Kobe University Repository : Kernel

PDF issue: 2025-12-04

Are twin hyperplanes necessary?

Abe, Shigeo

(Citation)
Pattern Recognition Letters,116:218-224

(Issue Date)
2018-12-01

(Resource Type)
journal article

(Version)
Accepted Manuscript

(Rights)
© 2018 Elsevier B.V.
This manuscript version is made available under the CC-BY-NC-ND 4.0 license
http://creativecommons.org/licenses/by-nc-nd/4.0/

(URL)
https://hdl.handle.net/20.500.14094/90006300

Pattern Recognition Letters
journal homepage: www.elsevier.com

Are Twin Hyperplanes Necessary?

Shigeo Abe[a,**]

[a]*Kobe University, Kobe, Japan*

ABSTRACT

In a twin support vector machine (TSVM), the separating hyperplane associated with one class is determined such that the hyperplane is near to the data belonging to the class and that it is away from the other class. A data sample is classified into the class with the nearer hyperplane. In this paper we discuss whether the twin hyperplanes are necessary for the TSVM. By theoretical analysis we first show the equivalence conditions that one of the two decision boundaries of the TSVM coincides with the decision boundary of the SVM. Then for the least squares (LS) version of the TSVM, we clarify the equivalence conditions with the LS SVM or that with two hyperparameters for imbalanced data (one for each class). A comparison of the LS TSVM with the LS SVMs, by computer experiments, shows that the generalization abilities of the LS TSVM are comparable but not superior for 13 two-class problems and an imbalanced two-class problem.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

A support vector machine (SVM) is a binary classifier that separates one class from the other by a single hyperplane [1, 2]. Since the introduction of the SVM, a number of variants have been developed to improve the generalization ability [3, 4, 5, 6].

In [3], a twin support vector machine (TSVM) with two nonparallel hyperplanes, which is an extension of the generalized eigenvalue proximal support vector machine (GEPSVM) [7], is proposed. Each hyperplane is associated with a class and is trained such that the training data of the associated class are enforced to the hyperplane with the square loss of the distance from the hyperplane and that the training data of the opposite class are constrained to be as far as possible. This formulation is the combination of the least squares SVM (LS SVM) [8] and the regular SVM. Because of the least squares formulation, the sparsity (with respect to the number of support vectors) of the TSVM is reduced compared to the regular SVM.

In [3], to avoid the singularity of the coefficient matrix, a small positive value is added to its diagonal elements. This corresponds to a regularization term with a small positive constant. In [9], the regularization term with a hyperparameter is used to control the generalization ability. According to computer experiments, this is shown to improve the generalization ability but with the heavier burden of computation for model selection.

Many variants of TSVMs have been proposed. In [10], the TSVM is extended to the least squares version of the TSVM (LS TSVM), which has a square loss function. In [11], the regular SVM based formulation with the hinge loss is used for the TSVM. This results in improved sparsity of the original TSVM. In addition to the hinge loss and the square loss, other types of loss functions are developed [12, 13].

As for the equivalence of the TSVMs with conventional classifiers, in [12], the LS TSVM with the same hyperparameter values is proved to be equivalent to the LS SVM.

In some papers that discuss the superiority of the proposed methods over previous methods, e.g., [3, 7, 9, 11], the accuracy of the data held-out during cross-validation is used. But, this may lead to biased comparison because the accuracy is improved by tuning the hyperparameters.

In this paper, we discuss the effect of twin separating hyperplanes to generalization ability. The twin separating hyperplanes produce two decision boundaries. Therefore, the TSVM has more separation power than the regular SVM has in the same feature space. In theoretical analysis we discuss the equivalence of one of the two decision boundaries of the TSVM to the decision boundary of the SVM.

Modifying the TSVM in [11], we define the regular SVM-based TSVM, and show that under the restricted conditions

**Corresponding author: Tel.: +81-78-994-3432; fax: +81-78-994-3432;
e-mail: abe@kobe-u.ac.jp (Shigeo Abe)

such that the bias terms are not included and the solutions are unbounded, one of the two decision boundaries of the TSVM reduces to that of the regular SVM.

For the LS TSVM, we define the regularized LS TSVM with four hyperparameters. Then we clarify the conditions that one of the decision boundaries of the LS TSVM is equivalent to the decision boundary of the LS SVM or that with two hyperparameters for imbalanced data. This is an extension of the equivalence proof of the LS TSVM and the LS SVM in [12].

Using several benchmark data sets, we examine whether the LS TSVM generalizes better than the LS SVM or that with two hyperparameters and the second decision boundary is important in improving the generalization ability.

In Section 2, we explain the idea of the TSVM. And in Sections 3 and 4, we discuss the equivalence of the TSVM to the SVM, and the LS TSVM to the LS SVMs, respectively. In Section 5, we compare generalization ability of the LS TSVM and LS SVMs using several benchmark data sets.

2. Idea of Twin Support Vector Machines

In contrast to the regular SVM, the TSVM has two decision functions for class i:

$$f_i(\mathbf{x}) = \mathbf{w}_i^\top \boldsymbol{\phi}(\mathbf{x}) + b_i \quad \text{for } i = 1, 2, \tag{1}$$

where $\mathbf{x}$ is the input vector, $\boldsymbol{\phi}(\mathbf{x})$ is the mapping function that maps $\mathbf{x}$ into the l-dimensional feature space, $\mathbf{w}_i$ is the l-dimensional coefficient vector of the decision function, and b_i is the bias term.

Each decision function is determined separately such that the training data of the associated class are as near as possible to the decision function and the training data of the opposite class are far from the decision function.

