Why pairwise is better than one-against-all or all-at-once

Tsujinishi, Daisuke

Koshiba, Yoshiaki

Abe, Shigeo

(Citation)
Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on,1:693-698

(Issue Date)
2004-07

(Resource Type)
conference paper

(Version)
Accepted Manuscript

(URL)
https://hdl.handle.net/20.500.14094/90000224

Why Pairwise Is Better than One-against-All or All-at-Once

Daisuke Tsujinishi Yoshiaki Koshiba Shigeo Abe
Graduate School of Science and Technology
Kobe University
Rokkodai, Nada, Kobe, Japan
Email: abe@eedept.kobe-u.ac.jp

***Abstract*— In this paper, first we discuss acceleration of classification by reducing support vectors. Then, we discuss multiclass least squares SVMs (LS-SVMs) that resolve unclassifiable regions for multiclass problems: fuzzy one-against-all LS-SVMs, fuzzy pairwise LS-SVMs, and all-at-once LS-SVMs. Next, we compare the three types of LS-SVMs from the standpoint of training difficulty and show that the fuzzy one-against-all LS-SVM and the all-at-once LS-SVM have similar decision boundaries when classification problems are linearly separable in the feature space. Finally, we evaluate three types of multiclass LS-SVMs for some benchmark data sets and show that classification performance of fuzzy one-against-all and one-against-all LS-SVMs are almost the same but inferior to that of fuzzy pairwise LS-SVMs.**

I. Introduction

In a least squares support vector machine (LS-SVM) [1], inequality constraints in an SVM [2] are replaced with equality constraints. Thus training of an LS-SVM results in solving a set of simultaneous linear equations, instead of a quadratic programming problem. Hence, although it is easier to handle the problem, the sparsity of a solution is lost. To avoid this, in [3], [4], training data associated with small absolute values of dual variables are pruned by repetitive training. In [5], a greedy algorithm is proposed, in which starting with only a bias term, the training pattern that minimizes the objective function is selected until some convergence test is satisfied. In [6, pp. 544–545], a simple method for calculating pre-images is discussed for the dot-product-based kernels if the pre-image exists. If this is applicable, we can replace the weight vector in the feature space with the mapping of the input vector, which results in a considerable speedup in classification.

Since LS-SVMs are formulated for two-class problems, an extension to multiclass problems is not unique. But the same techniques that are developed for multiclass SVMs can be used:

1) one-against-all SVMs [2] where one class is separated from the remaining classes,
2) pairwise SVMs [7], where any one class is separated from any other class,
3) error-correcting-output code (ECOC) SVMs [8], [9], where error correcting codes are used for improving the generalization ability, and
4) all-at-once SVMs [2], where all the decision functions are determined at once.

In the original formulations of one-against-all and pairwise SVMs, unclassifiable regions, where more than one decision function give the maximum value, exist. To solve this problem, for one-against-all SVMs continuous decision functions are used [2] and fuzzy membership functions are introduced [10]. In [11], these methods are shown to be equivalent.

To resolve unclassifiable regions for pairwise classification, decision-tree-based pairwise classification called Decision Directed Acyclic Graph [12] (DDAG) and the use of tennis tournament rules [13] are proposed. Not knowing [13], in [14] the same method is proposed and is called Adaptive Directed Acyclic Graph (ADAG). Classification by DDAGs or ADAGs is faster than by pairwise fuzzy SVMs. But the problem is that the generalization ability depends on the structure of decision trees. To solve this problem, optimization of structures is proposed [15], [16]. Like fuzzy one-against-all SVMs, for pairwise SVMs, fuzzy pairwise SVMs [17] and fuzzy pairwise LS-SVMs [18] are proposed.

For all-at-once SVMs, there is no unclassifiable region but since we need to determine all the decision functions at once, a computational cost is large [19].

In this paper, we discuss the possibility of reducing the number of support vectors and clarify classification performance of fuzzy one-against-all, fuzzy pairwise, and all-at-once LS-SVMs.

Contrary to the expectation by [6, pp. 544–545], we show that the existence of pre-images is restricted even for a dot-product-based kernel, if the input space is mapped into a higher dimensional feature space.

We compare fuzzy one-against-all, fuzzy pairwise, and all-at-once LS-SVMs from the standpoint of training difficulty. In a fuzzy one-against-all LS-SVM, a datum is classified into the class with the maximum value of the decision functions. This is the constraint imposed by the all-at-once LS-SVM. Thus the decision boundaries of both types of LS-SVMs are similar when the problem is linearly separable in the feature space. By computer simulations, we confirm that this holds for the benchmark data sets.

