# Fast Training of Support Vector Machines by Extracting Boundary Data

Abe, Shigeo

Inoue, Takuya

(Citation)
Lecture Notes in Computer Science : Artificial Neural Networks - ICANN 2001,2130:308-313

(Issue Date)
2001-08

(Resource Type)
journal article

(Version)
Accepted Manuscript

(URL)
https://hdl.handle.net/20.500.14094/90000232

# Fast Training of Support Vector Machines by Extracting Boundary Data

Shigeo Abe and Takuya Inoue

Graduate School of Science and Technology, Kobe University
Rokkodai, Nada, Kobe Japan
{abe, t-inoue}@eedept.kobe-u.ac.jp
http://www.eedept.kobe-u.ac.jp/eedept_english.html

**Abstract.** Support vector machines have gotten wide acceptance for their high generalization ability for real world applications. But the major drawback is slow training for classification problems with a large number of training data. To overcome this problem, in this paper, we discuss extracting boundary data from the training data and train the support vector machine using only these data. Namely, for each training datum we calculate the Mahalanobis distances and extract those data that are misclassified by the Mahalanobis distances or that have small relative differences of the Mahalanobis distances. We demonstrate the effectiveness of the method for the benchmark data sets.

## 1 Introduction

Support vector machines are based on the theoretical learning theory developed by Vapnik [1], [2], [3, pp. 47–61]. In support vector machines, an $n$-class problem is converted into $n$ two-class problems in which one class is separated from the remaining classes. For each two-class problem, the original input space is mapped into the high dimensional dot product space called feature space and in the feature space, the optimal hyperplane that maximizes the generalization ability from the standpoint of the VC dimension is determined.

The high generalization ability compared to other methods has been shown for many applications but the major problem is slow training especially when the number of training data is large. Therefore, many methods for speeding up training have been proposed [2]. If support vectors are known in advance, training of support vector machines can be accelerated using only those data as the training data. Thus, in this paper we calculate the Mahalanobis distances for each data and estimate, as candidates of the boundary data, the training data that are misclassified by the Mahalanobis distances or that have small relative differences of the Mahalanobis distances. Finally, using two benchmark data sets we demonstrate the speedup of training by the proposed method.

## 2 Architecture of Support Vector Machines

Let $m$-dimensional inputs $\mathbf{x}_i\,(i = 1, \ldots, M)$ belong to Class 1 or 2 and the associated labels be $y_i = 1$ for Class 1 and $-1$ for Class 2. If these data are linearly separable, we can determine the decision function:

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

where $\mathbf{w}$ is an $m$-dimensional vector, $b$ is a scalar, and

$$y_i\, (\mathbf{w}^t\, \mathbf{x}_i + b) \geq 1 \quad \text{for} \quad i = 1, \ldots, M. \tag{2}$$

The hyperplane $D(\mathbf{x}) = \mathbf{w}^t\, \mathbf{x} + b = c$ for $-1 < c < 1$ forms a separating hyperplane that separates $\mathbf{x}_i\, (i = 1, \ldots, M)$. The distance between the separating hyperplane and the training datum nearest to the hyperplane is called the margin. The hyperplane $D(\mathbf{x}) = 0$ with the maximum margin for $-1 < c < 1$ is called the optimal separating hyperplane.

Now consider determining the optimal separating hyperplane. The Euclidean distance from a training datum $\mathbf{x}$ to the separating hyperplane is given by $|D(\mathbf{x})|/\|\mathbf{w}\|$. Thus assuming the margin $\delta$, all the training data must satisfy

$$\frac{y_k D(\mathbf{x}_k)}{\|\mathbf{w}\|} \geq \delta \qquad \text{for} \quad k = 1, \ldots, M. \tag{3}$$

If $\mathbf{w}$ is a solution, $a\mathbf{w}$ is also a solution where $a$ is a scalar. Thus we impose the following constraint:

$$\delta\, \|\mathbf{w}\| = 1. \tag{4}$$

From (3) and (4), to find the optimal separating hyperplane, we need to find $\mathbf{w}$ with the minimum Euclidean norm that satisfies (2).

The data that satisfy the equality in (2) are called support vectors.

Now the optimal separating hyperplane can be obtained by minimizing

$$\frac{1}{2}\, \|\mathbf{w}\|^2 \tag{5}$$

with respect to $\mathbf{w}$ and $b$ subject to the constraints:

$$y_i\, (\mathbf{w}^t\, \mathbf{x}_i + b) \geq 1 \qquad \text{for} \qquad i = 1, \ldots, M. \tag{6}$$

The number of variables for the convex optimization problem given by (5) and (6) is the number of features plus 1: $m + 1$. We convert (5) and (6) into the equivalent dual problem whose number of variables is the number of training data.