In classification, $\mathbf{x}$ is classified into the class with the nearer decision function:

$$\arg\min_{i=1,2} |f_i(\mathbf{x})|. \tag{2}$$

Therefore, the decision boundaries are given by

$$|f_1(\mathbf{x})| = |f_2(\mathbf{x})|. \tag{3}$$

This means that there are two decision boundaries:

$$f_1(\mathbf{x}) = f_2(\mathbf{x}), \quad f_1(\mathbf{x}) = -f_2(\mathbf{x}). \tag{4}$$

Figure 1 shows an example of the decision functions and the decision boundaries for two-class data in the two-dimensional input space. In general, the two decision functions are not parallel. Therefore, unlike regular SVMs, the feature space is divided into four regions and each class has two separate regions. If the two decision functions are in parallel, the four regions reduce to two, but this is rare.

3. Twin Support Vector Machines

In this section, we discuss the architecture of the original TSVM [3] and define the sparse version of the TSVM based on [11]. Finally, we discuss the equivalence of one of the decision boundaries of the TSVM to the decision boundary of the SVM.

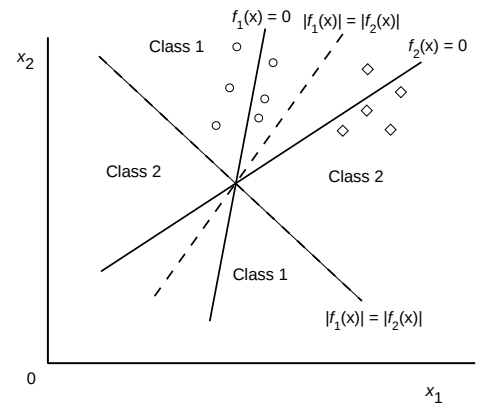

Fig. 1. Decision functions of the TSVM in a two-dimensional space

3.1. Architecture

Assume that we have M training data $\mathbf{x}_1, \ldots, \mathbf{x}_M$ and the first M_1 data belong to Class 1 and the remaining data, to Class 2. Then the twin support vector machine proposed in [3] is defined by

$$\text{minimize} \quad \frac{1}{2}\sum_{i=1}^{M_1} \xi_{1i}^2 + C_1 \sum_{i=M_1+1}^{M} \xi_{1i} \tag{5}$$

$$\text{subject to} \quad \mathbf{w}_1^\top \boldsymbol{\phi}(\mathbf{x}_i) + b_1 + \xi_{1i} = 0 \quad \text{for } i = 1, \ldots M_1, \tag{6}$$

$$-(\mathbf{w}_1^\top \boldsymbol{\phi}(\mathbf{x}_i) + b_1) + \xi_{1i} \geq 1, \quad \xi_{1i} \geq 0 \quad \text{for } i = M_1 + 1, \ldots M, \tag{7}$$

and

$$\text{minimize} \quad \frac{1}{2}\sum_{i=M_1+1}^{M} \xi_{2i}^2 + C_2 \sum_{i=1}^{M_1} \xi_{2i} \tag{8}$$

$$\text{subject to} \quad \mathbf{w}_2^\top \boldsymbol{\phi}(\mathbf{x}_i) + b_2 + \xi_{2i} \geq 1, \quad \xi_{2i} \geq 0 \quad \text{for } i = 1, \ldots M_1, \tag{9}$$

$$\mathbf{w}_2^\top \boldsymbol{\phi}(\mathbf{x}_i) + b_2 + \xi_{2i} = 0 \quad \text{for } i = M_1 + 1, \ldots M, \tag{10}$$

where ξ_{ki} ($k = 1, 2,\ i = 1, \ldots, M$) are slack variables, C_1 and C_2 are control parameters for Classes 1 and 2, respectively.

The first term of the objective function given by (5) is the sum of squares of the slack variables for Class 1 and the second term is the sum of the slack variables for Class 2 multiplied by C_1. Therefore, if $C_1 = 0$, the minimization of the objective function is linear regression for $\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_{M_1})$. If $C_1 \neq 0$, the second term works to minimize the violation of the constraints $f_1(\mathbf{x}_i) \leq -1$ for $i = M_1 + 1, \ldots, M$ as far as possible.

Because of the square loss, sparsity of the solution, which is one of the advantages of SVMs, is lost. In addition, unlike SVMs, the objective function does not include the regularization term $\mathbf{w}_i^\top \mathbf{w}_i$ ($i = 1, 2$). According to the computer experiments in [9], by the introduction of the regularization term, the generalization ability is improved. Therefore, in the following, we define regularized sparse TSVM with the hint of the Twin Parametric-Margin SVM (TPMSVM) given by (8) and (9) in [11].

The regularized sparse TSVM is given by

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}_1^\top \mathbf{w}_1 + C_{11}\sum_{i=1}^{M_1}\xi_{1i} + C_{12}\sum_{i=M_1+1}^{M}\xi_{1i} \tag{11}$$

$$\text{subject to} \quad y_i\,(\mathbf{w}_1^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b_1) + \xi_{1i} \geq 0 \text{ for } i = 1, \ldots M_1, \tag{12}$$

$$y_i\,(\mathbf{w}_1^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b_1) + \xi_{1i} \geq 1 \quad \text{for } i = M_1 + 1, \ldots M \tag{13}$$

and

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}_2^\top \mathbf{w}_2 + C_{21}\sum_{i=1}^{M_1}\xi_{2i} + C_{22}\sum_{i=M+1}^{M}\xi_{2i}, \tag{14}$$

$$\text{subject to} \quad y_i\,(\mathbf{w}_2^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b_2) + \xi_{2i} \geq 1 \text{ for } i = 1, \ldots M_1, \tag{15}$$

$$y_i\,(\mathbf{w}_2^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b_2) + \xi_{2i} \geq 0 \quad \text{for } i = M_1 + 1, \ldots M, \tag{16}$$

where $\xi_{1i} \geq 0$, $\xi_{2i} \geq 0$ ($i = 1, \ldots, M$), $y_i = 1$ for Class 1 and -1 for Class 2 ($i = 1, \ldots, M$), and C_{11}, C_{12}, C_{21}, and C_{22} are hyperparameters to control the trade-off between the accuracy for the training data and the generalization ability.