This paper is organized as follows. In Section II, we describe the architecture of the two-class LS-SVM and we investigate the possibility of reducing the number of support vectors. Then in Section III, we discuss LS-SVMs that resolve unclassifiable regions in multiclass problems. Finally, in Section IV we show performance comparison of three types of LS-SVMs using some benchmark data sets.

II. Two-Class LS-SVMs

A. Architecture

In this section, we describe an LS-SVM for a two-class problem. Let m-dimensional training data be $\mathbf{x}_i$ $(i = 1, ..., M)$ and their class labels be y_i, where $y_i = 1$ and $y_i = -1$ for Classes 1 and 2, respectively. We consider the linear decision function in the feature space as follows:

$$D(\mathbf{x}) = \mathbf{w}^t \mathbf{g}(\mathbf{x}) + b, \tag{1}$$

where $\mathbf{g}(\mathbf{x})$ is a mapping function that maps $\mathbf{x}$ into the l-dimensional space, $\mathbf{w}$ is an l-dimensional vector, and b is a scalar.

Assuming that the training data are not linearly separable, they satisfy

$$y_i(\mathbf{w}^t \mathbf{g}(\mathbf{x}_i) + b) = 1 - \xi_i, \quad \xi_i \geq 0 \quad i = 1, ..., M, \tag{2}$$

where ξ_i are slack variables.

The optimal separating hyperplane is determined so that the maximization of the margin, i.e., the minimum distance from the separating hyperplane to the training data in the feature space, and the minimization of the training error are achieved. Namely, minimize

$$\frac{1}{2}\mathbf{w}^t \mathbf{w} + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 \tag{3}$$

subject to the constraints (2), where C is a parameter that determines the tradeoff between the maximum margin and the minimum classification error.

To derive the dual problem of (2) and (3), we introduce the Lagrange multipliers as follows:

$$\begin{aligned} Q(\mathbf{w}, b, \alpha, \xi) &= \frac{1}{2}\mathbf{w}^t \mathbf{w} + \frac{C}{2}\sum_{i=1}^{M} \xi_i^2 \\ &\quad - \sum_{i=1}^{M} \alpha_i \{y_i(\mathbf{w}^t \mathbf{g}(\mathbf{x}_i) + b) - 1 + \xi_i\}, \end{aligned} \tag{4}$$

where $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_M)^t$ is the Lagrange multipliers, which can be positive or negative in case of LS-SVM formulation. The conditions for optimality are derived by differentiating (4) with respect to $\mathbf{w}$, ξ_i, and b and equating the resulting equations to zero:

$$\mathbf{w} = \sum_{i=1}^{M} \alpha_i y_i \mathbf{g}(\mathbf{x}_i), \quad \sum_{i=1}^{M} \alpha_i y_i = 0, \quad \alpha_i = C\xi_i. \tag{5}$$

In a matrix form, (2) and (5) are expressed by

$$\begin{bmatrix} \mathbf{\Omega} & \mathbf{Y} \\ \mathbf{Y}^t & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix}, \tag{6}$$

where $\mathbf{\Omega}$, $\mathbf{Y}$ and $\mathbf{1}$ are, respectively

$$\mathbf{\Omega}_{ij} = y_i y_j \mathbf{g}(\mathbf{x}_i)^t \mathbf{g}(\mathbf{x}_j) + \frac{\delta_{ij}}{C}, \tag{7}$$

$$\delta_{ij} = \begin{cases} 1 & i = j, \\ 0 & i \neq j, \end{cases}$$

$$\mathbf{Y} = (y_1, \cdots, y_M)^t, \tag{8}$$

$$\mathbf{1} = (1, \cdots, 1)^t. \tag{9}$$

One of the characteristic of the SVM is that it uses the technique called kernel trick. In (7), defining

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{g}(\mathbf{x})^t \mathbf{g}(\mathbf{x}'), \tag{10}$$

where $K(\mathbf{x}, \mathbf{x}')$ is a kernel function, we can avoid treating variables in the feature space. In the following study, we use the kernel functions as follows:

- linear kernels: $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^t \mathbf{x}'$,
- polynomial kernels: $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^t \mathbf{x}' + 1)^d$, where d is a positive integer,
- RBF kernels: $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2)$, where γ is a positive parameter.

B. Reducing the Number of Support Vectors

If $\mathbf{z}$ that satisfies $\mathbf{g}(\mathbf{z}) = \mathbf{w}$ exists, we can evaluate the decision function by $D(\mathbf{x}) = K(\mathbf{x}, \mathbf{z}) + b$, which will result in a considerable speedup in classification.

