Sparse Least Squares Support Vector Machines by Forward Selection Based on Linear Discriminant Analysis

Shigeo Abe

Graduate School of Engineering
Kobe University
Rokkodai, Nada, Kobe, Japan
abe@kobe-u.ac.jp
http://www2.eedept.kobe-u.ac.jp/~abe

Abstract. In our previous work, we have developed sparse least squares support vector machines (sparse LS SVMs) trained in the reduced empirical feature space, spanned by the independent training data selected by the Cholesky factorization. In this paper, we propose selecting the independent training data by forward selection based on linear discriminant analysis in the empirical feature space. Namely, starting from the empty set, we add a training datum that maximally separates two classes in the empirical feature space. To calculate the separability in the empirical feature space we use linear discriminant analysis (LDA), which is equivalent to kernel discriminant analysis in the feature space. If the matrix associated with the LDA is singular, we consider that the datum does not contribute to the class separation and permanently delete it from the candidates of addition. We stop the addition of data when the objective function of LDA does not increase more than the prescribed value. By computer experiments for two-class and multi-class problems we show that in most cases we can reduce the number of support vectors more than with the previous method.

1 Introduction

A least squares support vector machine (LS SVM) [1] is a variant of a regular SVM [2]. One of the advantages of LS SVMs over SVMs is that we only need to solve a set of linear equations instead of a quadratic programming program. But the major disadvantage of LS SVMs is that all the training data become support vectors instead of sparse support vectors for SVMs. To solve this problem, in [1], support vectors with small absolute values of the associated dual variables are pruned and the LS SVM is retrained using the reduced training data set. This process is iterated until sufficient sparsity is realized. In [3], LS SVMs are reformulated using the kernel expansion of the square of Euclidian norm of the weight vector in the decision function. But the above pruning method is used to reduce support vectors. Because the training data are reduced during pruning, information for the deleted training data is lost for the trained LS

L. Prevost, S. Marinai, and F. Schwenker (Eds.): ANNPR 2008, LNAI 5064, pp. 54–65, 2008.

SVM. To overcome this problem, in [4], independent data in the feature space are selected from the training data, and using the selected training data the solution is obtained by the least squares method using all the training data. Along the line of kernel expansion, there are some approaches to realize sparse kernel expansion by forward selection of basis vectors based on some criterion such as least squares errors [5,6,7]. Based on the concept of the empirical feature space [8], which is closely related to kernel expansion, in [9] a sparse LS SVM is developed restricting the dimension of the empirical feature space by the Cholesky factorization.

Instead of the Cholesky factorization used in [9], in this paper we propose using forward selection based on the class separability calculated by the linear discriminant analysis (LDA) in the empirical feature space. Namely, starting from the empty set, we add training datum, one at a time, that maximally separates two classes in the empirical feature space. To calculate the separability in the empirical feature space we use LDA in the empirical feature space. Linear discriminant analysis in the empirical feature space is equivalent to kernel discriminant analysis (KDA) in the feature space, but since the variables in the empirical feature space are treated explicitly, the calculation of LDA is much faster. If the matrix associated with LDA is singular, we consider that the datum does not contribute to the class separation and permanently delete it from the candidates of addition. In addition to this, using the incremental calculation of LDA, we speed up forward selection. We stop the addition of data when the objective function of LDA does not increase more than the prescribed value. We evaluate the proposed method using two-class and multi-class problems.

In Section 2, we summarize sparse LS SVMs trained in the empirical feature space, and in Section 3 we discuss forward selection of independent variables based on LDA. In Section 4, we evaluate the validity of the proposed method by computer experiments.

2 Sparse Least Squares Support Vector Machines

Let the M training data pairs be $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_M, y_M)$, where $\mathbf{x}_i$ and y_i are the m-dimensional input vector and the associated class label, and $y_i = 1$ and -1 if $\mathbf{x}_i$ belongs to Classes 1 and 2, respectively. In training LS SVMs in the empirical feature space we need to transform input variables into variables in the empirical feature space. To speed up generating the empirical feature space we select independent training data that span the empirical feature space [9]. Let the $N\,(\leq M)$ training data $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_N}$ be independent in the empirical feature space, where $\mathbf{x}_{i_j} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ and $j = 1, \ldots, N$. Then, we use the following mapping function: $\mathbf{h}(\mathbf{x}) = (H(\mathbf{x}_{i_1}, \mathbf{x}), \ldots, H(\mathbf{x}_{i_N}, \mathbf{x}))^T$, where $H(\mathbf{x}, \mathbf{x}')$ is a kernel. By this formulation, $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_N}$ become support vectors. Thus, support vectors do not change even if the margin parameter changes. And the number of support vectors is the number of selected independent training data that span the empirical feature space. Then reducing N without deteriorating the generalization ability we can realize sparse LS SVMs.