Each SVM of the TSVM given by (11) to (16) is very similar to the regular SVM: If we set $C_{11} = C_{12}$ in (11) and replace 0 in the right hand-side of (12) with 1, we obtain the regular SVM. This model is used to show the conditions that the TSVM reduces to SVM.

We solve (11) to (13) in the dual form. Introducing the Lagrange multipliers α_{1i} and β_{1i}, we obtain the unconstrained minimization problem:

$$\begin{aligned}\text{minimize} \quad & \frac{1}{2}\mathbf{w}_1^\top \mathbf{w}_1 + C_{11}\sum_{i=1}^{M_1}\xi_{1i} + C_{12}\sum_{i=M_1+1}^{M}\xi_{1i} \\ & - \sum_{i=1}^{M_1}\alpha_{1i}\left(y_i\,(\mathbf{w}_1^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b_1) + \xi_{1i}\right) \\ & - \sum_{i=M_1+1}^{M}\alpha_{1i}\left(y_i\,(\mathbf{w}_1^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b_1) + \xi_{1i} - 1\right) \\ & - \sum_{i=1}^{M}\beta_i\,\xi_{1i},\end{aligned} \tag{17}$$

where $\alpha_{1i} \geq 0$ and $\beta_{1i} \geq 0$ for $i = 1, \ldots, M$.

The optimality conditions for (17) are given by

$$\mathbf{w}_1 = \sum_{i=1}^{M}\alpha_{1i}\,y_i\,\boldsymbol{\phi}(\mathbf{x}_i), \tag{18}$$

$$\sum_{i=1}^{M} y_i\,\alpha_{1i} = 0, \tag{19}$$

$$C_{11} - \alpha_{1i} - \beta_{1i} = 0 \quad \text{for } i = 1, \ldots, M_1, \tag{20}$$

$$C_{12} - \alpha_{1i} - \beta_{1i} = 0 \quad \text{for } i = M_1 + 1, \ldots, M, \tag{21}$$

$$\alpha_{1i}\left(y_i(\mathbf{w}_1^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b_1) + \xi_{1i}\right) = 0 \text{ for } i = 1, \ldots, M_1, \tag{22}$$

$$\alpha_{1i}\left(y_i(\mathbf{w}_1^\top\,\boldsymbol{\phi}(\mathbf{x}_i) + b_1) + \xi_{1i} - 1\right) = 0 \quad \text{for } i = M_1 + 1, \ldots, M. \tag{23}$$

Substituting (18) to (21) into (17), we obtain the following optimization problem

$$\text{maximize} \quad \sum_{i=M_1+1}^{M}\alpha_{1i} - \frac{1}{2}\sum_{i,j=1}^{M}\alpha_{1i}\,\alpha_{1j}\,y_i\,y_j\,K(\mathbf{x}_i, \mathbf{x}_j) \tag{24}$$

$$\text{subject to} \quad \sum_{i=1}^{M} y_i\,\alpha_{1i} = 0, \tag{25}$$

$$0 \leq \alpha_{1i} \leq C_{11} \quad \text{for } i = 1, \ldots M_1, \tag{26}$$

$$0 \leq \alpha_{1i} \leq C_{12} \quad \text{for } i = M_1 + 1, \ldots M, \tag{27}$$

where $K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}^\top(\mathbf{x}_i)\,\boldsymbol{\phi}(\mathbf{x}_j)$ and $K(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function. In our study we use RBF kernels: $\exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$ where $\gamma\,(> 0)$ is a positive parameter to control the spread.

Likewise, the optimization problem given by (14) to (16) are converted into

$$\text{maximize} \quad \sum_{i=1}^{M_1}\alpha_{2i} - \frac{1}{2}\sum_{i,j=1}^{M}\alpha_{2i}\,\alpha_{2j}\,y_i\,y_j\,K(\mathbf{x}_i, \mathbf{x}_j) \tag{28}$$

$$\text{subject to} \quad \sum_{i=1}^{M} y_i\,\alpha_{2i} = 0, \tag{29}$$

$$0 \leq \alpha_{2i} \leq C_{21} \quad \text{for } i = 1, \ldots M_1, \tag{30}$$

$$0 \leq \alpha_{2i} \leq C_{22} \quad \text{for } i = M_1 + 1, \ldots M. \tag{31}$$

3.2. Relations between Twin Support Vector Machines and Regular Support Vector Machines

Now investigate the relation between the TSVM and the regular SVM.

To simplify the discussion, we make the following assumptions:

1. The decision functions do not include the bias terms. Then, we can delete the equality constraints (25) and (29).
2. Let the set of support vectors for (24) to (27), that for (28) to (31), and that for the regular SVM be S_1, S_2, and S. And let $S_1 = S_2 = S$.
3. All the support vectors are unbounded, namely, $0 \leq \alpha_{ki} < C_{ki}$ for $k = 1, 2$, $i = 1, \ldots, M$, where $C_{ki} = C_{k1}$ for $i = 1, \ldots, M_1$ and $C_{ki} = C_{k2}$ for $i = M_1 + 1, \ldots, M$.

Exclusion of the bias terms is not a serious problem because the constant terms in the kernels work to generate bias terms implicitly. The widely-used RBF kernels include constant terms. And if the constant terms are not included, we need only to add 1 to the kernels [2].

