Sparse support vector machines trained in the reduced empirical feature space

Iwamura, Kazuki

Abe, Shigeo

(Citation)
Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on:2399-2405

(Issue Date)
2008-06

(Resource Type)
conference paper

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/90000925

Sparse Support Vector Machines Trained in the Reduced Empirical Feature Space

Kazuki Iwamura and Shigeo Abe

***Abstract*— We discuss sparse support vector machines (sparse SVMs) trained in the reduced empirical feature space. Namely, we select the linearly independent training data by the Cholesky factorization of the kernel matrix, and train the SVM in the dual form in the reduced empirical feature space. Since the mapped linearly independent training data span the empirical feature space, the linearly independent training data become support vectors. Thus if the number of linearly independent data is smaller than the number of support vectors trained in the feature space, sparsity is increased. By computer experiments we show that in most cases we can reduce the number of support vectors without deteriorating the generalization ability.**

I. Introduction

Sparse solutions are one of the advantages of support vector machines (SVMs) in that among training data only support vectors are necessary to represent a solution. But for a difficult classification problem with a large number of training data, many training data may become support vectors and thus the classification speed may be slow [1]. There are many approaches to overcome this problem [2], [3], [4], [5], [6], [7], [8]. Burges proposed a method for reducing the number of support vectors after training. Keerthi et al. [4] proposed training L2 support vector machines in the primal form. The idea is to select basis vectors by forward selection and for the selected basis vectors train support vector machines by Newton's method. This process is iterated until some stopping condition is satisfied. Wu et al. [5] imposed, as a constraint, the weight vector that is expressed by a fixed number of kernel functions and solved the optimization problem by the steepest descent method. Wang et al. [6] proposed selecting basis vectors by the orthogonal forward selection.

Xiong et al. [9] proposed the empirical feature space whose dimension is the number of training data at most and whose kernel function values are equivalent to those for the feature space for the training data pairs and proposed that kernel-based methods can be reformulated in the finite empirical feature space without loosing any information. Based on the idea of the empirical feature space, Abe [10] proposed sparse least squares support vector machines (LS SVMs) reducing the dimension of the empirical feature space by the Cholesky factorization and training the LS SVM in the primal form in the empirical feature space. The support vectors, which correspond to all the training data, are reduced to the training data that are selected by the Cholesky factorization.

In this paper we extend the method for realizing sparse LS SVMs [10] to that for sparse SVMs. Namely, we select the independent training data by the Cholesky factorization of the kernel matrix. The selected training data become support vectors and by loosening the threshold to select linearly independent data, we can reduce the dimension of the empirical feature space, namely, the number of support vectors. Since the empirical feature space can be handled explicitly, we can train the SVM in the primal form. But since it is more efficient in the dual form we train the SVM in the dual form.

In Section II, we summarize the characteristic of the empirical feature space based on [10]. In Section III, we formulate sparse SVMs. In Section IV, we solve dual problems, and derive the decision functions in the feature space and the empirical feature space. In section V, we show the validity of the proposed method by computer experiments.

II. Empirical Feature Space

In this section, we summarize the results for the empirical feature space based on [10].

Let the kernel be $H(\mathbf{x}, \mathbf{x}') = \mathbf{g}^T(\mathbf{x})\,\mathbf{g}(\mathbf{x}')$, where $\mathbf{g}(\mathbf{x})$ is the mapping function that maps m-dimensional vector $\mathbf{x}$ into the l-dimensional space. For the M m-dimensional data $\mathbf{x}_i$, the $M \times M$ kernel matrix $H = \{H_{ij}\}$ $(i, j = 1,..., M)$, where $H_{ij} = H(\mathbf{x}_i, \mathbf{x}_j)$ is symmetric and positive semi-definite. Let the rank of H be N $(\leq M)$. Then H is expressed by

$$H = P \Lambda P^T, \tag{1}$$

where Λ is a diagonal matrix containing only the N positive eigenvalues of H, and P consists of the eigenvectors corresponding to the positive eigenvalues. Then $P^T P = I_{N \times N}$ but $P P^T \neq I_{M \times M}$.