The LS SVM in the empirical feature space is trained by solving

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

$$\text{subject to} \quad \mathbf{v}^T\mathbf{h}(\mathbf{x}_i) + b = y_i - \xi_i \quad \text{for} \quad i = 1, \ldots, M, \tag{2}$$

where $\mathbf{v}$ is the N-dimensional vector, b is the bias term, ξ_i is the slack variable for $\mathbf{x}_i$, and C is the margin parameter.

Substituting (2) into (1) and minimizing the resultant objective function, we obtain

$$b = \frac{1}{M}\sum_{i=1}^{M}(y_i - \mathbf{v}^T\mathbf{h}(\mathbf{x}_i)), \tag{3}$$

$$\left(\frac{1}{C} + \sum_{i=1}^{M}\mathbf{h}(\mathbf{x}_i)\,\mathbf{h}^T(\mathbf{x}_i) - \frac{1}{M}\sum_{i,j=1}^{M}\mathbf{h}(\mathbf{x}_i)\,\mathbf{h}^T(\mathbf{x}_j)\right)\mathbf{v} = \sum_{i=1}^{M} y_i\,\mathbf{h}(\mathbf{x}_i) - \frac{1}{M}\sum_{i,j=1}^{M} y_i\,\mathbf{h}(\mathbf{x}_j). \tag{4}$$

We call the LS SVM obtained by solving (4) and (3) primal LS SVM if N is the same as the dimension of the mapped training data in the feature space. If N is smaller, the primal LS SVM is called sparse LS SVM. We call the LS SVM in dual form [1] dual LS SVM.

3 Selection of Independent Data

3.1 Idea

In [9], independent data are selected by Cholesky factorization of the kernel matrix. During factorization, if the argument of the square root associated with the diagonal element is smaller than the prescribed threshold value, we delete the associated row and column and continue decomposing the matrix. By increasing the threshold value, we can increase the sparsity of the LS SVM.

So long as we select the independent data that span the empirical feature space, different sets of independent data do not affect the generalization ability of the LS SVM, because the different sets span the same empirical feature space.

But the different sets of the independent data for the reduced empirical feature space span different reduced empirical feature spaces. Thus, the processing order of the training data affects the generalization ability of the LS SVM. Therefore, selection may be inefficient if the empirical feature space is reduced.

To overcome this problem, we consider selecting independent data that maximally separate two classes using LDA calculated in the reduced empirical feature space. In the following first we summarize the selection of independent data by the Cholesky factorization and then discuss our proposed method based on LDA.

3.2 Selection of Independent Data by Cholesky Factorization

Let the kernel matrix $H = \{H(\mathbf{x}_i, \mathbf{x}_j)\}$ $(i, j = 1, \ldots, M)$ be positive definite. Then H is decomposed by the Cholesky factorization into $H = L\,L^T$, where L is the regular lower triangular matrix and each element L_{ij} is given by

$$L_{op} = \frac{H_{op} - \sum_{n=1}^{p-1} L_{pn} L_{on}}{L_{pp}} \qquad \text{for} \quad o = 1, \ldots, M, \quad p = 1, \ldots, o-1, \tag{5}$$

$$L_{aa} = \sqrt{H_{aa} - \sum_{n=1}^{a-1} L_{an}^2} \qquad \text{for} \quad a = 1, 2, \ldots, M. \tag{6}$$

Here, $H_{ij} = H(\mathbf{x}_i, \mathbf{x}_j)$.

Then during the Cholesky factorization, if the argument of the square root associated with the diagonal element is smaller than the prescribed value η_{C} (> 0):

$$H_{aa} - \sum_{n=1}^{a-1} L_{an}^2 \leq \eta_{\mathrm{C}}, \tag{7}$$

we delete the associated row and column and continue decomposing the matrix. The training data that are not deleted in the Cholesky factorization are independent.