If support vectors are all unbounded, the non-zero elements of the solution for (24) to (27) are obtained by solving

$$K_1\,\boldsymbol{\alpha}_{S_1} = \mathbf{g}_1, \tag{32}$$

where $\boldsymbol{\alpha}_{S_1}$ is the $|S_1|$-dimensional vector whose elements are α_{1i} for $i \in S_1$, $\mathbf{g}_1$ is the $|S_1|$-dimensional vector whose elements are 0 for $i \in \{1, \ldots, M_1\}$ and 1 for $i \in \{M_1 + 1, \ldots, M\}$, and

$$K_1 = \left\{y_i\,y_j\,K(\mathbf{x}_i, \mathbf{x}_j)\right\} \quad \text{for } i, j \in S_1. \tag{33}$$

Likewise, the non-zero elements of the solution for (28) to (31) are given by solving

$$K_2\,\boldsymbol{\alpha}_{S_2} = \mathbf{g}_2, \tag{34}$$

where $K_2 = K_1$, $\boldsymbol{\alpha}_{S_2}$ is the $|S_2|$-dimensional vector whose elements are α_{2i} for $i \in S_2$, $\mathbf{g}_2$ is the $|S_2|$-dimensional vector whose elements are 1 for $i \in \{1, \ldots, M_1\}$ and 0 for $i \in \{M_1+1, \ldots, M\}$. Therefore,

$$\mathbf{g}_1 + \mathbf{g}_2 = \mathbf{1}_{S_1}, \tag{35}$$

where $\mathbf{1}_{S_1}$ is the $|S_1|$-dimensional vector whose elements are all 1.

The non-zero elements of the solution for the regular SVM are obtained by solving

$$K\,\boldsymbol{\alpha}_S = \mathbf{1}_{S_1}, \tag{36}$$

where $K = K_1 = K_2$.

Assume that K is nonsingular. Then, from (32) to (36),

$$\boldsymbol{\alpha}_S = \boldsymbol{\alpha}_{S_1} + \boldsymbol{\alpha}_{S_2}. \tag{37}$$

Therefore one of the decision boundaries for the TSVM, $f_1(\mathbf{x}) = f_2(\mathbf{x})$, is equivalent to the decision boundary for the regular SVM.

The above assumptions for the TSVM are too restrictive to realize the equivalent decision boundary by the regular SVM in real applications. For the LS TSVM, however, we can prove the equivalence of the LS TSVM with the LS SVM without any assumptions as the following section shows.

4. Least Squares Twin Support Vector Machines

In this section, we define the regularized version of the LS TSVM [10]. Then, we clarify the conditions where one of the decision boundaries of the LS TSVM coincides with the decision boundary of the LS SVM.

4.1. Architecture

The regularized version of the LS TSVM proposed in [10] is given by

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}_1^\top \mathbf{w}_1 + \frac{C_{11}}{2}\sum_{i=1}^{M_1} \xi_{1i}^2 + \frac{C_{12}}{2}\sum_{i=M_1+1}^{M} \xi_{1i}^2 \tag{38}$$

$$\text{subject to} \quad \mathbf{w}_1^\top \boldsymbol{\phi}(\mathbf{x}_i) + b_1 + \xi_{1i} = y_{1i} \quad \text{for } i = 1, \ldots M, \tag{39}$$

where ξ_{1i} are slack variables, and $y_{1i} = 0$ for Class 1 and -1 for Class 2, and

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}_2^\top \mathbf{w}_2 + \frac{C_{21}}{2}\sum_{i=1}^{M_1} \xi_{2i}^2 + \frac{C_{22}}{2}\sum_{i=M_1+1}^{M} \xi_{2i}^2, \tag{40}$$

$$\text{subject to} \quad \mathbf{w}_2^\top \boldsymbol{\phi}(\mathbf{x}_i) + b_2 + \xi_{2i} = y_{2i} \quad \text{for } i = 1, \ldots M, \tag{41}$$

where ξ_{2i} are slack variables, and $y_{2i} = 1$ for Class 1 and 0 for Class 2.

In (38) and (39), if $y_{1i} = 1$ for Class 1 and -1 for Class 2, we obtain the LS SVM with a separate hyperparameter for both classes. If $C_{11} = C_{12}$, we obtain the regular LS SVM. The LS SVM with two hyperparameters are used for unbalanced data to improve the generalization ability of the class with a smaller number of data.

To solve (38) and (39) in the dual form, we derive the unconstrained optimization problem:

$$\begin{aligned}\text{minimize} \quad & \frac{1}{2}\mathbf{w}_1^\top \mathbf{w}_1 + \frac{C_{11}}{2}\sum_{i=1}^{M_1} \xi_{1i}^2 + \frac{C_{12}}{2}\sum_{i=M_1+1}^{M} \xi_{1i}^2 \\ & - \sum_{i=1}^{M} \alpha_{1i}(\mathbf{w}_1^\top \boldsymbol{\phi}(\mathbf{x}_i) + b_1 + \xi_{1i} - y_{1i}),\end{aligned} \tag{42}$$

where α_{1i} are Lagrange multipliers. Taking the partial derivative of (42) with respect to $\mathbf{w}_1$, b_1, and ξ_{1i} together with the equality constraints, we obtain the following optimality conditions:

$$\mathbf{w}_1 = \sum_{i=1}^{M} \alpha_{1i}\,\boldsymbol{\phi}(\mathbf{x}_i), \tag{43}$$

$$\sum_{i=1}^{M} \alpha_{1i} = 0, \tag{44}$$

$$\alpha_{1i} = C_{11}\,\xi_{1i} \quad \text{for } i = 1, \ldots, M_1, \tag{45}$$

$$\alpha_{1i} = C_{12}\,\xi_{1i} \quad \text{for } i = M_1+1, \ldots, M, \tag{46}$$

$$\mathbf{w}_1^\top \boldsymbol{\phi}(\mathbf{x}_i) + b_1 + \xi_{1i} = y_{1i} \quad \text{for } i = 1, \ldots M. \tag{47}$$

Substituting (43) into (47), we obtain

$$\begin{aligned}& \sum_{i=1}^{M} \alpha_{1i}\,\boldsymbol{\phi}^\top(\mathbf{x}_i)\,\boldsymbol{\phi}(\mathbf{x}_j) + b_1 + \xi_{1j} \\ = & \sum_{i=1}^{M} \alpha_{1i}\,K_{ij} + b_1 + \xi_{1j} = y_{1j} \quad \text{for } j = 1, \ldots M,\end{aligned} \tag{48}$$

where $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}^\top(\mathbf{x}_i)\,\boldsymbol{\phi}(\mathbf{x}_j)$. We substitute (45) and (46) into (48) and obtain

$$\sum_{i=1}^{M} \alpha_{1i}\,K_{ij} + b_1 + \alpha_{1j}/C_{1j} = y_{1j} \quad \text{for } j = 1, \ldots M, \tag{49}$$

where $C_{1j} = C_{11}$ for $j = 1, \ldots, M_1$ and $C_{1j} = C_{12}$ for $j = \ldots, M_1+1, \ldots, M$.