In [6, pp. 544–545], a simple method for calculating the pre-image is proposed if the pre-image exists and $K(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}^t \mathbf{x}')$, where $f(\cdot)$ is some scalar function. Let $\{\mathbf{e}_i, \ldots, \mathbf{e}_m\}$ be the basis vectors of the input space, where the ith element of $\mathbf{e}_i$ is 1 and others, 0. Then

$$K(\mathbf{z}, \mathbf{e}_j) = f(z_j) = \mathbf{w}^t \mathbf{g}(\mathbf{e}_j) = \sum_{i=1}^{M} \alpha_i y_i x_{ij}, \tag{11}$$

where x_{ij} is the jth element of $\mathbf{x}_i$. For the polynomial kernel, $f(z_j) = (z_j + 1)^d$. Thus, if d is odd, the inverse exists and

$$z_j = f^{-1}\left(\sum_{i=1}^{M} \alpha_i y_i x_{ij}\right). \tag{12}$$

The above equation is satisfied if the pre-image exists. Indeed, with linear kernels, $\mathbf{g}^{-1}(\mathbf{w}) = \mathbf{w}$. But if an m-dimensional vector $\mathbf{x}$ is mapped into an l-dimensional space $(l > m)$, the inverse of $\mathbf{g}(\mathbf{z})$ does not exist. Consider the case where $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^t \mathbf{x}' + 1$ and $\mathbf{g}(\mathbf{z}) = a_1 \mathbf{g}(\mathbf{x}_1) + a_2 \mathbf{g}(\mathbf{x}_2)$. Thus, $\mathbf{g}(\mathbf{x}) = (1, \mathbf{x}^t)^t$ and $l = m + 1$. Then the following $(m + 1)$ equations must be satisfied for m variables:

$$1 = a_1 + a_2, \tag{13}$$

$$z_1 = a_1 x_{11} + a_2 x_{21}, \tag{14}$$

$$\cdots$$

$$z_m = a_1 x_{1m} + a_2 x_{2m}. \tag{15}$$

The above set of simultaneous equations is solved only when (13) is satisfied.

For the polynomial kernel with degree 2 with a one-dimensional input x, $K(x, x') = (1 + xx')^2$. Thus, $\mathbf{g}(x)$ is given by

$$\mathbf{g}(x) = (1, \sqrt{2}x, x^2)^t. \tag{16}$$

Therefore, the following equations must be satisfied:

$$\begin{aligned} 1 &= a_1 + a_2, & (17)\\ z &= a_1 x_1 + a_2 x_2, & (18)\\ z^2 &= a_1 x_1^2 + a_2 x_2^2, & (19) \end{aligned}$$

which is, in general, unsolvable.

In general, a set of l equations must be satisfied for m variables. Thus, if $l \neq m$, the inverse does not exist.

This is discouraging since the inverse exists only for linear kernels. But at least for linear kernels, we should use this fact.

III. Multiclass LS-SVMs

In this section we discuss fuzzy one-against-all, fuzzy pairwise, and all-at-once LS-SVMs for n-class problems.

A. Fuzzy One-against-all LS-SVMs

For a one-against-all SVM, we determine n decision functions that separate one class from the remaining classes. Let the ith decision function, with the maximum margin, that separates class i from the remaining classes be

$$D_i(\mathbf{x}) = \mathbf{w}_i^t \, \mathbf{g}(\mathbf{x}) + b_i. \tag{20}$$

The hyperplane $D_i(\mathbf{x}) = 0$ forms the optimal separating hyperplane and if the classification problem is separable, the training data belonging to class i satisfy $D_i(\mathbf{x}) \geq 1$ and those belonging to the remaining classes satisfy $D_i(\mathbf{x}) \leq -1$.

In classification, if for the input vector $\mathbf{x}$

$$D_i(\mathbf{x}) > 0 \tag{21}$$

is satisfied for one i, $\mathbf{x}$ is classified into class i. Since only the sign of the decision function is used, the decision is discrete.

If (21) is satisfied for plural i's, or there is no i that satisfies (21), $\mathbf{x}$ is unclassifiable.

To avoid this, instead of the discrete decision functions, continuous decision functions are proposed for classification. Namely, datum $\mathbf{x}$ is classified into the class:

$$\arg \max_{i=1,\dots,n} D_i(\mathbf{x}). \tag{22}$$

Another way of avoiding unclassifiable regions is to introduce membership functions. For class i we define one-dimensional membership functions $m_{ij}(\mathbf{x})$ in the directions orthogonal to the optimal separating hyperplanes $D_j(\mathbf{x}) = 0$ as follows:

1) For $i = j$

$$m_{ii}(\mathbf{x}) = \begin{cases} 1 & \text{for} \quad D_i(\mathbf{x}) \geq 1, \\ D_i(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{23}$$

2) For $i \neq j$

$$m_{ij}(\mathbf{x}) = \begin{cases} 1 & \text{for} \quad D_j(\mathbf{x}) \leq -1, \\ -D_j(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{24}$$

For $i \neq j$, class i is on the negative side of $D_j(\mathbf{x}) = 0$.