The above Cholesky factorization can be done incrementally [10,11]. Namely, instead of calculating the full kernel matrix in advance, if (7) is not satisfied, we overwrite the ath column and row with those newly calculated using the previously selected data and $\mathbf{x}_{a+1}$. Thus the dimension of L is the number of selected training data, not the number of training data.

To increase sparsity of LS SVMs, we increase the value of η_{C}. The optimal value is determined by cross-validation. We call thus trained LS SVMs sparse LS SVMs by Cholesky factorization, sparse LS SVMs (C) for short.

3.3 Linear Discriminant Analysis in the Empirical Feature Space

We formulate linear discriminant analysis in the empirical feature space, which is equivalent to kernel discriminant analysis in the feature space. To make notations simpler, we redefine the training data: Let the sets of m-dimensional data belonging to Class i $(i = 1, 2)$ be $\{\mathbf{x}_1^i, \ldots, \mathbf{x}_{M_i}^i\}$, where M_i is the number of data belonging to Class i. Now we find the N-dimensional vector $\mathbf{w}$ in which the two classes are separated maximally in the direction of $\mathbf{w}$ in the empirical feature space.

The projection of $\mathbf{h}(\mathbf{x})$ on $\mathbf{w}$ is $\mathbf{w}^T\,\mathbf{h}(\mathbf{x})/\|\mathbf{w}\|$. In the following we assume that $\|\mathbf{w}\| = 1$. We find such $\mathbf{w}$ that maximizes the difference of the centers and minimizes the variance of the projected data.

The square difference of the centers of the projected data, d^2, is

$$d^2 = (\mathbf{w}^T(\mathbf{c}_1 - \mathbf{c}_2))^2 = \mathbf{w}^T(\mathbf{c}_1 - \mathbf{c}_2)\,(\mathbf{c}_1 - \mathbf{c}_2)^T\mathbf{w}, \tag{8}$$

where $\mathbf{c}_i$ are the centers of class i data:

$$\mathbf{c}_i = \frac{1}{M_i} \sum_{j=1}^{M_i} \mathbf{h}(\mathbf{x}_j^i) \qquad \text{for} \quad i = 1, 2. \tag{9}$$

We define $Q_B = (\mathbf{c}_1 - \mathbf{c}_2)(\mathbf{c}_1 - \mathbf{c}_2)^T$ and call Q_B the between-class scatter matrix.

The variance of the projected data is $s^2 = \mathbf{w}^T Q_W \mathbf{w}$, where

$$Q_W = \frac{1}{M} \sum_{j=1}^{M} \mathbf{h}(\mathbf{x}_j)\, \mathbf{h}(\mathbf{x}_j)^T - \mathbf{c}\, \mathbf{c}^T, \quad \mathbf{c} = \frac{1}{M} \sum_{j=1}^{M} \mathbf{h}(\mathbf{x}_j) = \frac{M_1 \mathbf{c}_1 + M_2 \mathbf{c}_2}{M_1 + M_2}. \tag{10}$$

We call Q_W the within-class scatter matrix.

Now, we want to maximize

$$J(\mathbf{w}) = \frac{d^2}{s^2} = \frac{\mathbf{w}^T Q_B \mathbf{w}}{\mathbf{w}^T Q_W \mathbf{w}}. \tag{11}$$

Taking the partial derivative of (11) with respect to $\mathbf{w}$ and equating the resulting equation to zero, we obtain the following generalized eigenvalue problem:

$$Q_B \mathbf{w} = \lambda Q_W \mathbf{w}, \tag{12}$$

where λ is a generalized eigenvalue.

Substituting

$$Q_W \mathbf{w} = \mathbf{c}_1 - \mathbf{c}_2 \tag{13}$$

into the left-hand side of (12), we obtain $(\mathbf{w}^T Q_W \mathbf{w}) Q_W \mathbf{w} = \lambda Q_W \mathbf{w}$. Thus, by letting $\lambda = \mathbf{w}^T Q_W \mathbf{w}$, (13) is a solution of (12).