Expressing (44) and (49) in a matrix form,

$$\Omega(C_{11}, C_{12})\begin{pmatrix}\boldsymbol{\alpha}_1 \\ b_1\end{pmatrix} = \begin{pmatrix}\mathbf{y}_1 \\ 0\end{pmatrix}, \tag{50}$$

where $\boldsymbol{\alpha}_1 = (\alpha_{11}, \ldots, \alpha_{1M})^\top$, $\mathbf{y}_1 = (y_{11}, \ldots, y_{1M})^\top$,

$$\Omega(C_{11}, C_{12}) = \begin{pmatrix} D(C_{11}, C_{12}) + K & \mathbf{1}_M \\ \mathbf{1}_M^\top & 0 \end{pmatrix}, \tag{51}$$

$$D(C_{11}, C_{12}) = \mathrm{diag}(1/C_{11}, \ldots, 1/C_{11}, 1/C_{12}, \ldots, 1/C_{12}), \tag{52}$$

$K = \{K_{ij}\}$, and $\mathbf{1}_M$ is the M-dimensional column vector with all elements being 1.

Similarly the solution of (40) and (41) is expressed by

$$\Omega(C_{21}, C_{22})\begin{pmatrix}\boldsymbol{\alpha}_2 \\ b_2\end{pmatrix} = \begin{pmatrix}\mathbf{y}_2 \\ 0\end{pmatrix}, \tag{53}$$

where $\boldsymbol{\alpha}_2 = (\alpha_{21}, \ldots, \alpha_{2M})^\top$ and $\mathbf{y}_2 = (y_{21}, \ldots, y_{2M})^\top$.

4.2. Analysis of Least Squares Twin Support Vector Machines

In this section, we clarify the relationship between LS TSVMs and LS SVMs and consider the limitation of the number of hyperparameters from model selection.

According to the definition of the LS TSVM given by (38) to (41), one of the decision boundaries of the LS TSVM given by (4), $f_1(\mathbf{x}) + f_2(\mathbf{x}) = 0$, relates to the boundary given by the LS SVM. The following theorem shows the equivalence of the decision boundaries of the LS TSVM and LS SVM.

Theorem 1. *The decision boundary of $f_1(\mathbf{x}) + f_2(\mathbf{x}) = 0$ for the LS TSVM given by (38) to (41) with $C_{11} = C_{21}$ and $C_{12} = C_{22}$ is the same as that of the LS SVM with the same hyperparameters, C_{11}, C_{12}.*

Proof. From (50) and (53), and from the assumption of $C_{11} = C_{21}$ and $C_{12} = C_{22}$,

$$\Omega(C_{11}, C_{12})\begin{pmatrix}\boldsymbol{\alpha}\\ b\end{pmatrix} = \begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix}, \tag{54}$$

where $\boldsymbol{\alpha} = \boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2$, $b = b_1 + b_2$, $\mathbf{y} = (y_1, \ldots, y_M)^\top$ and $y_i = y_{1i} + y_{2i}$. Therefore, $y_{1i} = 1$ for Class 1 and -1 for Class 2. This means that $(\boldsymbol{\alpha}, b)$ is the solution for the LS SVM with the hyperparameters of C_{11} and C_{12}.

Therefore,

$$\begin{pmatrix}\mathbf{w}\\ b\end{pmatrix} = \begin{pmatrix}\mathbf{w}_1\\ b_1\end{pmatrix} + \begin{pmatrix}\mathbf{w}_2\\ b_2\end{pmatrix}. \tag{55}$$

Thus for the decision boundaries of the LS SVM and the LS TSVM, the following relation holds:

$$\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) + b = f_1(\mathbf{x}) + f_2(\mathbf{x}) = 0. \tag{56}$$

This means that the decision boundary of the LS SVM coincides with one of the decision boundaries of the LS TSVM. ■

It is easy to show that twin hyperplanes of the LS TSVM are nonparallel even when $C_{11} = C_{21}$ and $C_{12} = C_{22}$. Therefore, as shown in Fig. 2, we must notice that in addition to the decision boundary with $-f_1(\mathbf{x}) = f_2(\mathbf{x})$, which is the same as that of the LS SVM, the decision boundary with $f_1(\mathbf{x}) = f_2(\mathbf{x})$ exists; and that the both classifiers give the same classification results only for the upper half plane of $f_1(\mathbf{x}) = f_2(\mathbf{x})$.

By the LS TSVM, the linearly inseparable case as shown in Fig. 3 can be separated using linear kernels. This is, however, not possible by the LS SVM without using nonlinear kernels.

The LS TSVM given by (38) to (41) includes four hyperparameters and if RBF kernels are used, five hyperparameter values need to be determined. The generalization ability of the LS TSVM depends heavily on the parameter values selected and usually reliable but inefficient cross-validation is used. Because the regular LS SVM uses two hyperparameters including a kernel parameter, cross-validation for five parameters is prohibitive. Therefore, we restrict the number of hyperparameters to two excluding the kernel parameter.

Let C_1 and C_2 be two hyperparameters assigned to C_{11}, C_{12}, C_{21}, and C_{22} and C_1 be selected even times. Then the combinations of the assignment are shown in Table 1. According to Theorem 1, the LS TSVM with Case 1 is equivalent to the regular LS SVM, and the LS TSVM with Case 3 is equivalent to the LS SVM with two hyperparameters. For Cases 2 and 4, there is no equivalent LS SVM. In [11], Case 4 is used.