Now we define the mapping function that maps the m-dimensional vector $\mathbf{x}$ into the N-dimensional space called empirical feature space [9]:

$$\mathbf{h}(\mathbf{x}) = \Lambda^{-\frac{1}{2}} P^T (H(\mathbf{x}_1, \mathbf{x}), ..., H(\mathbf{x}_M, \mathbf{x}))^T. \tag{2}$$

We define the kernel associated with the empirical feature space by

$$H_{\mathrm{e}}(\mathbf{x}, \mathbf{x}') = \mathbf{h}^T(\mathbf{x})\mathbf{h}(\mathbf{x}'). \tag{3}$$

The remarkable fact is that the kernel for the empirical feature space is equivalent to the kernel for the feature space if they are evaluated using the training data as follows [9]:

$$H_{\mathrm{e}}(\mathbf{x}_i, \mathbf{x}_j) = H(\mathbf{x}_i, \mathbf{x}_j) \quad \text{for } i, j = 1, ..., M. \tag{4}$$

Kazuki Iwamura is a graduate student of Electrical Engineering, Kobe University, Japan (email: 073t203t@stu.kobe-u.ac.jp). Shigeo Abe is a professor of Graduate School of Engineering, Kobe University, Japan (email: abe@kobe-u.ac.jp).

978-1-4244-1821-3/08/$25.00©2008 IEEE

Now we prove (4). From (2)

$$\begin{aligned} H_{\mathrm{e}}(\mathbf{x}_i, \mathbf{x}_j) &= \mathbf{h}^T(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_j) \\ &= (H(\mathbf{x}_1, \mathbf{x}_i), ..., H(\mathbf{x}_M, \mathbf{x}_i)) P \Lambda^{-1} P^T \\ &\quad \times (H(\mathbf{x}_1, \mathbf{x}_j), ..., H(\mathbf{x}_M, \mathbf{x}_j))^T. \qquad (5) \end{aligned}$$

And from (1)

$$(H(\mathbf{x}_1, \mathbf{x}_i), ..., H(\mathbf{x}_M, \mathbf{x}_i))^T = P \Lambda \mathbf{q}_i, \qquad (6)$$

where $\mathbf{q}_i$ is the ith column vector of P^T. Substituting (6) into (5), we obtain

$$\begin{aligned} H_{\mathrm{e}}(\mathbf{x}_i, \mathbf{x}_j) &= \mathbf{q}_i^T \Lambda P^T P \Lambda^{-1} P^T P \Lambda \mathbf{q}_j \\ &= \mathbf{q}_i^T \Lambda \mathbf{q}_j \\ &= H(\mathbf{x}_i, \mathbf{x}_j). \qquad (7) \end{aligned}$$

The relation given by (4) is important in that a problem expressed using kernels can be interpreted, without introducing any approximation, as the problem defined in the associated empirical feature space. The dimension of the feature space is sometimes very high but that of the empirical feature space is the number of training data at most for pattern classification. Thus, instead of analyzing the feature space, we only need to analyze the empirical feature space associated with the feature space.

III. Training in the Reduced Empirical Feature Space

In training SVMs in the empirical feature space, first we need to carry out the eigenvalue decomposition of the kernel matrix and then to transform the input variables into variables in the empirical feature space by (2). But this is time consuming. Thus, instead of using (2), we select linearly independent training data that span the empirical feature space. Namely, instead of (2), we use the following equation:

$$\mathbf{h}(\mathbf{x}) = (H(\mathbf{x}_{i_1}, \mathbf{x}), ..., H(\mathbf{x}_{i_N}, \mathbf{x}))^T, \qquad (8)$$

where $i_j \in \{1, \ldots, M\}$ and $j = \{1, \ldots, N\}$. By this formulation, since the solution is expressed by the linear combination of $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_N}$, they 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. Although the empirical feature space spanned by (8) is equivalent to that spanned by (2), both coordinates are different. Thus, the solutions trained using (8) are different from those using (2) because SVMs are not invariant for the linear transformation of input variables [1]. But, this is not a problem if we select kernels and the margin parameter properly such as by cross-validation.

One way to select linearly independent variables is to use the Cholesky factorization [1]. Let H be positive definite. Then H is decomposed by the Cholesky factorization into

$$H = L L^T, \qquad (9)$$

where L is the regular lower triangular matrix and each element L_{ij} is given by

$$\begin{aligned} L_{op} &= \frac{H_{op} - \sum_{n=1}^{p-1} L_{pn} L_{on}}{L_{pp}} \\ &\quad \text{for } o = 1, ..., M, \;\; p = 1, ..., o-1, \qquad (10) \\ L_{aa} &= \sqrt{H_{aa} - \sum_{n=1}^{a-1} L_{an}^2} \quad \text{for } a = 1, ..., M. \qquad (11) \end{aligned}$$

Here, $H_{ij} = H(\mathbf{x}_i, \mathbf{x}_j)$. Then, during the Cholesky factorization, if the argument of the square root in (11) is smaller than the prescribed value η (>0):

$$H_{aa} - \sum_{n=1}^{a-1} L_{an}^2 \leq \eta, \qquad (12)$$

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

The above Cholesky factorization can be done incrementally [1], [12]. Namely, instead of calculating the full kernel matrix in advance, if (12) 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 data, not the number of training data.

