ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Sparse Support Vector Machines by Kernel Discriminant Analysis

Kazuki Iwamura and Shigeo Abe

Kobe University - Graduate School of Engineering
Kobe, Japan

Abstract. We discuss sparse support vector machines (SVMs) by selecting the linearly independent data in the empirical feature space. First we select training data that maximally separate two classes in the empirical feature space. As a selection criterion we use linear discriminant analysis in the empirical feature space and select training data by forward selection. Then the SVM is trained in the empirical feature space spanned by the selected training data. We evaluate our method by computer experiments and show that our method can realize sparse SVMs with comparable generalization performance with that of regular SVMs.

1 Introduction

Support vector machines (SVMs) are known to realize sparse solutions in that only support vector are necessary to represent solutions. But for difficult classification problems, many training data become support vectors and sparsity of solutions decreases. Thus there are many approaches to improve sparsity of solutions.

Wang et al. [1] proposed selecting basis vectors by the orthogonal forward selection. There are some approaches to realize sparse kernel expansion by forward selection of basis vectors [2, 3, 4]. Based on the concept of the empirical feature space [5], which is closely related to kernel expansion, in [6, 7] sparse (LS) SVMs are developed restricting the dimension of the empirical feature space by the Cholesky factorization. And in [8] a sparse LS SVM is realized by selecting data that separate two-classes in the empirical feature space, in which separability is evaluated by linear discriminant analysis (LDA). This idea of selecting data by LDA is essentially the same with that used in [9].

In this paper based on [8], we realize sparse SVMs selecting the maximally separating data by LDA. Namely, by forward selection we select training data that maximally separate two classes in the empirical feature space. If the matrix associated with LDA is singular, the newly added data sample does not contribute to the class separation. Thus, we 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. Then we train SVMs in the empirical feature space spanned by the data selected by forward selection. In this formulation, support vectors in the empirical feature space are expressed by a linear combination of mapped, selected training data. Therefore, only the selected data are necessary to form the solution of the SVM.

ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

In Section 2, we discuss sparse 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 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. Assume that we have $N\,(\leq M)$ training data $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_N}$ that are linearly independent in the empirical feature space, where $\mathbf{x}_{i_j} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ and $j = 1, \ldots, N$. We map the input space into the empirical feature space by

$$\mathbf{h}(\mathbf{x}) = (H(\mathbf{x}_{i_1}, \mathbf{x}), \ldots, H(\mathbf{x}_{i_N}, \mathbf{x}))^T, \tag{1}$$

where $H(\mathbf{x}, \mathbf{x}')$ is a kernel.

Since the empirical feature space is finite, we can train the SVM either in the primal or dual form but since we can use the same training method as that of the regular SVM, we train the L1 SVM in the dual form as follows:

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

$$\text{subject to} \qquad \sum_{i=1}^{M} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C \quad \text{for} \quad i = 1, ..., M. \tag{3}$$

The decision function in the empirical feature space is

$$D(\mathbf{x}) = \mathbf{v}^T \mathbf{h}(\mathbf{x}) + b = \mathbf{v}^T (H(\mathbf{x}_{i_1}, \mathbf{x}), ..., H(\mathbf{x}_{i_N}, \mathbf{x}))^T + b, \tag{4}$$

where $\mathbf{v}$ is a constant vector. The difference of the SVMs in the feature space and the empirical feature space is whether we use $H(\mathbf{x}, \mathbf{x}')$ or $\mathbf{h}^T(\mathbf{x})\,\mathbf{h}(\mathbf{x}')$.

In this formulation, $\mathbf{x}_i$ with $\alpha_i\,(> 0)$ for the solution of (2) and (3) do not constitute support vectors since they are not used in expressing the decision function given by (4). Rather we need only the selected linearly independent training data and they are support vectors in the empirical feature space. By a small number of linearly independent data we obtain sparse SVMs. If the dimension of the empirical feature space is considerably smaller than the number of support vectors in the feature space, faster training is possible.

3 Selecting Independent Data by Forward Selection

3.1 Linear Discriminant Analysis in the Empirical Feature Space

In this section we discuss LDA that is used for forward selection. 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.

In LDA we maximize the following objective function:

$$J(\mathbf{w}) = \frac{d^2}{s^2} = \frac{\mathbf{w}^T Q_B \mathbf{w}}{\mathbf{w}^T Q_W \mathbf{w}}, \tag{5}$$

where d^2 is the square difference of the centers of the projected data given by

$$d^2 = \mathbf{w}^T Q_B \mathbf{w} = \mathbf{w}^T (\mathbf{c}_1 - \mathbf{c}_2) (\mathbf{c}_1 - \mathbf{c}_2)^T \mathbf{w}. \tag{6}$$

Here, Q_B is the between-class scatter matrix and $\mathbf{c}_i$ are the centers of class i data:

$$\mathbf{c}_i = \frac{1}{M_i} \sum_{j=1}^{M_i} \mathbf{h}(\mathbf{x}_j^i) \quad \text{for} \quad i = 1, 2. \tag{7}$$