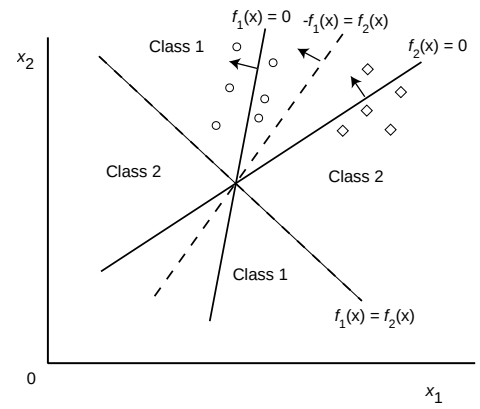

Fig. 2. Decision boundaries of the LS TSVM and the LS SVM in a two-dimensional space. The decision boundaries of the LS TSVM are $f_1(\mathbf{x}) = f_2(\mathbf{x})$ and $f_1(\mathbf{x}) = -f_2(\mathbf{x})$, while the decision boundary of the LS SVM is $f_1(\mathbf{x}) = -f_2(\mathbf{x})$.

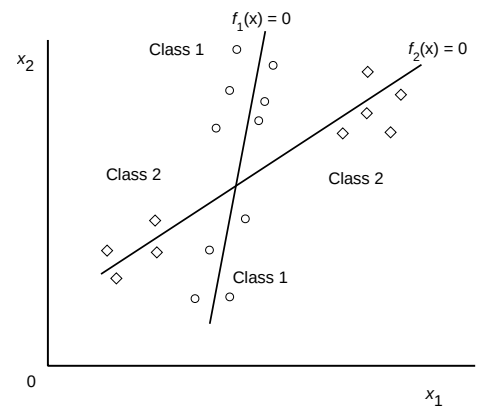

Fig. 3. An example of inseparable case in a two-dimensional space

5. Performance Evaluation

We compared the accuracies of the LS TSVMs for the four cases shown in Table 1 using UCI two-class data sets [14]. For Cases 1 and 3, we used the LS SVM, because one of the decision boundaries of the LS TSVM is equivalent to the decision boundary of the LS SVM.

Because the imbalanced data may widen the difference of the generalization abilities, we also analyze the effect of the LS TSVM and the LS SVM to imbalanced data.

5.1. Model Selection

To compare classifier performance we need to determine the optimum parameter values. This process is called model se-

Table 1. Equivalence of LS TSVMs with LS SVMs

Case	C_{11}	C_{12}	C_{21}	C_{22}	Equivalence
1	C_1	C_1	C_1	C_1	LS SVM
2	C_1	C_1	C_2	C_2	—
3	C_1	C_2	C_1	C_2	LS SVM
4	C_1	C_2	C_2	C_1	—

Table 2. Benchmark data for two-class problems

Problem	Inputs	Train	Test	Sets	Ratio
Banana	2	400	4,900	100	1.20 (1.23)
Breast cancer	9	200	77	100	2.40 (2.47)
Diabetes	8	468	300	100	1.86 (1.88)
Flare-solar	9	666	400	100	0.81 (0.81)
German	20	700	300	100	2.32 (2.35)
Heart	13	170	100	100	1.25 (1.25)
Image	18	1,300	1,010	20	0.74 (0.76)
Ringnorm	20	400	7,000	100	1.01 (1.02)
Splice	60	1,000	2,175	20	1.07 (1.08)
Thyroid	5	140	75	100	2.28 (2.36)
Titanic	3	150	2,051	100	2.11 (2.10)
Twonorm	20	400	7,000	100	0.98 (1.00)
Waveform	21	400	4,600	100	2.02 (2.04)

Table 3. The distribution of p-values

Problem	$p < 0.05$	$p_{\min}$	$p_{\max}$
Banana	—	0.0707	0.7730
Breast cancer	—	0.0958	0.5557
Diabetes	—	0.7068	0.9716
Flare-solar	—	0.2509	1.0000
German	—	0.5546	0.8753
Heart	—	0.2329	0.4650
Image	—	0.2329	0.7804
Ringnorm	—	0.0973	1.0000
Splice	—	0.6524	0.8916
Thyroid	—	0.4447	0.8264
Titanic	—	0.6185	0.9399
Twonorm	0.0498	0.0994	1.0000
Waveform	—	0.2480	0.8212

lection. In our case we need to determine the parameter values of C_1, C_2, and γ for RBF kernels. We used fivefold cross-validation for model selection because it gives a reliable result although time consuming.

In cross-validation, we randomly split a training data set into five approximately-equal-size subsets and train the classifier using four subsets and evaluate the classifier using the remaining subset. We repeat the procedure five times with different combinations and evaluate the total accuracy of the subsets that are held out during training using the four subsets. We call the accuracy thus evaluated accuracy by cross-validation. We evaluate the accuracies by cross-validation for the parameter values assigned for each hyperparameter and select the hyperparameter values that realize the best accuracy by cross-validation.

In cross-validation, we selected the C_1 and C_2 values from {0.1, 1, 10, 50, 100, 500, 1000, 2000} and the γ values from {0.01, 0.1, 0.5, 1, 5, 10, 15, 20, 50, 100, 200}.

For Cases 2 to 4, we need to determine one more parameter value than for Case 1. To speed up cross-validation, in addition to grid search we also tried the combination of grid search and line search; First we determine the values of C_1 and γ. Then we determine the value of C_2 fixing the values of C_1 and γ.

Setting the parameter values determined by cross-validation, we trained the classifier using the training data and evaluated the accuracy of the associated test data. We evaluated the average accuracies and the standard deviations and compared the pairwise statistical significance using Welch's t test.

To avoid insufficient convergence by iterative methods, and thus to avoid the imprecise comparison of accuracies, we train the LS TSVM and LS SVM by matrix inversion. This will lead to slow training especially for large size problems. But we use matrix inversion to keep the comparison of classifiers accurate.