If Q_W is positive definite, the optimum $\mathbf{w}$, $\mathbf{w}_{\text{opt}}$, is given by

$$\mathbf{w}_{\text{opt}} = Q_W^{-1} (\mathbf{c}_1 - \mathbf{c}_2). \tag{14}$$

If Q_W is positive semi-definite, i.e., singular, one way to overcome singularity is to add positive values to the diagonal elements [12]:

$$\mathbf{w}_{\text{opt}} = (Q_W + \varepsilon I)^{-1} (\mathbf{c}_1 - \mathbf{c}_2), \tag{15}$$

where ε is a small positive parameter.

Assuming that Q_W is positive definite, we substitute (14) into (11) and obtain

$$J(\mathbf{w}_{\text{opt}}) = (\mathbf{c}_1 - \mathbf{c}_2)^T \mathbf{w}_{\text{opt}}. \tag{16}$$

Linear discriminant analysis in the empirical feature space discussed above is equivalent to kernel discriminant analysis in the feature space, but since we can explicitly treat the variables in the empirical feature space, the calculation is much simpler.

3.4 Forward Selection

Starting from an empty set we add one datum at a time that maximizes (11) if the datum is added. Let the set of selected data indices be S^k and the set of remaining data indices be T^k, where k denotes that k data points are selected. Initially $S^0 = \phi$ and $T^0 = \{1, \ldots, M\}$. Let S^k_j denote that $\mathbf{x}_j$ for $j \in T^k$ is temporarily added to S^k. Let $\mathbf{h}^{k,j}(\mathbf{x})$ be the mapping function with $\mathbf{x}_j$ temporarily added to the selected data with indices in S^k:

$$\mathbf{h}^{k,j}(\mathbf{x}) = (H(\mathbf{x}_{i_1}, \mathbf{x}), \ldots, H(\mathbf{x}_{i_k}, \mathbf{x}), H(\mathbf{x}_j, \mathbf{x}))^T, \tag{17}$$

where $S^k = \{i_1, \ldots, i_k\}$. And let $J^{k,j}_{\rm opt}$ be the optimum value of the objective function with the mapping function $\mathbf{h}^{k,j}(\mathbf{x})$. Then we calculate

$$j_{\rm opt} = \arg_j J^{k,j}_{\rm opt} \qquad \text{for} \quad j \in T^k \tag{18}$$

and if the addition of $\mathbf{x}_{j_{\rm opt}}$ results in a sufficient increase in the objective function:

$$\left(J^{k,j_{\rm opt}}_{\rm opt} - J^k_{\rm opt}\right) / J^{k,j_{\rm opt}}_{\rm opt} \geq \eta_{\rm L}, \tag{19}$$

where $\eta_{\rm L}$ is a positive parameter, we increment k by 1 and add $j_{\rm opt}$ to S^k and delete it from T^k. If the above equation does not hold we stop forward selection. We must notice that $J^{k,j}_{\rm opt}$ is non-decreasing for the addition of data [13]. Thus the left-hand side of (19) is non-negative.

If the addition of a datum results in the singularity of $Q^{k,j}_{\mathbf{w}}$, where $Q^{k,j}_{\mathbf{w}}$ is the within-class scatter matrix evaluated using the data with $S^{k,j}$ indices, we consider the datum does not give useful information in addition to the already selected data. Thus, instead of adding a small value we do not consider this datum for a candidate of addition. This is equivalent to calculating the pseudo-inverse of $Q^{k,j}_{\mathbf{w}}$.

The necessary and sufficient condition of a matrix being positive definite is that all the principal minors are positive. And notice that the exchange of two rows and then the exchange of the associated two columns do not change the singularity of the matrix. Thus, if $\mathbf{x}_j$ causes the singularity of $Q^{k,j}_{\mathbf{w}}$, later addition will always cause singularity of the matrix. Namely, we can delete j from T^k permanently. If there are many training data that cause singularity of the matrix, forward selection becomes efficient.

Thus the procedure of independent data selection is as follows.

1. Set $S^0 = \phi$, $T^0 = \{1, \ldots, M\}$, and $k = 0$. Calculate $j_{\rm opt}$ given by (18) and set $S^1 = \{j_{\rm opt}\}$, $T^1 = T^0 - \{j_{\rm opt}\}$, and $k = 1$.
2. If for some $j \in T^k$, $Q^{k,j}_{\mathbf{w}}$ is singular, permanently delete j from T^k and calculate $j_{\rm opt}$ given by (18). If (19) is satisfied, go to Step 3. Otherwise terminate the algorithm.
3. Set $S^{k+1} = S^k \cup \{j_{\rm opt}\}$ and $T^{k+1} = T^k - \{j_{\rm opt}\}$. Increment k by 1 and go to Step2.