First we convert the constrained problem given by (5) and (6) into the unconstrained problem:

$$Q(\mathbf{w}, b, \alpha) = \frac{1}{2}\, \mathbf{w}^t\, \mathbf{w} - \sum_{i\,=\,1}^{M} \alpha_i\, \{y_i\, (\mathbf{w}^t\, \mathbf{x}_i + b) - 1\}, \tag{7}$$

where $\alpha = (\alpha_1, \ldots, \alpha_M)^t$ is the Lagrange multiplier. The optimal solution of (7) is given by the saddle point where (7) is minimized with respect to $\mathbf{w}$ and $b$ and it is maximized with respect to $\alpha_i \, (\geq 0)$. Then, we obtain the following dual problem. Namely, maximize

$$Q(\alpha) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,\, j=0}^{M} \alpha_i \, \alpha_j \, y_i \, y_j \, \mathbf{x}_i^t \, \mathbf{x}_j \tag{8}$$

with respect to $\alpha_i$ subject to the constraints

$$\sum_{i=1}^{M} y_i \, \alpha_i = 0, \, \alpha_i \geq 0 \quad \text{for} \quad i = 1, .., M. \tag{9}$$

Solving (8) and (9) for $\alpha_i \, (i = 1, \ldots, M)$, we can obtain the support vectors for Classes 1 and 2. Then the optimal hyperplane is placed at the equal distances from the support vectors for Classes 1 and 2.

To allow the data that do not have the maximum margin to exist, we introduce the nonnegative slack variables into (2). The resulting optimization problem is similar to the above formulation. The difference is the addition of the upper bound $C$ for $\alpha_i$.

If the original input $\mathbf{x}$ are not sufficient to guarantee linear separability of the training data, the obtained classifier may not have high generalization ability although the hyperplanes are determined optimally. Thus to enhance linear separability, in the support vector machines, the original input space is mapped into a high-dimensional dot product space called feature space using the kernel function that satisfies Mercer's condition. The kernel functions used in this paper are 1) polynomials with the degree of $d$: $H(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^t \, \mathbf{x}' + 1)^d$, and 2) radial basis functions: $H(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \, \|\mathbf{x} - \mathbf{x}'\|)$.

## 3 Speeding-up Training by Extracting Boundary Data

According to the architecture of the support vector machine, only the training data that are near the boundaries are necessary. In addition, since the training time becomes longer as the number of training data increases, the training time is shortened if the data that are far from the boundary are deleted. Therefore, if we can delete unnecessary data from the training data efficiently prior to training, we can speed up the training. In the following, we estimate the data that are near the boundaries using the classifier based on the Mahalanobis distance [4] and extracting the misclassified data and the data that are near the boundaries.

### 3.1 Approximation of Boundary Data

The decision boundaries of the classifier using the Mahalanobis distance are expressed by the polynomials, of the input variables, with the degree of two. Therefore, the boundary data given by the classifier are supposed to well approximate the boundary data for the support vector machine, especially with the polynomials with the degree of two as kernel functions.

For the class $i$ data $\mathbf{x}$, the Mahalanobis distance $d_i(\mathbf{x})$ is given by

$$d_i^2(\mathbf{x}) = (\mathbf{c}_i - \mathbf{x})^t Q_i^{-1} (\mathbf{c}_i - \mathbf{x}), \tag{10}$$

where $\mathbf{c}_i$ and $Q_i$ are the center vector and the covariance matrix for the data belonging to class $i$, respectively:

$$\mathbf{c}_i = \frac{1}{|X_i|} \sum_{\mathbf{x} \in X_i} \mathbf{x}, \tag{11}$$

$$Q_i = \frac{1}{|X_i|} \sum_{\mathbf{x} \in X_i} (\mathbf{x} - \mathbf{c}_i)\,(\mathbf{x} - \mathbf{c}_i)^t. \tag{12}$$

Here, $X_i$ denotes the set of data belonging to class $i$ and $|X_i|$ is the number of data in the set. The data $\mathbf{x}$ is classified into the class with the minimum Mahalanobis distance. The most important feature of the Mahalanobis distance is that it is invariant for linear transformation of input variables. Therefore, we do not worry about the scaling of each input variable.

For the datum belonging to class $i$, we check whether

$$r(\mathbf{x}) = \frac{\displaystyle\min_{j \neq i, j=1,\ldots,n} d_j(\mathbf{x}) - d_i(\mathbf{x})}{d_i(\mathbf{x})} \leq \eta \tag{13}$$

is satisfied, where $r(\mathbf{x})$ is the relative difference of distances, $\eta\,(> 0)$ controls the nearness to the boundary. If $r(\mathbf{x})$ is negative, the datum is misclassified. We assume the misclassified data are near the decision boundary. Inequality (13) is satisfied when the second minimum Mahalanobis distance is shorter than or equal to $(1+\eta)\,d_i(\mathbf{x})$ when the datum is correctly classified.

In extracting boundary data, we set some appropriate value to $\eta$ and for each class we select the boundary data that are at least equal to or more than the prespecified minimum number $N_{\min}$ and that are equal to or smaller than the maximum number $N_{\max}$. Here the minimum number is set so that the number of boundary data is not too small for some classes because the data that satisfy (13) are scarce. The maximum number is set not to allow too many data to be selected. The general procedure for extracting boundary data is as follows.