Using $m_{ij}(\mathbf{x})\,(j = 1, \dots, n)$, we define the class i membership function of $\mathbf{x}$ using the minimum operator:

$$m_i(\mathbf{x}) = \min_{j=1,\dots,n} m_{ij}(\mathbf{x}), \tag{25}$$

or the average operator:

$$m_i(\mathbf{x}) = \frac{1}{m} \sum_{j=1,\dots,n} m_{ij}(\mathbf{x}). \tag{26}$$

The datum $\mathbf{x}$ is classified into the class

$$\arg \max_{i=1,\dots,n} m_i(\mathbf{x}). \tag{27}$$

We can prove that one-against-all SVMs with continuous decision functions and one-against-all fuzzy SVMs with minimum or average operators are equivalent in that they give the same classification result for an input.

B. Fuzzy Pairwise LS-SVMs

In pairwise classification we require a binary classifier for each possible pair of classes and the number of the total pairs is $n(n-1)/2$ for an n-class problem. The decision function for the pair of classes i and j is given by

$$D_{ij}(\mathbf{x}) = \mathbf{w}_{ij}^t \mathbf{g}(\mathbf{x}) + b_{ij}, \tag{28}$$

where $D_{ij}(\mathbf{x}) = -D_{ji}(\mathbf{x})$. Then for the datum $\mathbf{x}$ we calculate

$$D_i(\mathbf{x}) = \sum_{j \neq i, i=1}^{n} \operatorname{sign}(D_{ij}(\mathbf{x})), \tag{29}$$

and this datum is classified into the class

$$\arg \max_{i=1,\dots,n} D_i(\mathbf{x}). \tag{30}$$

If (30) is satisfied for one i, $\mathbf{x}$ is classified into class i. But if (30) is satisfied for plural i's, $\mathbf{x}$ is unclassifiable.

To avoid this, similar to fuzzy one-against-all LS-SVMs, we introduce the fuzzy membership function. First, we define the one-dimensional membership function $m_{ij}(\mathbf{x})$ in the direction orthogonal to the optimal separating hyperplane $D_{ij}(\mathbf{x})$ as follows:

$$m_{ij} = \begin{cases} 1 & \text{for} \quad D_{ij}(\mathbf{x}) \geq 1, \\ D_{ij}(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{31}$$

Like one-against-all LS-SVMs we use the average operator as well as the minimum operator. The average operator is complementary to the continuous Hamming distance used in error correcting output codes [8].

Using the minimum operator the membership function, $m_i(\mathbf{x})$, of $\mathbf{x}$ for class i is given by

$$m_i(\mathbf{x}) = \min_{j=1,\dots,n} m_{ij}(\mathbf{x}). \tag{32}$$

Using the average operator the membership function, $m_i(\mathbf{x})$, of $\mathbf{x}$ for class i is given by

$$m_i(\mathbf{x}) = \frac{1}{n-1} \sum_{j=1, j \neq i}^{n} m_{ij}(\mathbf{x}). \tag{33}$$

Using either (32) or (33), the data $\mathbf{x}$ is classified into the class

$$\arg \max_{i=1,\dots,n} m_i(\mathbf{x}). \tag{34}$$

Fuzzy pairwise LS-SVMs with minimum and average operators are not equivalent.

C. All-at-Once LS-SVMs

In an all-at-once LS-SVM, for $\mathbf{x}$ belonging to class i, we determine the decision function $D_i(\mathbf{x})$ by

$$D_i(\mathbf{x}) > D_j(\mathbf{x}) \qquad \text{for} \quad j \neq i,\, j = 1, \ldots, n. \tag{35}$$

In this formulation we need to determine the n decision functions at once [2, pp. 437–440] which results in solving a problem with larger number of variables than the above mentioned methods.