To obtain the empirical feature space, we set a small value to η. But to increase sparsity of the solutions, we increase the value of η. The optimal value is determined by cross-validation. We call thus trained SVMs sparse SVMs.

If we use the linear kernels we do not need to select linearly independent variables. Instead of (8), we use

$$\mathbf{h}(\mathbf{x}) = \mathbf{x}. \qquad (13)$$

This is equivalent to using $\mathbf{e}_i$ $(i = 1, ..., m)$, where $\mathbf{e}_i$ are the basis vectors in the input space, in which the ith element is 1 and other elements 0. We call the SVM using (13) SVM with orthogonal support vectors (OSV), and the SVM with selected linearly independent training data using (8) SVM with non-orthogonal support vectors (NOSV).

IV. Sparse Support Vector Machines

In the following we discuss SVMs trained in the reduced empirical feature space. By increasing the value of η we obtain sparse SVMs.

Expressing the SVMs in the empirical feature space, we can explicitly treat variables in the empirical feature space. Thus we can train the SVM either in the primal or dual form in the reduced empirical feature space. In the following we consider training the SVM in the dual form because we can implement the sparse SVM by a small modification of the existing SVM software.

The optimal separating hyperplane in the feature space is determined by minimizing

$$Q(\mathbf{w}, \boldsymbol{\xi}, b) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{r}\sum_{i=1}^{M}\xi_i^r \tag{14}$$

subject to the constraints

$$y_i(\mathbf{w}^T\mathbf{g}(\mathbf{x}_i) + b) \geq 1 - \xi_i, \tag{15}$$

where $\mathbf{w}$ is the l-dimensional vector in the feature space, $\mathbf{g}$ is the mapping function that maps the input space into the feature space, b is the bias term, $\boldsymbol{\xi} = (\xi_1, ..., \xi_M)^T$ is a slack variable vector, C is the margin parameter that determines the trade-off between the maximization of the margin and minimization of the classification error, and $r = 1$ for L1 SVMs and $r = 2$ for L2 SVMs.

The optimal separating hyperplane in the empirical feature space is determined by minimizing

$$Q(\mathbf{v}, \boldsymbol{\xi}, b) = \frac{1}{2}\mathbf{v}^T\mathbf{v} + \frac{C}{r}\sum_{i=1}^{M}\xi_i^r \tag{16}$$

subject to the constraints

$$y_i(\mathbf{v}^T\mathbf{h}(\mathbf{x}_i) + b) \geq 1 - \xi_i, \tag{17}$$

where $\mathbf{v}$ is the N-dimensional vector in the empirical feature space, $\mathbf{h}$ is the mapping function that maps the input space into the empirical feature space, and b is the bias term.

Since the dimension of $\mathbf{v}$ is at most M the optimization problem given by (16) and (17) is solvable in its primal or dual form although the optimization problem given by (14) and (15) with the infinite dimension $\mathbf{g}(\mathbf{x})$ is solvable only in dual form.

In the following we derive the dual problem of (16) and (17). Introducing nonnegative Lagrange variables α_i and β_i, we obtain

$$\begin{aligned} Q(\mathbf{v}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \;\; = \;\; & \frac{1}{2}\|\mathbf{v}\|^2 + \frac{C}{r}\sum_{i=1}^{M}\xi_i^r \\ & -\sum_{i=1}^{M}\alpha_i(y_i(\mathbf{v}^T\mathbf{h}(\mathbf{x}_i) + b) - 1 + \xi_i) \\ & -\sum_{i=1}^{M}\beta_i\xi_i, \end{aligned} \tag{18}$$

where $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_M)^T$ and $\boldsymbol{\beta} = (\beta_1, ..., \beta_M)^T$.

Then we obtain the following dual problem for the L1 SVM: Maximize

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{M}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{M}\alpha_i\alpha_j y_i y_j \mathbf{h}^T(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_j) \tag{19}$$

subject to the constraints

$$\sum_{i=1}^{M} y_i\alpha_i = 0, \quad 0 \leq \alpha_i \leq C \quad \text{for} \;\; i = 1, ..., M, \tag{20}$$

and for the L2 SVM: Maximize

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{M}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{M}\left(\alpha_i\alpha_j y_i y_j \mathbf{h}^T(\mathbf{x}_i)h(\mathbf{x}_j) + \frac{\delta_{ij}}{C}\right) \tag{21}$$

subject to the constraints

$$\sum_{i=1}^{M} y_i\alpha_i = 0 \qquad \text{for} \;\; i = 1, ..., M, \tag{22}$$

where $\delta_{ij} = 0$ for $i \neq j$ and $\delta_{ij} = 1$ for $i = j$.