5.2. Two-class problems

Table 2 lists the numbers of inputs, training data, test data, data set pairs, and the ratio of imbalance for the two-class problems. Each data set pair consists of the training data set and the test data set. The ratio of imbalance shows the ratio of the number of data for Class 2 to that for Class 1. The numeral in the parentheses shows the ratio for the test data. The ratios of imbalance range in 0.7 to 3 for both training data and test data and thus all the problems are relatively balanced.

We performed Welch's t test with the confidence intervals of 95% ($p < 0.05$) for the average accuracies of each problem obtained by the LS TSVM^4 and each of the remaining classifiers, where superscript 4 denotes Case 4 in Table 1. Table 3 shows the distribution of the p-values. In the table, column "$p < 0.05$" shows the p-value smaller than 0.05, if it exists, and $p_{\min}$ and $p_{\max}$ show the minimum and maximum p-values for $p > 0.05$, respectively. If $p < 0.05$, the average accuracies of LS TSVM^4 and each of the remaining classifiers are statistically different, and otherwise, statistically the same. The statistical difference only appear for the twonorm problem and also the associated p-value is very close to 0.05.

Table 4 shows the average accuracies and their standard deviations of the classifiers. The superscript for each classifier name is the case number shown in Table 1 and (l) denotes that the cross-validation was performed by combining the grid search and line search.

Among seven classifiers the best average accuracy is shown in bold and the worst one 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. The "W/T/L" row shows the results of Welch's t test; W, T, and L denote the numbers of times that the LS TSVM^4 shows statistically better than, the same as, and worse than another classifier, respectively. The superscript "−" for the average accuracy means that the LS TSVM^4 is statistically worse than the associated classifier. For example, W/T/L for the LS TSVM^4 (l) is 0/12/1: W = 0 because the LS TSVM^4 is not statistically better than the LS TSVM^4 (l) for any problem; T = 12, because statistically comparable for 12 problems; and L = 1 because statistically inferior for the twonorm problem.

From the average accuracy, the LS TSVM^2 and LS TSVM^2 (l) performed best and the LS TSVM^4, worst. (For the validation data sets, the LS TSVM^4 was the best. But because of the space limitation, we cannot include the results.) Statistically, the accuracy of LS TSVM^4 is comparable to LS TSVM^2, LS TSVM^2 (l), and LS SVM^3, but slightly inferior to the LS TSVM^4(l), LS SVM^3 (l), and LS SVM^1. Therefore, seven classifiers are statistically comparable.

The above results indicate that the second decision boundary, $f_1(\mathbf{x}) = f_2(\mathbf{x})$, did not work to improve the generalization

Table 4. Accuracy comparison of the test data for the two-class problems

Problem	LS TSVM^4	LS TSVM^4 (l)	LS TSVM^2	LS TSVM^2 (l)	LS SVM^3	LS SVM^3 (l)	LS SVM^1
Banana	89.06±0.81	**89.24**±0.57	89.21±0.64	89.17±0.65	89.09±0.65	89.17±0.67	89.17±0.66
B. cancer	72.75±4.42	73.25±4.63	73.61±4.72	73.39±4.94	**73.82**±4.62	73.27±4.81	73.13±4.68
Diabetes	76.20±1.96	76.13±1.99	**76.30**±1.79	76.25±2.00	76.13±2.07	76.19±2.00	76.19±2.00
Flare-solar	65.94±1.94	66.23±1.97	66.22±1.98	**66.26**±1.99	65.94±1.98	66.25±1.98	66.25±1.98
German	75.92±2.20	76.09±2.11	76.00±2.27	76.07±2.08	75.97±2.30	**76.10**±2.10	**76.10**±2.10
Heart	82.05±3.74	82.47±3.62	**82.74**±3.48	82.52±3.52	82.64±3.50	82.43±3.60	82.49±3.60
Image	**97.64**±0.47	97.45±0.52	97.60±0.43	97.62±0.48	97.44±0.58	97.49±0.51	97.52±0.54
Ringnorm	**98.21**±0.26	**98.21**±0.33	98.17±0.36	98.18±0.33	98.14±0.33	98.19±0.33	98.19±0.33
Splice	**89.03**±0.65	88.95±0.72	88.99±0.65	88.94±0.71	89.00±0.73	88.93±0.74	88.98±0.70
Thyroid	94.84±2.62	95.09±2.54	94.92±2.53	**95.12**±2.55	94.95±2.35	95.08±2.55	95.08±2.55
Titanic	**77.45**±0.88	77.40±0.82	77.43±0.86	77.40±0.82	77.44±0.99	77.39±0.82	77.39±0.83
Twonorm	97.36±0.23	**97.43**$^-$±0.27	97.35±0.31	97.42±0.28	97.36±0.33	**97.43**$^-$±0.27	**97.43**$^-$±0.27
Waveform	89.96±0.63	89.98±0.62	89.90±0.57	**90.06**±0.59	89.88±0.68	90.05±0.59	90.05±0.59
Average (B/W)	85.11 (4/6)	85.23 (3/1)	**85.26** (2/1)	**85.26** (3/0)	85.22 (1/5)	85.23 (2/2)	85.23 (2/1)
W/T/L	—	0/12/1	0/13/0	0/13/0	0/13/0	0/12/1	0/12/1

ability. To check this, we carried out the simulation for the LS SVM^1 using the LS TSVM^4 with $C_1 = C_2$. For all the 13 problems, both LS SVM^1 and LS TSVM^4 with $C_1 = C_2$ gave the same classification accuracies for the validation and test data. Therefore, the decision boundary, $f_1(\mathbf{x}) = f_2(\mathbf{x})$, was not used at all.