For the all-at-once LS-SVM, we minimize

$$\frac{1}{2}\sum_{j=1}^{n}\|\mathbf{w}_j\|^2 + \frac{C}{2}\sum_{i=1}^{M}\sum_{\substack{j=1,\\ j\neq y_i}}^{n}\xi_{ij}^2 \tag{36}$$

subject to the equality constraints

$$(\mathbf{w}_{y_i} - \mathbf{w}_j)^t\mathbf{g}(\mathbf{x}_i) + b_{y_i} - b_j = 1 - \xi_{ij}$$
$$\text{for} \qquad j \neq y_i, j = 1, \ldots, n, i = 1, \ldots, M, \tag{37}$$

where $y_i\,(\in \{1, \ldots, n\})$ is the class label for $\mathbf{x}_i$ and C is the margin parameter.

Introducing the Lagrange multipliers α_{ij}, we obtain

$$\begin{aligned} Q(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \;&=\; \frac{1}{2}\sum_{j=1}^{n}\|\mathbf{w}_j\|^2 + \frac{C}{2}\sum_{i=1}^{M}\sum_{\substack{j=1,\\ k\neq y_i}}^{n}\xi_{ij}^2 \\ &\quad -\sum_{i=1}^{M}\sum_{\substack{j=1,\\ j\neq y_i}}^{n}\alpha_{ij}\left((\mathbf{w}_{y_i} - \mathbf{w}_j)^t\mathbf{g}(\mathbf{x}_i)\right. \\ &\quad \left. + b_{y_i} - b_j - 1 + \xi_{ij}\right). \end{aligned} \tag{38}$$

Taking the partial derivatives of (38) with respect to $\mathbf{w}_j, b_j, \alpha_{ij}$, and ξ_{ij}, respectively and equating them to zero, we obtain the optimal conditions as follows:

$$\mathbf{w}_j = \sum_{i=1}^{M} z_{ij}\,\mathbf{g}(\mathbf{x}_i) \qquad \text{for} \quad j = 1, \ldots, n, \tag{39}$$

$$\sum_{i=1}^{M} z_{ij} = 0 \qquad \text{for} \quad j = 1, \ldots, n, \tag{40}$$

$$\alpha_{ij} = C\xi_{ij}, \quad \alpha_{ij} \geq 0$$
$$\text{for} \quad i = 1, \ldots, M,\, j \neq y_i,\, j = 1, \ldots, n, \tag{41}$$

$$(\mathbf{w}_{y_i} - \mathbf{w}_j)^t\mathbf{g}(\mathbf{x}_i) + b_{y_i} - b_j - 1 + \xi_{ij} = 0,$$
$$\text{for} \quad i = 1, \ldots, M,\, j \neq y_i,\, j = 1, \ldots, n, \tag{42}$$

where

$$z_{ij} = \begin{cases} \displaystyle\sum_{\substack{k=1,\\ k\neq y_i}}^{n}\alpha_{ik} & \text{for } j = y_i, \\ -\alpha_{ij} & \text{otherwise.} \end{cases} \tag{43}$$

Similar to a two-class problem, substituting (39) and (41) into (42), we can solve the resulting equation and (40) for α_{ij} and b_i.

The decision function is given by

$$D_i(\mathbf{x}) = \sum_{j=1}^{M} z_{ji}\,K(\mathbf{x}_j, \mathbf{x}) + b_i. \tag{44}$$

D. Training Difficulty

Table I summarizes the characteristics of the three types of LS-SVMs from the number of decision functions to be determined for n-class problems and the number of variables solved simultaneously for the M training data. In the table, M_i is the number of training data belonging to class i. For large n, the number of decision functions for a pairwise LS-SVM becomes large compared to the other two. But since the number of data trained at a time is the smallest, the training cost is usually the smallest.

TABLE I

COMPARISON OF LS-SVMS

	One-against-all	Pairwise	All-at-once
Decision functions	n	$n(n-1)/2$	n
Variables	M	$M_i + M_j$	nM

In the following we discuss the difference of LS-SVMs from the standpoint of separability. We say that training data are separable by an LS-SVM if each decision function for the LS-SVM separates training data in the feature space. This means that training data are separated correctly 100% by the LS-SVM.

Consider a one-dimensional three-class classification problem, where Class 1 is in $(-\infty, a)$, Class 2 in (a, b), and Class 3 in (b, ∞). We use linear kernels.

Since Class 1 is not separated from Classes 2 and 3 by a linear decision function, the problem is not separable by one-against-all formulation. But by pairwise formulation, by setting

$$D_{12}(x) = x - a, D_{13}(x) = -x + b, D_{23}(x) = -x + c, \tag{45}$$

where $b > c > a$, the problem is separable.

By all-at-once formulation, the problem is also separable by

$$\begin{aligned} D_1(x) &= (x - a)/(b - a), \\ D_2(x) &= -(x - a)/a, \\ D_3(x) &= 2(x - (a + b)/2))/(b - a). \end{aligned}$$

Therefore, the separation power of one-against-all formulation is lower than pairwise or all-at-once formulation.