Likewise, we can derive the dual problem for (14) and (15) for the L1 SVM: Maximize

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{M}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{M}\alpha_i\alpha_j y_i y_j H(\mathbf{x}_i, \mathbf{x}_j) \tag{23}$$

subject to the constraints

$$\sum_{i=1}^{M} y_i\alpha_i = 0, \quad 0 \leq \alpha_i \leq C \quad \text{for} \;\; i = 1, ..., M, \tag{24}$$

and for the L2 SVM: Maximize

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{M}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{M}\left(\alpha_i\alpha_j y_i y_j H(\mathbf{x}_i, \mathbf{x}_j) + \frac{\delta_{ij}}{C}\right) \tag{25}$$

subject to the constraints

$$\sum_{i=1}^{M} y_i\alpha_i = 0 \quad \text{for} \;\; i = 1, ..., M. \tag{26}$$

The obtained decision function in the feature space is

$$D(\mathbf{x}) = \sum_{i=1}^{M}\alpha_i y_i H(\mathbf{x}_i, \mathbf{x}) + b. \tag{27}$$

This equation shows that only support vectors with $\alpha > 0$ are necessary to represent a solution for the SVM.

The decision function in the empirical feature space is

$$\begin{aligned} D(\mathbf{x}) \;\; & = \;\; \mathbf{v}^T\mathbf{h}(\mathbf{x}) + b \\ & = \;\; \mathbf{v}^T(H(\mathbf{x}_{i_1}, \mathbf{x}), ..., H(\mathbf{x}_{i_N}, \mathbf{x}))^T + b. \end{aligned} \tag{28}$$

In this formulation, $\mathbf{x}_i$ with $\alpha_i\,(> 0)$ for the solution of (19) and (20) or (21) and (22) do not constitute support vectors since they are not used in expressing the decision function given by (28). Rather we need only the selected linearly independent training data and they are support vectors in the empirical feature space. By setting a small value to η we obtain sparse SVMs. If the dimension of the reduced empirical feature space is considerably smaller than the number of support vectors in the feature space, faster training is possible.

The difference of the SVMs in the feature space and the empirical feature space is whether we use $H(\mathbf{x}, \mathbf{x}')$ or $\mathbf{h}^T(\mathbf{x})\,\mathbf{h}(\mathbf{x}')$. Thus by adding the program to select linearly independent data and by changing the calculation of $H(\mathbf{x}, \mathbf{x}')$ by that of $\mathbf{h}^T(\mathbf{x})\,\mathbf{h}(\mathbf{x}')$ in the training program, we obtain the program for training SVMs in the empirical feature space.

V. Experimental Results

A. Evaluation Conditions

We compared the generalization ability of sparse SVMs and regular SVMs using 13 two-class data sets [13], [14] shown in Table I, which lists the number of inputs, training data, test data, and the data sets. Each problem has 100 or 20 training data sets and their corresponding test data sets. We measured the computation time using a workstation (2.6 GHz, 2 GB memory, Linux operating system).

TABLE I

Benchmark data sets for two-class problems

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

In all studies, we normalized the input ranges into [0, 1] and used linear and RBF kernels. We determined the values of C for linear kernels, C and γ for RBF kernels, and η and C for sparse SVMs by fivefold cross-validation; the value of C was selected from among $\{$1; 10; 50; 1,000; 2,000; 3,000; 5,000; 8,000; 10,000; 50,000; 100,000$\}$, the value of γ from among $\{0.1, 0.5, 1, 5, 10, 15\}$, and the value of η from among $\{0.5, 0.1, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$.

For linear kernels, we determined the optimal value of C for SVMs and sparse SVMs by cross-validation for the first five training data sets and selected the median of the optimal values. And we set the $\eta = 10^{-9}$ for sparse SVMs.

For RBF kernels, we determined the optimal values of γ and C for dual SVMs, and η and C for sparse SVMs by cross-validation. In the following we explain the procedure for determining γ and C for dual SVMs.

For RBF kernels for a value of γ we performed cross-validation of the first five training data sets changing C, and selected the optimal value of C for each value of γ and each training data set. If the recognition rate of the validation sets took the maximum value for different values of C, we took the smallest value as the optimal value. We chose the value of γ as the optimal value that shows the maximum average recognition rates for the first five training data sets. For the selected value of γ, we selected the median of the optimal values of C associated with the five training data sets. Then, for the optimal values of γ and C, we trained the SVM for 100 or 20 training data sets and calculated the average recognition rates and the standard deviations for the test data sets.