5.3. *Effect of LS TSVMs and LS SVMs to Imbalanced Data*

To clarify the effect of the LS TSVMs and LS SVMs to the imbalanced data, we used two-class data from the three-class thyroid data (93 data for Class 1 and 3488 data for Class 3) [15]. We changed the ratio of imbalance from 1 (93 data for Class 3) to 37.5 (3488 data for Class 3) randomly selecting Class 3 data, and examined the total accuracies of the test data (73 data for Class 1 and 3178 for Class 3) and the accuracies of the Class 1 data (the number of correctly classified data divided by the number of Class 1 data).

Table 5 shows the total accuracy of the test data. The numerals in the parenthesis show the accuracy of Class 1 data. And B and W in "B/W" denote the numbers of times that the classifier associated with the column shows the best and worst accuracies, respectively. From the table the total accuracy of LS TSVM^4 is the best and that of LS SVM^3, the worst. But the accuracy of Class 1 data for the LS SVM^3 is the best and that for the LS SVM^2 (l), the worst. So, at least a separate hyperparameter for each class works to improve the accuracy of the LS SVM^3 for the class with the smaller number of data.

6. Conclusions

We investigated theoretically and experimentally whether twin support vector machines (TSVMs), which employ twin hyperplanes, improve generalization abilities compared to regular support vector machines (SVMs). We focused on one of the two decision boundaries, which works similar to the decision boundary of the SVM. Under restricted conditions such as all the support vectors are unbounded, we proved that one of the decision boundaries of the TSVM is equivalent to the decision boundary of the SVM. For the LS (least squares) TSVM, we clarify the conditions that one of the decision boundaries of the LS TSVM is equivalent to the decision boundary of the LS SVM.

By computer experiments we compared accuracies of the LS TSVMs and LS SVMs for the test data. For the UCI two-class problems, pairwise statistical analysis of the accuracy of one type of LS TSVM with those of the other classifiers for the test data shows that the LS TSVM is comparable or inferior to other classifiers. By experimenting the LS SVM by the LS TSVM with the same hyperparameters, the accuracies of both classifiers were the same for the test data. Therefore, the second decision boundary of the LS TSVM did not work to improve the generalization ability.

For the imbalanced data, the LS SVM with two hyperparameters showed better accuracy for the class with the smaller number of data.

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] Jayadeva, R. Khemchandani, and S. Chandra. Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):905–910, May 2007.

[4] V. Vapnik and R. Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16:2023–2049, 2015.

[5] X. Peng and D. Xu. Twin Mahalanobis distance-based support vector machines for pattern recognition. *Information Sciences*, 200:22–37, 2012.

[6] 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.

[7] O. L. Mangasarian and E. W. Wild. Multisurface proximal support vector classification via generalized eigenvalues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):69–74, Jan. 2006.

[8] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific Publishing, Singapore, 2002.

[9] Y. H. Shao, C. H. Zhang, X. B. Wang, and N. Y. Deng. Improvements on twin support vector machines. *IEEE Transactions on Neural Networks*, 22(6):962–968, June 2011.

[10] M. A. Kumar and M. Gopal. Least squares twin support vector machines for pattern classification. *Expert Systems with Applications*, 36(4):7535–7543, 2009.

Table 5. Accuracies of the test data for the two-class thyroid problem

Ratio	LS TSVM^4	LS TSVM^4 (l)	LS TSVM^2	LS TSVM^2 (l)	LS SVM^3	LS SVM^3 (l)	LS SVM^1
1	**96.89** (94.52)	96.16 (**97.26**)	96.25 (95.89)	96.16 (**97.26**)	96.16 (**97.26**)	96.16 (**97.26**)	96.16 (**97.26**)
5	98.28 (83.56)	**98.65** (82.19)	98.15 (73.97)	**98.65** (82.19)	98.34 (**91.78**)	98.19 (86.30)	**98.65** (82.19)
10	98.92 (75.34)	**98.95** (73.97)	**98.95** (78.08)	98.59 (72.60)	98.59 (**89.04**)	98.55 (72.60)	**98.95** (73.97)
15	98.99 (75.34)	98.86 (69.86)	**99.08** (80.82)	98.80 (68.49)	98.68 (**84.93**)	98.80 (68.49)	98.80 (68.49)
20	98.89 (75.34)	98.89 (75.34)	**99.20** (80.82)	98.89 (75.34)	98.62 (**83.56**)	98.89 (75.34)	98.89 (75.34)
25	**99.11** (75.34)	**99.11** (75.34)	98.89 (65.75)	98.99 (73.97)	98.80 (**86.30**)	98.99 (73.97)	98.99 (73.97)
30	98.99 (72.60)	98.99 (72.60)	**99.02** (72.60)	98.99 (72.60)	98.86 (**87.67**)	98.99 (72.60)	98.99 (72.60)
37.5	**99.08** (**71.23**)	**99.08** (**71.23**)	**99.08** (**71.23**)	**99.08** (**71.23**)	98.92 (**71.23**)	**99.08** (**71.23**)	**99.08** (**71.23**)
Average	**98.64** (77.91)	98.59 (77.23)	98.58 (77.40)	98.52 (76.71)	98.37 (**86.47**)	98.45 (77.23)	98.56 (76.88)
B/W	3/0 (1/3)	4/1 (2/2)	5/1 (1/3)	2/1 (2/4)	0/7 (8/0)	1/2 (2/4)	3/1 (2/3)

[11] Y. Xu, Z. Yang, and X. Pan. A novel twin support-vector machine with pinball loss. *IEEE Transactions on Neural Networks and Learning Systems*, 28(2):359–370, Feb 2017.

[12] S. Mehrkanoon, X. Huang, and J. A. K. Suykens. Non-parallel support vector classifiers with different loss functions. *Neurocomputing*, 143:294–301, 2014.

[13] Y.-H. Shao, W.-J. Chen, Z. Wang, C.-N. Li, and N.-Y. Deng. Weighted linear loss twin support vector machine for large-scale classification. *Knowledge-Based Systems*, 73:276–288, 2015.

[14] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001.

[15] A. Asuncion and D. J. Newman. UCI machine learning repository, http://www.ics.uci.edu/~mlearn/MLRepository.html. 2007.