Keeping the Cholesky factorization of $Q_{\mathbf{w}}^k$, the Cholesky factorization of $Q_{\mathbf{w}}^{k,j}$ is done incrementally; namely, using the factorization of $Q_{\mathbf{w}}^k$, the factorization of $Q_{\mathbf{w}}^{k,j}$ is obtained by calculating the $(k+1)$st diagonal element and column elements. This accelerates the calculation of the inverse of the within-class scatter matrix.

We call thus trained sparse LS SVM sparse LS SVM by forward selection, sparse LS SVM (L) for short.

4 Performance Evaluation

We compared the generalization ability and sparsity of primal, sparse, and dual LS SVMs using two groups of data sets: (1) two-class data sets [14,15] and (2) multi-class data sets [11,16]. We also evaluated regular SVMs to compare sparsity.

We normalized the input ranges into $[0, 1]$ and used RBF kernels. For the primal LS SVM we set $\eta_{\mathrm{C}} = 10^{-9}$ and for the primal and dual LS SVMs, we determined the parameters C and γ by fivefold cross-validation; the value of C was selected from among {1,10,50, 100, 500, 1000, 2000, 3000, 5000, 8000, 10000, 50000, 100000}, the value of γ from among $\{0.1, 0.5, 1, 5, 10, 15\}$. For the sparse LS SVMs, we used the value of γ determined for the primal LS SVM, and determined the value of η_{C} from $\{10^{-4}, 10^{-3}, 10^{-2}, 0.05, 0.2\}$ and the value of η_{L} from $\{10^{-4}, 10^{-3}\}$ by fivefold cross-validation. We measured the computation time using a workstation (3.6GHz, 2GB memory, Linux operating system).

Then we compared the sparse LS SVM (L) with the methods discussed in [5]. We used diagonal Mahalanobis kernels [17] since in [5] the training inputs were converted into those with zero means and unit variances. We selected the value of the scale factor δ from $[0.1, 0.5, 1.0, 1.5, 2.0]$ and the value of C from $[1, 10, 50, 100, 500, 1000]$. We determined these values setting $\eta_{\mathrm{L}} = 10^{-2}$ and then for the determined values we selected, from $i \times 10^{-4}\,(i = 1, \ldots, 9)$, the largest value of η_{L} that realize the generalization ability comparable with that in [5].

4.1 Evaluation for Two-Class Problems

The two-class classification problems have 100 or 20 training data sets and their corresponding test data sets. We determined the parameter values by fivefold cross-validation for the first five training data sets.

Table 1 lists the determined parameters. In the table "Sparse (C)" and "Sparse (L)" denote the sparse LS SVM by the Cholesky factorization and that by forward selection proposed in this paper, respectively. The values of γ are not always the same for primal and dual problems. In most cases the values of η_{L} were 10^{-4} and were more stable than those of η_{C}. The table also lists the parameter values for the SVM. The values of γ are similar for the LS SVM and SVM.

Table 2 shows the average classification errors and standard deviations. Excluding the SVM, we statistically analyzed the average and standard deviations with the significance level of 0.05. Numerals in italic show that they are statistically inferior. Primal and sparse solutions for the ringnorm problem, primal

Table 1. Parameter setting for two-class problems

Data	Primal		Sparse (C)		Sparse (L)		Dual		SVM	
	γ	C	C	η_{C}	C	η_{L}	γ	C	γ	C
Banana	10	10^5	10^6	10^{-4}	10^5	10^{-4}	10	500	15	100
B. Cancer	0.5	500	1000	10^{-4}	1000	10^{-4}	0.5	10	1	10
Diabetes	1	500	10^4	10^{-2}	2000	10^{-3}	10	1	10	1
German	1	100	50	10^{-3}	100	10^{-4}	0.5	50	5	1
Heart	0.1	100	50	10^{-4}	1000	10^{-4}	0.1	10	0.1	50
Image	10	10^6	10^7	10^{-4}	10^8	10^{-4}	10	3000	10	1000
Ringnorm	0.1	1	10	10^{-3}	50	10^{-4}	10	1	15	1
F. Solar	0.5	100	500	10^{-3}	500	10^{-4}	0.1	100	1	1
Splice	5	100	100	0.2	500	10^{-4}	5	50	10	10
Thyroid	10	500	1000	10^{-4}	10^5	10^{-4}	10	50	5	1000
Titanic	5	100	100	10^{-2}	10	10^{-3}	0.5	500	10	10
Twonorm	0.1	1	100	10^{-5}	3000	10^{-3}	0.1	10	1	1
Waveform	10	10	1	5×10^{-2}	10	10^{-4}	10	1	5	10