In the similar way we determined the optimal value of η and C for sparse SVMs, setting the optimal value of γ determined for SVMs.

B. Results for L1 SVMs

Table II lists the parameters for L1 SVMs with linear kernels obtained by the preceding procedure. As the theory tells us, the values of C are the same for the SVM and the OSV. The values of C for the NOSV differed very much from those of the SVM (OSV) in some cases.

TABLE II

Determined margin parameter values of C for L1 SVMs with linear kernels. The parameters were determined by fivefold cross-validation

Data	L1 SVM	OSV	NOSV
Banana	10,000	10,000	10
B. cancer	1,000	1,000	10
Diabetes	5,000	5,000	5,000
German	8,000	8,000	1
Heart	10	10	2,000
Image	5,000	5,000	50,000
Ringnorm	5,000	5,000	5,000
F. solar	1	1	1
Splice	1	1	10
Thyroid	100	100	100,000
Titanic	1	1	10
Twonorm	1	1	10
Waveform	1	1	10

Table III lists the parameters for RBF kernels obtained by the preceding procedure. In several cases, the values of C for the sparse SVM are larger than those for the L1 SVM.

TABLE III

Parameter setting for L1 SVMs with RBF kernels. The parameters were determined by fivefold cross-validation

Data	L1 SVM		Sparse	
	C	γ	C	η
Banana	100	15	5,000	0.1
B. cancer	500	0.1	100,000	10^{-5}
Diabetes	3,000	0.1	100,000	10^{-3}
German	50	0.1	2,000	10^{-3}
Heart	50	0.1	500	10^{-3}
Image	500	15	10,000	10^{-3}
Ringnorm	1	15	1	0.1
F. solar	10	0.5	50	0.1
Splice	100,000	10	10	0.1
Thyroid	100	15	100	10^{-2}
Titanic	50	0.5	10	10^{-3}
Twonorm	1	0.5	100	10^{-3}
Waveform	1	10	1	0.1

Table IV shows the average recognition rates and standard deviations for the linear kernels. According to the theory, the recognition rates and standard deviations are the same for SVM and OSV. Except for the banana problem, the recognition rates and the standard deviations are almost the same for the SVM (OSV) and the NSOV. For the banana problem, the average and the standard deviation of the NOSV are worse. This is due to mal-selection of the margin parameter; for some test data sets the recognition rates were 0. If we use the C=10,000 instead of $C = 10$, the recognition rate and the standard deviation were 52.7±5.0, which are comparable to those of the SVM (OSV).

TABLE IV

COMPARISON OF THE AVERAGE RECOGNITION RATES AND THE STANDARD DEVIATIONS FOR L1 SVMS WITH LINEAR KERNELS

Data	SVM	OSV	NOSV
Banana	53.2±4.9	53.2±4.9	52.9±8.4
B. cancer	70.2±5.4	70.2±5.4	70.9±5.5
Diabetes	75.9±1.8	75.9±1.8	75.7±2.0
German	75.8±2.2	75.8±2.2	76.1±2.2
Heart	82.6±3.2	82.6±3.2	82.7±3.0
Image	84.5±1.0	84.5±1.0	83.8±1.3
Ringnorm	73.6±0.92	73.6±0.92	73.7±0.88
F. solar	67.7±1.8	67.7±1.8	67.7±1.8
Splice	83.6±0.66	83.6±0.66	83.0±0.63
Thyroid	90.5±2.7	90.5±2.7	87.1±4.1
Titanic	77.4±0.44	77.4±0.44	77.2±1.3
Twonorm	97.3±0.20	97.3±0.20	97.2±0.23
Waveform	86.9±0.57	86.9±0.57	85.7±0.61

Table V shows the average recognition rates and standard deviations for the RBF kernels. If an average or a standard deviation of a classification problem is statistically better than the other with the significance level of 0.05, it is shown in boldface. For five data sets, the average recognition rate and/or the standard deviation by the sparse SVM is worse, but the difference is not so large. For the ringnorm problem the results by the sparse SVM was better.

TABLE V

COMPARISON OF THE AVERAGE RECOGNITION RATES AND THE STANDARD DEVIATIONS FOR L1 SVMS WITH RBF KERNELS

Data	L1 SVM	Sparse
Banana	**89.3**±0.52	89.1±0.60
B. cancer	72.4±4.7	72.0±5.3
Diabetes	**76.3**±1.8	75.8±1.7
German	76.2±2.3	76.0±2.3
Heart	83.7±3.4	83.1±3.4
Image	**97.3±0.41**	96.1±0.74
Ringnorm	97.8±0.30	**98.1±0.19**
F. solar	67.6±1.7	67.6±1.7
Splice	89.2±0.71	88.8±0.79
Thyroid	96.1±2.1	96.1±2.1
Titanic	77.5±0.55	77.4±0.47
Twonorm	**97.6±0.14**	97.4±0.19
Waveform	**90.0±0.44**	89.4±1.0

Table VI lists the number of support vectors for linear kernels. The numbers of support vectors for OSV and NOSV are the same with the number of inputs. Compared with those of the SVM, the numbers of support vectors of the OSV and the NOSV are extremely small.