Now compare one-against-all and all-at-once SVMs when a problem is separable by a one-against-all SVM. In a one-against-all SVM, the decision function for class i is determined so that $D_i(\mathbf{x}) > 1$ for $\mathbf{x}$ belonging to class i and $D_i(\mathbf{x}) < -1$, otherwise. But by classification using continuous decision functions or fuzzy membership functions, if

$$D_i(\mathbf{x}) > D_j(\mathbf{x}) \quad \text{for} \quad j \neq i, j = 1, \ldots, n, \tag{46}$$

$\mathbf{x}$ is classified into class i. Equation (46) is the same constraint as that of the all-at-once LS-SVM. Thus, the decision boundaries obtained by on-against-all and all-at-once LS-SVMs are similar when the training data are separable by the one-against-all LS-SVM.

Now compare the pairwise and all-at-once LS-SVMs. Separation of a smaller number of data is easier than that by a larger number of data. In addition for $n \geq 4$, the pairwise LS-SVM has larger number of decision functions. Thus, training data are more separable by pairwise LS-SVMs than by one-against-all LS-SVMs.

IV. Performance Evaluation

A. Condition of Experiments

Using the iris data [20], the numeral data for license plate recognition [21], the thyroid data [22], and the blood cell data [23] listed in Table II, we compared the performance of the fuzzy one-against-all LS-SVM, the fuzzy pairwise LS-SVM with minimum and average operators, and the all-at-once LS-SVM.

TABLE II
Benchmark Data Specification

Data	Inputs	Classes	Training data	Test data
Iris	4	3	75	75
Numeral	12	10	810	820
Thyroid	21	3	3772	3428
Blood cell	13	12	3097	3100

We used the linear, polynomial, and RBF kernels. For a fixed kernel, we determine the optimum value of C by 5-fold cross validation for the values of C from 10 to 100000. The simulations were done on an AthlonMP 2GHz PC.

B. Classification Performance

Table III shows the recognition performance of the fuzzy one-against-all LS-SVM, the fuzzy pairwise LS-SVMs with minimum and average operators, and all-at-once LS-SVM for the values of C optimized by 5-fold cross validation. The highest recognition rates of the test data are shown in boldfaces. We could not get the results of the all-at-once LS-SVM for the blood cell data due to memory overflow. In the following, the recognition rate means that of the test data.

For all the data sets, the fuzzy pairwise LS-SVMs performed best. There is no much difference between the recognition rates by the minimum and average operators, but the minimum operator was stabler; for the iris data and the blood cell data the recognition rates by the average operator with linear kernels were much lower.

Except for the iris data with linear kernels, the recognition rates of the test data for the one-against-all and all-at-once LS-SVMs are almost the same. This verifies our theoretical analysis.

C. Training Speed

Table IV shows the training time of the one-against-all, pairwise, and all-at-once LS-SVMs for the polynomial kernels with degree 2. In training the LS-SVM, we used the Cholesky factorization to solve the set of linear equations. For all the cases training of the pairwise LS-SVM was the fastest and the all-at-once LS-SVM was the slowest.[1] For the blood cell data we could not train the all-at-once LS-SVM because of the memory overflow. Therefore, as indicated in [1], we need to use iterative methods for speedup and efficient memory use.

TABLE IV
Training Time in Seconds

Data	One-against-all	Pairwise	All-at-once
Numeral	25	1	2026
Thyroid	716	409	1565
Blood cell	1593	59	—

D. Classification Speed

Table V lists the classification time for linear kernels with the values of C listed in Table III. The conventional and proposed methods mean that the weights are calculated for each datum and the weights are calculated once before classification, respectively. Except for the iris data set, which is very small, speed-up by the proposed method is evident.

TABLE V
Classification Time in Seconds

Data	Method	One-against-all	Pairwise	All-at-once
Iris	Conventional	0.01	0.00	0.01
	Proposed	0.00	0.00	0.00
Numeral	Conventional	3.13	2.72	3.19
	Proposed	0.01	0.02	0.00
Thyroid	Conventional	43.27	28.43	43.45
	Proposed	0.02	0.02	0.02
Blood Cell	Conventional	80.03	52.40	–
	Proposed	0.02	0.26	–

E. Discussions

Theoretical analysis and computer experiments indicate that it is sufficient to use one-against-all LS-SVMs instead of all-at-once LS-SVMs from the standpoint of similar classification performance and a lower computation cost.

