48**±0.87	77.39±0.83	77.39±0.74
Twonorm	**97.43**±0.25	**97.43**±0.25	97.37±0.28	**97.43**±0.27	97.38±0.26
Waveform	90.19±0.52	90.19±0.53	**90.22**±0.51	90.05±0.59	$^{+}$89.76±0.66
W/T/L	—	0/13/0	0/13/0	0/13/0	1/11/1
Average	**85.33** (3/2)	**85.33** (3/1)	85.30 (5/3)	85.23 (4/3)	85.26 (4/7)

Table 7. Execution time comparison of the two-class problems (in seconds)

Data	ULDM	LS LDM(l)	LS LDM(g)	LS SVM	L1 SVM
Banana	28.13	30.67	249.08	12.03	**4.92**
B. cancer	2.91	3.17	25.83	**1.69**	7.08
Diabetes	44.13	48.63	428.30	**20.3**	22.96
Flare-solar	223.96	249.05	2067.59	**67.28**	218.67
German	383.45	431.55	3387.80	**98.72**	776.53
Heart	1.66	1.87	15.04	**1.12**	1.75
Image	4813.18	5419.68	46138.67	1826.86	**56.7**
Ringnorm	26.68	29.42	237.83	13.15	**12.57**
Splice	1919.64	1986.73	15747.32	740.76	**30.71**
Thyroid	0.96	1.06	8.68	0.69	**0.33**
Titanic	1.20	1.33	10.93	**0.75**	21.25
Twonorm	27.81	30.83	271.14	13.33	**10.46**
Waveform	26.64	29.96	246.24	**13.64**	35.61
B/W	0/0	0/0	0/12	7/0	6/1

of linear equations with the equal number of variables, slower training by the ULDM is due to more complex calculation in setting the coefficients of the linear equations. Because the matrix size is the number of training data plus one and because the numbers of training data are smaller than 1000 except for the image and splice data sets, the execution time is relatively short.

The L1 SVM is trained by iterative method. Therefore the training speed depends on the parameter values and for the titanic data, training of the L1 SVM is the slowest. For the ULDM, LS LDM, and LS SVM, the execution time depends on the number of training data not on the parameter values.

5 Conclusions

The unconstrained large margin distribution machine (ULDM) maximizes the margin mean and minimizes the margin variance without constraints.

In this paper, we investigated the effect of the constraints to the ULDM. To do this, we derived the ULDM as a special case of the least squares (LS) LDM, which is the least squares version of the LDM. If the hyperparameter associated with the constraints is set to be zero, the LS LDM reduces to the ULDM. In computer experiments, we carried out model selection of the LS LDM including the zero value of the hyperparameter as a candidate value. For the two-class problems with 100 or 20 data set pairs, in most cases, the LS LDM reduced to the ULDM and if not, there was no statistical difference of generalization abilities. According to the results, the effect of the equality constraints to the generalization ability of the LS LDM is considered to be small and the ULDM can be used instead of the LS LDM.

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). https://doi.org/10.1007/978-1-84996-098-4
3. Reyzin, L., Schapire, R.E.: How boosting the margin can also boost classifier complexity. In: Proceedings of the 23rd International Conference on Machine learning, pp. 753–760. ACM (2006)
4. Gao, W., Zhou, Z.-H.: On the doubt about margin explanation of boosting. Artif. Intell. **203**, 1–18 (2013)
5. Garg, A., Roth, D.: Margin distribution and learning. In: Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003), Washington, DC, USA, pp. 210–217 (2003)
6. Pelckmans, K., Suykens, J., Moor, B.D.: A risk minimization principle for a class of Parzen estimators. In: Platt, J.C., Koller, D., Singer, Y., Roweis, S.T. (eds.) Advances in Neural Information Processing Systems 20, pp. 1137–1144. Curran Associates Inc. (2008)
7. Aiolli, F., Da San Martino, G., Sperduti, A.: A kernel method for the optimization of the margin distribution. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008. LNCS, vol. 5163, pp. 305–314. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87536-9_32
8. Zhang, L., Zhou, W.-D.: Density-induced margin support vector machines. Pattern Recognit. **44**(7), 1448–1460 (2011)

9. Zhang, T., Zhou, Z.-H.: Large margin distribution machine. In: Twentieth ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 313–322 (2014)
10. Zhou, Z.-H.: Large margin distribution learning. In: El Gayar, N., Schwenker, F., Suen, C. (eds.) ANNPR 2014. LNCS (LNAI), vol. 8774, pp. 1–11. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11656-3_1
11. Abe, S.: Improving generalization abilities of maximal average margin classifiers. In: Schwenker, F., Abbas, H.M., El Gayar, N., Trentin, E. (eds.) ANNPR 2016. LNCS (LNAI), vol. 9896, pp. 29–41. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46182-3_3
12. Abe, S.: Unconstrained large margin distribution machines. Pattern Recognit. Lett. **98**, 96–102 (2017)
13. Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S.S., Sundararajan, S.: A dual coordinate descent method for large-scale linear SVM. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, New York, pp. 408–415. ACM (2008)
14. Abe, S.: Fusing sequential minimal optimization and Newton's method for support vector training. Int. J. Mach. Learn. Cybern. **7**(3), 345–364 (2016)
15. Rätsch, G., Onoda, T., Müller, K.-R.: Soft margins for AdaBoost. Mach. Learn. **42**(3), 287–320 (2001)