Table VII lists the number of support vectors for L1 SVMs with RBF kernels. Except for the image, ringnorm, splice, and thyroid problems we could reduce the number of support vectors. And the numbers of support vectors of sparse SVMs for those problems, except the waveform problem, are from 5 to 52% of the numbers of support vectors of SVMs.

We measured the computation time of training and testing the 100 or 20 sets in a classification problem. Then we calculated the average computation time for a training data set and its associated test data set. Table VIII lists the results for linear kernels. The calculation of the OSV is in most cases shorter than that of the SVM.

TABLE VI

COMPARISON OF SUPPORT VECTORS FOR L1 SVMS WITH LINEAR KERNELS

Data	L1 SVM	OSV	NOSV
Banana	397±7.8	2	2
B. cancer	174±34	9	9
Diabetes	259±12	8	8
German	416±17	20	20
Heart	62.0±6.2	13	13
Image	609±26	15	15
Ringnorm	235±18	20	20
F. solar	544±14	9	9
Splice	366±20	60	60
Thyroid	40.9±4.4	5	5
Titanic	143±6.1	3	3
Twonorm	74.6±5.0	20	20
Waveform	145±9.7	21	21

TABLE VII

COMPARISON OF SUPPORT VECTORS FOR L1 SVMS WITH RBF KERNELS

Data	L1 SVM	Sparse
Banana	101±10	**17.3±1.2**
B. cancer	124±11	**64.4±1.9**
Diabetes	255.0±12	**9.9±0.74**
German	398±6.1	**35.1±1.5**
Heart	73.9±5.6	**25.3±1.2**
Image	**151±8.0**	385±9.7
Ringnorm	**130±5.5**	214±9.3
F. solar	530±14	**8.3±0.62**
Splice	**741±14**	968±5.8
Thyroid	**14.1±2.0**	42.8±2.3
Titanic	139±10	**8.5±1.0**
Twonorm	255±8.0	**67.7±5.0**
Waveform	153±8.9	**132±6.4**

Table IX lists the results for the L1 SVM with RBF kernels. Especially training the german and image problems is slower. These problems have large numbers of training data and thus they took long time in the Cholesky factorization.

Figure 1 shows the recognition rate of the test data and the number of support vectors with RBF kernels against the value of η for one data set of the banana problem. The recognition rate was constant for the value of η equal to or smaller than 0.1 but the number of support vectors increased linearly as the value of η was decreased. Thus, sparse L1 SVM was obtained for the banana problem.

C. Results for L2 SVMs

Since the results for linear kernels are similar to those of L1 SVMs we show the results only for RBF kernels. Table X lists the parameter values for L2 SVMs with RBF kernels. Parameter values are almost the same with those of L1 SVMs and sparse L1 SVMs.

Table XI shows the average recognition rates and standard deviations for the RBF kernels. The result in boldface shows that it is statistically better than the other with the significance level of 0.05. From the table for four problems the sparse L2 SVM is statistically better and for three problems

TABLE VIII

COMPARISON OF COMPUTATION TIME IN SECONDS FOR L1 SVMS WITH LINEAR KERNELS

Data	L1 SVM	OSV	NOSV
Banana	3.5	3.5	**2.1**
B. cancer	0.4	0.4	**0.3**
Diabetes	2.1	**2.0**	3.4
German	13	13	**11**
Heart	0.1	**0.04**	0.1
Image	66	**62**	114
Ringnorm	**2.0**	2.1	2.8
F. solar	4.2	**4.1**	9.4
Splice	5.0	**4.2**	17
Thyroid	0.06	**0.03**	0.13
Titanic	0.1	**0.07**	0.1
Twonorm	**0.2**	**0.2**	0.94
Waveform	**0.3**	**0.3**	1.0

TABLE IX

COMPARISON OF COMPUTATION TIME IN SECONDS FOR THE L1 SVM WITH RBF KERNELS

Data	L1 SVM	Sparse
Banana	**0.3**	1.1
B. cancer	**0.4**	0.6
Diabetes	**1.9**	4.6
German	**6.1**	17
Heart	**0.08**	0.10
Image	**1.0**	23
Ringnorm	**0.7**	1.2
F. solar	**5.3**	10
Splice	27	**20**
Thyroid	0.04	**0.03**
Titanic	0.2	**0.1**
Twonorm	1.5	**1.1**
Waveform	**0.6**	1.6