The theoretical analysis on LS-SVMs also hold for the conventional SVMs. Namely, one-against-all SVMs and all-at-once SVMs have similar decision boundaries when problems are separated by one-against-all SVMs, and pairwise SVMs are more easily trained than one-against-all and all-at-once SVMs.

V. Conclusions

In this paper, first we discussed acceleration of classification by reducing support vectors. Then, we discussed fuzzy one-against-all LS-SVMs, fuzzy pairwise LS-SVMs, and all-at-once LS-SVMs for multiclass problems, and show that the decision boundaries of one-against-all and all-at-once LS-SVMs are similar when the problems are separable by the one-against-all LS-SVM. According to the computer experiments using several benchmark data sets, classification performance of one-against-all and all-at-once LS-SVMs were shown to be quite similar and the pairwise LS-SVMs performed best from the standpoint of generalization ability and training time.

[1] For conventional SVMs, this fact was shown in [19].

TABLE III

RECOGNITION PERFORMANCE OF ONE-AGAINST-ALL, PAIRWISE, AND ALL-AT-ONCE LS-SVMS (%)

Data	Kernel	One-against-all		Pairwise (Min)		Pairwise (Avg)		All-at-once	
		C	Test (Train.)	C	Test (Train.)	C	Test (Train.)	C	Test (Train.)
Iris	linear	50	81.33 (85.33)	3000	**97.33** (100)	50	92.00 (100)	10	89.33 (86.67)
	$d = 2$	50	94.67 (98.67)	3000	**98.67** (100)	10	97.33 (100)	10	94.67 (96.00)
	$d = 3$	10^4	90.67 (100)	3000	**96.00** (100)	1000	94.67 (100)	5000	92.00 (100)
	$d = 4$	2000	94.67 (100)	500	**96.00** (100)	500	94.67 (100)	500	94.67 (100)
	$\gamma = 1.0$	2000	96.00 (100)	10^4	**96.00** (100)	100	**96.00** (100)	$5{\cdot}10^4$	94.67 (100)
	$\gamma = 0.5$	$2{\cdot}10^4$	96.00 (100)	5000	**97.33** (100)	100	94.67 (100)	7000	96.00 (100)
Numeral	linear	10	97.50 (97.90)	10	99.27 (99.75)	10	**99.39** (99.75)	10	98.05 (99.51)
	$d = 2$	10	99.02 (99.88)	10	99.76 (100)	10	**100** (100)	10	99.51 (100)
	$d = 3$	10	99.63 (100)	10	99.63 (100)	10	**99.88** (100)	10	99.63 (100)
	$d = 4$	10	98.90 (100)	10	99.51 (100)	10	**99.63** (100)	10	98.78 (100)
	$\gamma = 1.0$	10	99.02 (99.75)	50	**99.51** (100)	50	**99.51** (100)	10	99.15 (100)
	$\gamma = 0.5$	50	99.02 (99.75)	1000	99.88 (100)	1000	**100** (100)	10	99.15 (99.75)
Thyroid	linear	50	93.41 (93.24)	100	93.73 (94.01)	50	**93.76** (93.96)	50	93.38 (93.27)
	$d = 2$	5000	93.90 (94.99)	2000	95.04 (96.29)	10^4	**95.30** (96.45)	500	93.76 (94.88)
	$d = 3$	50	93.58 (96.00)	10	94.60 (96.79)	50	**94.72** (97.32)	10	93.55 (95.86)
	$d = 4$	10	93.44 (96.79)	10	**94.78** (97.83)	10	94.63 (98.11)	100	92.24 (97.67)
	$\gamma = 1$	$5{\cdot}10^4$	94.11 (95.55)	$8{\cdot}10^4$	**95.36** (96.79)	10^5	95.10 (97.03)	$5{\cdot}10^4$	94.46 (95.89)
	$\gamma = 0.5$	10^5	93.87 (95.23)	10^5	94.87 (96.26)	10^5	**95.07** (96.37)	10^5	94.11 (95.47)
Blood Cell	linear	50	76.23 (76.33)	500	**92.65** (94.80)	10	88.29 (89.67)	—	
	$d = 2$	10^5	91.90 (94.16)	100	**94.10** (96.87)	1000	92.74 (96.16)	—	
	$d = 3$	100	93.00 (96.06)	10	**93.94** (97.19)	100	93.61 (97.64)	—	
	$d = 4$	100	93.35 (97.26)	10	93.71 (98.22)	10	**93.77** (97.90)	—	
	$\gamma = 1$	$7{\cdot}10^4$	93.16 (96.13)	$7{\cdot}10^4$	**94.16** (97.29)	$8{\cdot}10^4$	93.55 (97.55)	—	
	$\gamma = 0.5$	10^5	92.84 (95.51)	$3{\cdot}10^4$	**94.19** (97.06)	$8{\cdot}10^4$	92.94 (96.38)	—	

REFERENCES

[1] J. A. K. Suykens and J. Vandewalle, “Least Squares Support Vector Machine Classifiers,” *Neural Processing Letters*, pp. 293–300, 1999.