1. Calculate the centers and covariance matrices for all the classes using (11) and (12).
2. For the training datum $\mathbf{x}$ belonging to class $i$, we calculate $r(\mathbf{x})$ and we put the data into the stack for class $i$, $S_i$, whose elements are sorted in the increasing order of the value of $r(\mathbf{x})$ and whose maximum length is $N_{\max}$. We iterate this for all the training data.
3. If the stack $S_i$ includes more than $N_{\min}$ data that satisfy (13), we select these data as the boundary data for class $i$. Otherwise, we select the first $N_{\min}$ data as the boundary data.

### 3.2 Performance Evaluation

Although the performance varies as kernels vary, the polynomial kernels with the degree of two performed relatively well. Thus in the following, unless otherwise stated, we use the polynomials with the degree of two as the kernel functions in evaluating the iris data [5] and blood cell data [6].

We ran the software developed by Royal Holloway, University of London [7] on a SUN UltraSPARC-IIi (335MHz) workstation. The software used the pairwise classification [8] to resolve unclassified regions that arise by the original two-class formulation.

**Iris Data** Since the number of the iris data is small, we checked only the lowest rankings, in the relative difference of the Mahalanobis distances, of support vectors for the pairs of classes. Table 1 lists the results when the boundary data were extracted for each class. The numeral in the $i$th row and the $j$th column shows the lowest ranking of the support vector, belonging to class $i$, that separate class $i$ from class $j$. The diagonal elements show the number of training data for the associated class. The maximum value among lowest rankings was 8, which was smaller than half the number of class data. Thus, the relative difference of the Mahalanobis distances well reflected the boundary data.

**Table 1.** The lowest rankings of support vectors for the iris data

| Class | 1 | 2 | 3 |
|---|---|---|---|
| 1 | (25) | 1 | 2 |
| 2 | 8 | (25) | 3 |
| 3 | 2 | 3 | (25) |

**Blood Cell Data** We set $N_{\max}$ as the half of the maximum number of class data, namely 200. And we set $N_{\min} = 50$ and evaluated the performance changing $\eta$. Table 2 lists the results for the blood cell data. When $\eta \geq 1$, sufficiently good recognition rates were obtained for the test data and training was speeded up two to three times. (The numerals in the brackets in the "Rates" column show the recognition rates of the training data.)

Table 3 lists the speed-up of training for different kernels when $\eta = 2.0$. For each kernel, the upper row shows the results using all the training data and the lower row shows the results using the extracted boundary data. For different kernels, training was speeded up about two times and the recognition rates of the test data were almost the same.

## 4 Conclusions

We discussed fast training of support vector machines by extracting boundary data that are determined by the relative differences of the Mahalanobis distances.

**Table 2.** Performance for the blood cell data

| $\eta$ | Data | Rates (%) | Time (s) | Speedup |
|---|---|---|---|---|
| 0.5 | 1136 | 90.81 (97.45) | 96 (2) | 9.4 |
| 1.0 | 1693 | 92.06 (99.61) | 266 (2) | 3.4 |
| 1.5 | 1978 | 92.10 (99.29) | 390 (2) | 2.4 |
| 2.0 | 2102 | 92.13 (99.29) | 448 (2) | 2.1 |
| — | 3097 | 92.13 (99.32) | 924 | 1 |

**Table 3.** Performance for the blood cell data for different kernels ($\eta = 2.0$)

| Kernel | Parameter | Rates (%) | Time (s) | Speedup |
|---|---|---|---|---|
| Polynomial | $d = 3$ | 91.94 (99.94) | 937 | 1 |
| | | 92.00 (99.81) | 461 | 2.0 |
| | $d = 4$ | 92.10 (100) | 948 | 1 |
| | | 92.10 (99.90) | 471 | 2.0 |
| RBF | $\gamma = 1$ | 92.13 (100) | 2736 | 1 |
| | | 92.13 (99.97) | 1331 | 2.1 |
| | $\gamma = 0.1$ | 92.16 (100) | 2799 | 1 |
| | | 92.13 (99.97) | 1387 | 2.0 |

The computer simulations using the iris data and blood cell data showed that by this method the boundary data were efficiently extracted and training was speeded up about two times for the blood cell data.

## References

1. V. N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, 1998.
2. B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods: Support Vector Learning.* The MIT Press, 1999.
3. S. Abe. *Pattern Classification: Neuro-fuzzy Methods and Their Comparison.* Springer-Verlag, 2001.
4. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.
5. R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.
6. A. Hashizume, J. Motoike, and R. Yabe. Fully automated blood cell differential system and its application. In *Proc. IUPAC Third International Congress on Automation and New Technology in the Clinical Laboratory*, pages 297–302, 1988.
7. C. Saunders, M. O. Stitson, J. Weston, L. Bottou, B. Schölkopf, and A. Smola. Support vector machine reference manual. Technical Report CSD-TR-98-03, Royal Holloway, University of London, 1998 (http://svm.cs.rhbnc.ac.uk/).
8. U. H.-G. Kreßel. Pairwise classification and support vector machines. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 255–268. The MIT Press, 1999.