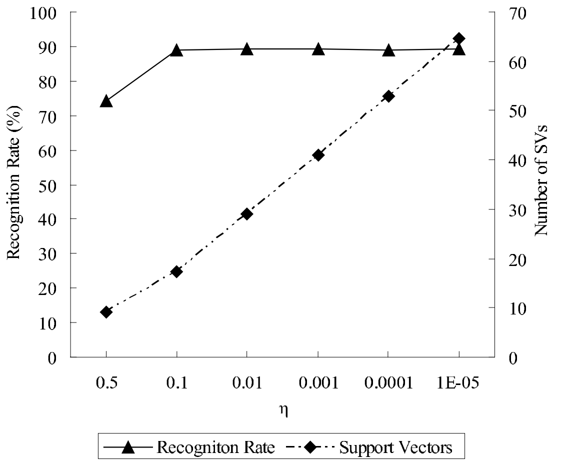

Fig. 1. Recognition rates and support vectors for L1 SVMs with RBF kernels against the value of η for the banana problem

TABLE X

PARAMETER SETTING FOR L2 SVMS WITH RBF KERNELS. THE PARAMETERS WERE DETERMINED BY FIVEFOLD CROSS-VALIDATION

Data	L2 SVM		Sparse	
	C	γ	C	η
Banana	3,000	10	8000	10^{-2}
B. cancer	500	0.1	100,000	10^{-5}
Diabetes	3,000	0.1	10,000	10^{-4}
German	50	0.1	100,000	10^{-3}
Heart	50	0.1	2000	10^{-4}
Image	500	15	10,000	10^{-3}
Ringnorm	1000	15	1	10^{-2}
F. solar	1	0.5	1	10^{-2}
Splice	1,000	10	100,000	10^{-5}
Thyroid	100	15	500	0.1
Titanic	10	0.5	50	10^{-3}
Twonorm	1	0.5	500	10^{-3}
Waveform	1	15	1	0.1

the L2 SVM is better. Thus, the two methods are comparable in the generalization ability.

TABLE XI

COMPARISON OF THE AVERAGE RECOGNITION RATES AND THE STANDARD DEVIATIONS FOR L2 SVMS WITH RBF KERNELS

Data	L2 SVM	Sparse
Banana	89.4±0.51	89.3±0.49
B. cancer	74.7±4.1	74.5±4.2
Diabetes	76.5±1.9	76.0±1.8
German	74.6±2.3	74.9±2.1
Heart	83.6±2.9	83.2±3.1
Image	96.4±0.72	**97.4±0.35**
Ringnorm	**98.2±0.20**	97.3±0.36
F. solar	64.3±3.2	**66.3±2.3**
Splice	85.9±1.3	**88.8±0.80**
Thyroid	96.2±1.9	95.7±2.2
Titanic	76.8±**1.4**	76.9±1.9
Twonorm	97.2±0.66	97.2±**0.25**
Waveform	**90.2**±0.40	89.9±0.39

Table XII lists the number of support vectors for L2 SVMs with RBF kernels. Sparsity was obtained for 8 problems out of 13. Comparing the results for the L1 SVM shown in Table VII, the numbers of support vectors for both methods are the same for the b. cancer, german, image, titanic, and twonorm problems. This is because the values of γ and η are the same as seen from Tables III and X. Thus the reduced empirical feature spaces selected for L1 and L2 SVMs are the same.

Table XIII lists the computation time for L2 SVMs with RBF kernels. For the banana, ringnorm, titanic, twonorm and waveform problems, training of sparse L2 SVMs was faster than L2 SVMs.

Figure 2 shows the recognition rate and the number of support vectors with RBF kernels against the value of η for one data set of the waveform problem. The recognition rate did not change very much for the change of η. But the number of support vectors saturated to the number of training data for the value of η smaller than 0.01. Thus, sparsity was not obtained for the sparse L2 SVM.