[2] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[3] J. A. K. Suykens, “Least Squares Support Vector Machines for Classification and Nonlinear Modelling,” *Neural Network World*, Vol. 10, Nos. 1–2, pp. 29–47, 2000.

[4] J. A. K. Suykens, S. Lukas, and J. Vandewalle, “Sparse Least Squares Support Vector Machine Classifiers,” *Proc. ESANN 2000*, pp. 37–42, 2000.

[5] G. C. Cawley and N. L. C. Talbot, “A Greedy Training Algorithm for Sparse Least-Squares Support Vector Machines,” *Proc. ICANN 2002*, pp. 681–686, 2002.

[6] B. Schölkopf and A. J. Smola, *Learning with Kernels*, MIT Press, 2002.

[7] U. H.-G. Kreßel, “Pairwise Classification and SVMs,” In B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., *Advances in Kernel Methods: Support Vector Learning*, pp. 255-268, MIT Press, 1999.

[8] T. G. Dietterich and G. Bakiri, “Solving Multiclass Learning Problems via Error-Correcting Output Codes,” *Journal of Artificial Intelligence Research*, Vol. 2, pp. 263–286, 1995.

[9] T. Kikuchi and S. Abe, “Error Correcting Output Codes vs. Fuzzy Support Vector Machines,” *Proc. ANNPR 2003*, pp. 192–196, 2003.

[10] T. Inoue and S. Abe, “Fuzzy SVMs for Pattern Classification,” *Proc. IJCNN’01*, pp. 1449–1454, 2001.

[11] S. Abe, “Analysis of Multiclass Support Vector Machines,” *Proc. CIMCA’2003*, pp. 385–396, 2003.

[12] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, “Large Margin DAGs for Multiclass Classification,” In S. A. Solla, T. K. Leen, and K.-R. Müller, Eds., *Advances in Neural Information Processing Systems 12*, pp. 547-553, MIT Press, 2000.

[13] M. Pontil and A. Verri, “Support Vector Machines for 3-D Object Recognition,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 20, No. 6, pp. 637–646, 1998.

[14] B. Kijsirikul and N. Ussivakul, “Multiclass Support Vector Machines Using Adaptive Directed Acyclic Graph,” *Proc. IJCNN 2002*, pp. 980–985, 2002.

[15] T. Phetkaew, B. Kijsirikul, and W. Rivepiboon, “Reordering Adaptive Directed Acyclic Graphs: An Improved Algorithm for Multiclass Support Vector Machines,” *Proc. IJCNN 2003*, Vol. 2, pp. 1605–1610, 2003.

[16] F. Takahashi and S. Abe, “Optimizing Directed Acyclic Graph Support Vector Machines,” *Proc. ANNPR 2003*, pp. 166–170, 2003.

[17] S. Abe and T. Inoue, “Fuzzy SVMs for Multiclass Problems,” *Proc. ESANN 2002*, pp. 113–118, 2002.

[18] D. Tsujinishi and S. Abe, “Fuzzy Least Squares Support Vector Machines for Multiclass Problems” *Neural Networks*, Vol. 16, Nos. 5–6, pp. 785-792, 2003.

[19] C.-W. Hsu and C.-J. Lin, “A Comparison of Methods for Multiclass Support Vector Machines,” *IEEE Trans. Neural Networks*, Vol. 13, 2, pp. 415–425, 2002.

[20] R. A. Fisher, “The Use of Multiple Measurements in Taxonomic Problems,” *Annals of Eugenics*, Vol. 7, pp. 179-188, 1936.

[21] H. Takenaga et al., “Input Layer Optimization of Neural Networks by Sensitivity Analysis and Its Application to Recognition of Numerals,” *Electrical Engineering in Japan*, Vol. 111, No. 4, pp. 130–138, 1991.

[22] S. M. Weiss and I. Kapouleas, “An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods,” *Proc. IJCAI-99, Workshop ML3*, pp. 55-60, 1999.

[23] A. Hashizume, J. Motoike, and R. Yabe, “Fully Automated Blood Cell Differential System and Its Application,” *Proc. IUPAC 3rd International Congress on Automation and New Technology in the Clinical Laboratory*, pp. 297–302, 1988.