Table 2. Comparison of the average classification errors (%) and the standard deviations of the errors

Data	Primal	Sparse (C)	Sparse (L)	Dual	SVM
Banana	*11.0*±0.55	*10.9*±0.54	*11.2*±0.63	10.7±0.52	10.4±0.46
B. Cancer	25.5±4.3	25.7±4.2	25.5±4.3	25.7±4.5	25.6±4.5
Diabetes	23.0±1.8	23.0±1.7	23.1±1.8	23.2±1.7	23.4±1.7
German	23.6±2.0	23.8±2.1	23.7±2.0	23.3±2.1	23.8±2.1
Heart	16.4±3.4	16.4±3.4	16.3±3.1	16.0±3.4	16.1±3.1
Image	2.89±0.37	2.85±0.38	*3.08*±0.38	2.66±0.37	2.84±0.50
Ringnorm	*6.02*±*3.0*	*7.56*±*6.3*	*6.20*±*2.4*	4.08±0.58	2.64±0.35
F. Solar	33.3±1.5	33.3±1.5	33.3±1.6	33.3±1.6	32.3±1.8
Splice	11.2±0.48	11.2±0.61	11.4±0.53	11.3±0.51	10.8±0.71
Thyroid	*5.59*±2.6	*5.60*±2.7	5.15±2.7	4.84±2.5	4.05±2.3
Titanic	22.5±0.94	22.5±0.94	22.7±0.85	22.5±0.97	22.4±1.0
Twonorm	*3.91*±*1.9*	*2.10*±*0.63*	*2.68*±0.22	1.90±*0.61*	2.02±0.64
Waveform	9.76±0.38	9.68±0.32	9.74±0.38	*14.9*±*0.98*	10.3±0.40

solutions for the twonorm problem and dual solutions for the waveform problem show significantly inferior performance. The inferior solutions arose because of imprecise model selection [9].

Comparing the results of the LS SVMs with the SVM, the SVM showed the better generalization ability for some problems than LS SVMs. Or LS SVMs are not robust for parameter changes.

Table 3 lists the number of support vectors. The smallest number of support vectors in LS SVMs is shown in boldface. The number of support vectors for primal solutions is the number of training data at most. The numbers of support vectors for the sparse LS SVMs are, in general, much smaller than those for the primal and dual LS SVM. This tendency is evident especially for the sparse LS SVM (L); except for three problems it performed best.

Table 3. Comparison of support vectors

Data	Primal	Sparse (C)	Sparse (L)	Dual	SVM
Banana	93	42	**33**	400	173
B. Cancer	187	101	**67**	200	118
Diabetes	447	**22**	34	468	268
German	700	386	**351**	700	416
Heart	170	68	**34**	170	74
Image	1215	476	**198**	1300	149
Ringnorm	400	21	**10**	400	131
F. Solar	82	**16**	37	666	522
Splice	977	921	**619**	1000	749
Thyroid	140	70	**55**	140	13
Titanic	11	11	**9**	150	113
Twonorm	400	169	**8**	400	193
Waveform	400	**233**	313	400	114

Table 4. Comparison of computation time in seconds

Data	Primal	Sparse (C)	Sparse (L)	Dual	SVM
Banana	9.4	2.1	**1.4**	1.0	0.4
B. Cancer	6.2	1.8	**1.0**	0.05	0.3
Diabetes	283	**0.9**	2.0	0.4	1.3
German	1550	**467**	822	1.4	7.5
Heart	3.7	0.6	**0.2**	0.4	0.09
Image	16181	2489	**1123**	10.7	1.5
Ringnorm	168	0.9	**0.6**	1.8	0.9
F. Solar	20	**4.8**	7.5	1.2	7.8
Splice	6170	**5503**	6845	5.6	13
Thyroid	1.7	0.4	**0.3**	0.02	0.01
Titanic	**0.04**	**0.04**	0.05	0.11	0.2
Twonorm	167	30	**0.6**	1.9	1.4
Waveform	167	**58**	110	1.3	0.6

In calculating the classification errors listed in Table 2, we measured the computation time of training and testing for the 100 or 20 data sets in a classification problem. Then we calculated the average computation time for a training data set and its associated test data set. Table 4 lists the results. As a reference we include the computation time for the SVM, which was trained by the primal-dual interior-point methods combined with the decomposition technique. Training of primal problems is slow especially for diabetes, german, image, and splice problems. Except for german, image, splice, and waveform computation time for sparse LS SVMs (L) is the shortest; speed-up was mostly caused by the frequent matrix singularity. But for those four problems, matrix singularity was rare and because of relatively large training data, forward selection took time.

4.2 Evaluation for Multi-class Problems

Each multi-class problem has one training data set and one test data set. We used fuzzy pairwise LS SVMs with minimum operators [11] to resolve unclassifiable regions. For comparisons we also used fuzzy pairwise SVMs.

Table 5. Parameter setting for multi-class problems

Data	Sparse (C)		Sparse (L)		Dual		SVM	
	C	η_C	C	η_L	C	γ	γ	C
Iris	10^6	10^{-4}	10^6	10^{-4}	0.1	500	0.1	5000
Numeral	10^6	10^{-4}	10^7	10^{-3}	0.1	100	5	10
Thyroid (M)	10^8	10^{-3}	10^9	10^{-4}	10	10^6	5	10^5
Blood cell	10^7	10^{-4}	10^7	10^{-4}	1	7000	10	1000
H-50	10^8	10^{-3}	10^5	10^{-4}	5	7000	10	500
H-13	10^9	10^{-4}	10^9	10^{-4}	1	10^8	10	3000
H-105	3000	10^{-2}	2000	10^{-3}	10	50	10	50

Table 6. Comparison of classification errors (%) and the numbers of support vectors

Data	Sparse (C)		Sparse (L)		Dual		SVM	
	ERs	SVs	ERs	SVs	ERs	SVs	ERs	SVs
Iris	2.67	8	4.00	**4**	2.67	50	5.33	10
Numeral	**0.73**	18	**0.73**	**4**	0.85	162	0.61	8
Thyroid (M)	4.87	354	4.96	**160**	**4.52**	2515	2.71	75
Blood cell	5.68	141	6.13	**25**	**5.65**	516	7.16	21
H-50	0.80	148	**0.74**	**37**	0.80	236	0.74	16
H-13	0.49	39	0.48	**23**	**0.14**	441	0.37	10
H-105	**0**	259	0.02	**15**	**0**	441	0	26

Table 7. Comparison of computation time in seconds

Data	Sparse (C)	Sparse (L)	Dual	SVM
Iris	**0.005**	0.007	0.02	0.01
Numeral	2.9	**1.6**	9.8	2.0
Thyroid (M)	39525	**27487**	524	9.1
Blood cell	364	**181**	210	9.6
H-50	12127	**1652**	2589	128
H-13	2387	**1585**	5108	368
H-105	112175	**1176**	14184	358

Table 5 lists the parameters determined by cross-validation. The values of C of sparse solutions are larger than those of dual solutions. The dual LS SVM and the SVM selected similar γ values, but unlike two-class problems, they selected the similar C values.

Table 6 lists the classification errors and the average number of support vectors per class pair. Excluding those of SVMs, the smallest errors and the support vectors are shown in boldface. Including SVMs, the difference of the classification errors is small. The number of support vectors for the sparse LS SVM (L) is the smallest among LS SVMs and sometimes smaller than those of SVMs. Therefore, sufficient sparsity is realized by the sparse LS SVMs (L).

Table 7 lists the computation time of training and testing. Between sparse LS SVMs, the shorter computation time is shown in boldface. Except for the iris

problem, computation time of sparse LS SVMs (L) is shorter. Comparing with the dual LS SVM and the SVM, the dual LS SVM is slower because of the slow Cholesky factorization for large matrices.

4.3 Comparison with Other Methods

We used the two-class ringnorm data set with 3000 training data and 4400 test data and the 6-class satimage data set with 4435 training data and 2000 test data used in [5]. Since training took time, we used pairwise LS SVMs instead of one-against-all LS SVMs in [5]. Table 8 shows the parameter setting for the sparse LS SVM (L) and the comparison results. The results other than Sparse (L) are from [5]. We measured the computation time by a 3.0GHz Windows machine with 2Gbyte memory, while in [5] the training time was measured by a 2.4GHz Windows machine with 1Gbyte memory. The "SVs" row denotes the total number of distinct support vectors and the "Rate" row denotes the recognition rate for the test data set. For the satimage data set we could not measure time but it took long time. From the table, the numbers of support vectors for Sparse (L) are smallest with comparable recognition rates. The training time for the ringnorm data set is comparable, but for the satimage data set, it was very slow because we did not use the subset selection method used in [5].

Table 8. Parameter setting and comparison with other methods

Data	Parm	Sparse (L)	Term	Sparse (L)	PFSALS-SVM	KMP	SGGP
Ringnorm	δ	1.5	SVs	**6**	661	77	74
	C	1	Rate	**98.64**	98.52	98.11	98.27
	η_{L}	10^{-3}	Time	9.9	7.1	7.5	50
Satimage	δ	1.5	SVs	1581	1726	—	—
	C	100	Rate	92.05	**92.25**	—	—
	η_{L}	2×10^{-4}	Time	—	50	—	—

5 Conclusions

In this paper we proposed sparse LS SVMs by forward selection of independent data based on linear discriminant analysis in the empirical feature space. Namely, starting from the empty set, we add the training data that maximally separate two-classes in the empirical feature space. We measure the class separability by the objective function of linear discriminant analysis. To speed up forward selection, we exclude data that cause matrix singularity from later addition and use the incremental calculation in calculating the inverse of the within-class scatter matrix. For most of the two-class and multi-class problems tested, sparsity of the solutions was increased drastically compared to the method using the Cholesky factorization in reducing the dimension of the empirical feature space and for most of the problems training of LS SVMs by forward selection was the fastest. For comparison with other methods, sparsity of the solutions was the highest.

References

1. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific Publishing, Singapore (2002)
2. Vapnik, V.N.: Statistical Learning Theory. John Wiley & Sons, Chichester (1998)
3. Cawley, G.C., Talbot, N.L.C.: Improved sparse least-squares support vector machines. Neurocomputing 48, 1025–1031 (2002)
4. Valyon, J., Horváth, G.: A sparse least squares support vector machine classifier. In: Proc. IJCNN 2004, vol. 1, pp. 543–548 (2004)
5. Jiao, L., Bo, L., Wang, L.: Fast sparse approximation for least squares support vector machine. IEEE Trans. Neural Networks 18(3), 685–697 (2007)
6. Smola, A.J., Bartlett, P.L.: Sparse greedy Gaussian process regression. Advances in Neural Information Processing Systems 13, 619–625 (2001)
7. Vincent, P., Bengio, Y.: Kernel matching pursuit. Machine Learning 48(1-3), 165–187 (2002)
8. Xiong, H., Swamy, M.N.S., Ahmad, M.O.: Optimizing the kernel in the empirical feature space. IEEE Trans. Neural Networks 16(2), 460–474 (2005)
9. Abe, S.: Sparse least squares support vector training in the reduced empirical feature space. Pattern Analysis and Applications 10(3), 203–214 (2007)
10. Kaieda, K., Abe, S.: KPCA-based training of a kernel fuzzy classifier with ellipsoidal regions. International Journal of Approximate Reasoning 37(3), 145–253 (2004)
11. Abe, S.: Support Vector Machines for Pattern Classification. Springer, Heidelberg (2005)
12. Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Müller, K.-R.: Fisher discriminant analysis with kernels. In: Proc. NNSP 1999, pp. 41–48 (1999)
13. Ashihara, M., Abe, S.: Feature selection based on kernel discriminant analysis. In: Kollias, S., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4132, pp. 282–291. Springer, Heidelberg (2006)
14. Rätsch, G., Onoda, T., Müller, K.-R.: Soft margins for AdaBoost. Machine Learning 42(3), 287–320 (2001)
15. `http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm`
16. `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/`
17. Abe, S.: Training of support vector machines with Mahalanobis kernels. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3697, pp. 571–576. Springer, Heidelberg (2005)