TABLE XII

COMPARISON OF SUPPORT VECTORS FOR L2 SVMS WITH RBF KERNELS

Data	L2 SVM	Sparse
Banana	168±21	**22.3±1.3**
B. cancer	176±6.4	**64.4±1.9**
Diabetes	393±9.1	**23.1±1.7**
German	575±12	**35.1±1.5**
Heart	123±7.8	**67.6±2.0**
Image	**221±15**	385±9.7
Ringnorm	**249±7.5**	373±5.9
F. solar	322±96	**21.8±1.4**
Splice	**376±125**	977±5.2
Thyroid	**18.0±4.5**	23.5±1.6
Titanic	149±3.5	**8.5±1.0**
Twonorm	259±45	**67.7±5.0**
Waveform	**208±9.9**	249±10

TABLE XIII

COMPARISON OF COMPUTATION TIME IN SECONDS FOR L2 SVMS WITH RBF KERNELS

Data	L2 SVM	Sparse
Banana	22	**4.1**
B. cancer	**1.4**	1.5
Diabetes	**15**	21
German	**53**	65
Heart	**0.6**	0.7
Image	**5.5**	31
Ringnorm	5.9	**2.0**
F. solar	**5.7**	68
Splice	**19**	53
Thyroid	**0.05**	0.13
Titanic	0.8	**0.7**
Twonorm	4.8	**1.4**
Waveform	3.3	**2.8**

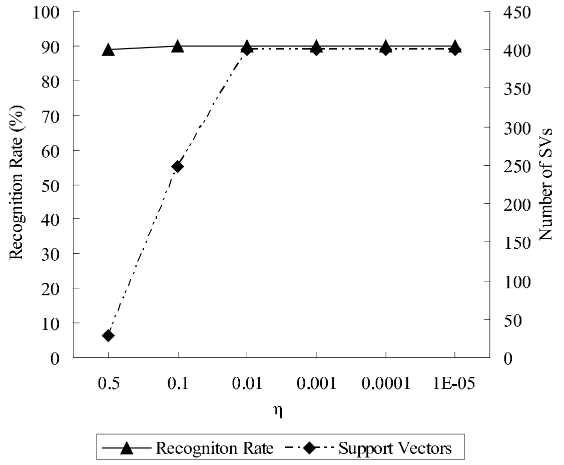

Fig. 2. Recognition rates and support vectors with L2 SVMs with RBF kernels against the value of η for the waveform problem

VI. CONCLUSIONS

In this paper we proposed sparse L1 and L2 SVMs restricting the dimension of the empirical feature space by the Cholesky factorization. Namely, in the reduced empirical feature space we formulated the sparse SVM in the dual form. According to the computer experiments, for almost all data sets tested the sparse SVMs could realize sparsity while realizing generalization ability comparable with that of the SVMs. And there was not much difference between sparse L1 SVMs and sparse L2 SVMs. The sparsity was realized most for linear kernels.

REFERENCES

[1] S. Abe, *Support Vector Machines for Pattern Classification*, Springer, London, 2005.

[2] C. J. C. Burges, "*Simplified support vector decision rules*," In L. Saitta, editor, *Machine Learning Proceedings of 13th International Conference*, Morgan Kaufmann, San Francisco, pp. 71–77, 1996.

[3] F. Shai and F. Katya, "Efficient SVM training using low-rank kernel representation," *The Journal of Machine Learning Research*, Vol. 2, pp. 243–264, 2002.

[4] S. S. Keerthi, O. Chaoelle, and D. DeCoste, "Building support machines with reduced classifier complexity," *The Journal of Machine Learning Research*, Vol. 7, pp. 1493–1515, 2006.

[5] M. Wu, B. Schölkopf, and G Bakir, "A direct method for building sparse kernel learning algorithms," *Journal of Machine Learning Research*, Vol. 7, pp. 603–624, 2006.

[6] X. X. Wang, S. Chen, D. Lowe, and C. J. Harris, "Sparse support vector regression based on orthogonal forward selection for the generalized kernel model," *Neurocomputing 70* Vol. 1, No. 3, pp. 462–474, 2006.

[7] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, Vol. 1, pp. 211–244, 2001.

[8] S. Chen, X. Hong, C. J. Harris, and P. M. Sharky, "Sparse modelling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol. 34, No. 2, pp. 898–911, 2004.

[9] H. Xiong, M. N. S. Swamy, and M. O. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Trans. Neural Networks*, Vol. 16, No. 2, pp. 460–474, 2005.

[10] S. Abe, "Sparse least squares support vector training in the reduced empirical feature space," *Pattern Analysis and Applications*, Springer-Verlag, Vol. 10, pp. 203–214, 2007.

[11] V. N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[12] K. Kaieda and S. Abe, "KPCA-based training of a kernel fuzzy classifier with ellipsoidal regions," *International Journal of Approximate Reasoning*, Vol. 37, No. 3, pp. 145–253, 2004.

[13] G. Rätsch, T. Onoda, and K.-R. Muller, "Soft margins for AdaBoost," *Machine Learning*, Vol. 42, No. 3, pp. 287–320, 2001.

[14] K.-R. Muller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks*, Vol. 2, No. 3, pp. 287–320, 2001.