And $s^2 = \mathbf{w}^T Q_W \mathbf{w}$ is the variance of the projected data and Q_W is the within-class scatter matrix:

$$Q_W = \frac{1}{M} \sum_{j=1}^{M} \mathbf{h}(\mathbf{x}_j) \mathbf{h}(\mathbf{x}_j)^T - \mathbf{c}\, \mathbf{c}^T, \quad \mathbf{c} = \frac{1}{M} \sum_{j=1}^{M} \mathbf{h}(\mathbf{x}_j) = \frac{M_1 \mathbf{c}_1 + M_2 \mathbf{c}_2}{M_1 + M_2}. \tag{8}$$

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

$$\mathbf{w}_{\text{opt}} = Q_W^{-1} (\mathbf{c}_1 - \mathbf{c}_2). \tag{9}$$

We substitute (9) into (5) and obtain

$$J(\mathbf{w}_{\text{opt}}) = (\mathbf{c}_1 - \mathbf{c}_2)^T \mathbf{w}_{\text{opt}}. \tag{10}$$

3.2 Forward Selection

Starting from an empty set we add one datum at a time that maximizes (5) 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_j^k denote that $\mathbf{x}_j$ ($j \in T^k$) 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:

$$\mathbf{h}^{k,j}(\mathbf{x}) = (H(\mathbf{x}_{i_1}, \mathbf{x}), \ldots, H(\mathbf{x}_{i_k}, \mathbf{x}), H(\mathbf{x}_j, \mathbf{x}))^T, \tag{11}$$

ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

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

$$j_{\rm opt} = \arg_j J_{\rm opt}^{k,j} \qquad \text{for} \quad j \in T^k \tag{12}$$

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

$$\left(J_{\rm opt}^{k,j_{\rm opt}} - J_{\rm opt}^{k}\right) / J_{\rm opt}^{k,j_{\rm opt}} \geq \eta, \tag{13}$$

where η 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.

If the addition of a data sample results in the singularity of $Q_{\mathbf{w}}^{k,j}$, where $Q_{\mathbf{w}}^{k,j}$ is the within-class scatter matrix evaluated using the data with $S^{k,j}$ indices, the data sample does not give useful information in addition to the already selected data. If $\mathbf{x}_j$ causes the singularity of $Q_{\mathbf{w}}^{k,j}$, later addition will always cause singularity of the matrix. Namely, we can delete j from T^k permanently.

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 (12) and set $S^1 = \{j_{\rm opt}\}$, $T^1 = T^0 - \{j_{\rm opt}\}$, and $k = 1$.

2. If for some $j \in T^k$, $Q_{\mathbf{w}}^{k,j}$ is singular, permanently delete j from T^k and calculate $j_{\rm opt}$ given by (12). If (13) 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.

If we keep the Cholesky factorization of $Q_{\mathbf{w}}^{k}$, the Cholesky factorization of $Q_{\mathbf{w}}^{k,j}$ can be 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. Or we can use the matrix inversion lemma. According to numerical experiments there was not much difference in computation time between the two methods.

We call thus trained SVM sparse SVM by forward selection, SSVM (L) for short and the sparse SVM by Cholesky factorization in [7] SSVM (C).

4 Experimental Results

We compared the generalization ability of SSVMs (L), SSVM (C), and regular SVMs using two-class problems [10].

In all studies, we normalized the input ranges into [0, 1] and used RBF kernels. We determined the values of C and γ for RBF kernels, and η and C for sparse SVMs by fivefold cross-validation using the first five training data sets; the value of C was selected from among {1, 10, 50, 100, 500, 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 {0.001, 0.0001} for SSVM (L) and from among {0.5, 0.1, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}} for SSVM (C). We determined the optimal values of γ and C for SVMs and using the same value of γ we determined the values of η and C for sparse SVMs by cross-validation. Table 1 shows the determined parameter values.

Table 1: Parameter values

Data	L1 SVM		SSVM (C)		SSVM (L)	
	C	γ	C	η	C	η
Banana	100	15	5,000	0.1	8,000	10^{-3}
B. cancer	500	0.1	100,000	10^{-5}	1,000	10^{-4}
Diabetes	3,000	0.1	100,000	10^{-3}	10,000	10^{-4}
F. solar	10	0.5	50	0.1	10	10^{-3}
German	50	0.1	2,000	10^{-3}	3,000	10^{-3}
Heart	50	0.1	500	10^{-3}	3,000	10^{-4}
Image	500	15	10,000	10^{-3}	100,000	10^{-3}
Ringnorm	1	15	1	0.1	1	10^{-3}
Splice	100,000	10	10	0.1	1	10^{-3}
Thyroid	100	15	100	10^{-2}	500	10^{-3}
Titanic	50	0.5	10	10^{-3}	10	10^{-4}
Twonorm	1	0.5	100	10^{-3}	10	10^{-3}
Waveform	1	10	1	0.1	1	10^{-3}

Table 2 shows the recognition rates and their standard deviations of the test data sets. The three methods show the comparable recognition rates, although in some cases, the proposed method shows slightly inferior results.

Table 2: Recognition rates of the test data

Data	L1 SVM	SSVM (C)	SSVM (L)
Banana	89.3±0.52	89.1±0.60	89.1±0.60
B. cancer	72.4±4.7	72.0±5.3	71.3±4.5
Diabetes	76.3±1.8	75.8±1.7	75.6±2.0
F. solar	67.6±1.7	67.6±1.7	67.4±1.7
German	76.2±2.3	76.0±2.3	76.2±2.3
Heart	83.7±3.4	83.1±3.4	83.4±3.4
Image	97.3±0.41	96.1±0.74	96.2±0.59
Ringnorm	97.8±0.30	98.1±0.19	98.2±0.22
Splice	89.2±0.71	88.8±0.79	84.5±0.70
Thyroid	96.1±2.1	96.1±2.1	95.7±1.9
Titanic	77.5±0.55	77.4±0.47	77.4±0.49
Twonorm	97.6±0.14	97.4±0.19	96.9±0.36
Waveform	90.0±0.44	89.4±1.0	89.5±0.43

Table 3 shows the number of support vectors and training time for the determined parameter values. We measured the time using a 2.6 GHz personal computer with 2 GB memory. Compared to SVMs, except for the thyroid problem the number of support vectors of the proposed method is drastically decreased and compared to SSVM (C), except for the two problems, the number of support vectors is smaller. But training time is sometimes much longer.

Table 3: The number of support vectors and training time (s)

Data	L1 SVM		SSVM (C)		SSVM (L)	
	SVs	Time	SVs	Time	SVs	Time
Banana	101±10	0.3	**17.3±1.2**	1.1	22.9±8.9	9.66
B. cancer	124±11	0.5	64.4±1.9	0.4	**10.9±1.4**	1.4
Diabetes	255±12	1.9	9.9±0.74	4.6	**7.0±0.66**	6.7
F. solar	530±14	27	**8.3±0.62**	20	12.1±3.1	32
German	398±6.1	6.1	35.1±1.5	17	**13.1±2.2**	49.1
Heart	73.9±5.6	0.08	25.3±1.2	0.10	**13.6±1.4**	1.09
Image	151±8.0	1.0	385±9.7	23	**66.2±7.7**	1515
Ringnorm	130±5.5	2.0	214±9.3	2.1	**32.2±24**	52
Splice	741±14	27	968±5.8	20	**241±9.0**	14830
Thyroid	**14.1±2.0**	0.04	42.8±2.3	0.03	27.1±5.0	1.42
Titanic	139±10	0.2	8.5±1.0	0.1	**6.4±0.74**	0.4
Twonorm	255±8.0	1.5	67.7±5.0	1.1	**7.6±1.0**	9.0
Waveform	153±8.9	0.6	132±6.4	1.6	**112±24**	356

5 Conclusions

In this paper we proposed sparse SVMs by forward selection of independent data based on linear discriminant analysis in the empirical feature space. Namely, we select training data that maximally separate two classes in the empirical feature space. Then we train the SVM in the empirical feature space. For most of the two-class problems tested, sparsity of the solutions was increased drastically compared to regular SVMs and was better than the method using the Cholesky factorization.

References

[1] L. Wang, S. Sun, and K. Zhang, A fast approximate algorithm for training L1-SVMs in primal space, *Neurocomputing*, Vol. 70, pp. 1554–1560, 2007.

[2] L. Jiao, L. Bo, and L. Wang, Fast sparse approximation for least squares support vector machine, *IEEE Trans. Neural Networks*, Vol. 18, No. 3, pp. 685–697, 2007.

[3] A. J. Smola and P. L. Bartlett, Sparse greedy Gaussian process regression, *NIPS 13*, pp. 619-625, MIT Press, 2001.

[4] P. Vincent and Y. Bengio, Kernel matching pursuit, *Machine Learning*, Vol. 48, Nos. 1-3, pp. 165–187, 2002.

[5] 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.

[6] S. Abe, Sparse least squares support vector training in the reduced empirical feature space, *Pattern Analysis and Applications*, Vol. 10, No. 3, pp. 203–214, 2007.

[7] K. Iwamura and S. Abe, Sparse support vector machines trained in the reduced empirical feature space, *Proc. IJCNN 2008*, pp. 2399–2405, 2008.

[8] S. Abe, Sparse least squares support vector machines by forward selection based on linear discriminant analysis, *ANNPR 2008*, pp. 54–65, 2008.

[9] Y. Xu, D. Zhang, Z. Jin, M. Li, and J.-Y. Yang, A fast kernel-based nonlinear discriminant analysis for multi-class problems, *Pattern Recognition*, Vol. 39, pp. 1026–1033, 2006.

[10] Intelligent Data Analysis Group, http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm.