Fuzzy support vector machines for pattern classification

Abe, Shigeo

Inoue, Takuya

Fuzzy Support Vector Machines for Pattern Classification

Takuya Inoue and Shigeo Abe
Graduate School of Science and Technology, Kobe University, Kobe, Japan
E-mail: abe@eedept.kobe-u.ac.jp

Abstract

In conventional support vector machines (SVMs), an n-class problem is converted into n two-class problems. For the ith two-class problem we determine the optimal decision function which separates class i from the remaining classes. In classification, a datum is classified into class i only when the value of the ith decision function is positive. In this architecture, the datum is unclassifiable if the values of more than one decision function are positive or all the values are negative. In this paper, to overcome this problem, we propose fuzzy support vector machines (FSVMs). Using the decision functions obtained by training the SVM, for each class, we define a truncated polyhedral pyramidal membership function. Since, for the data in the classifiable regions, the classification results are the same for the two methods, the generalization ability of the FSVM is the same with or better than that of the SVM. We evaluate our method for three benchmark data sets and demonstrate the superiority of the FSVM over the SVM.

1 Introduction

Support vector machines (SVMs) are based on the theoretical learning theory developed by Vapnik. SVMs have been gained wide acceptance because of the high generalization ability for a wide range of applications [1, 2]. In the SVMs, original input space is mapped into a high-dimensional dot product space called feature space, and in the feature space the optimal hyperplane is determined to maximize the generalization ability.

However, there is a difficulty in extending binary two-class problems to n-class problems. In conventional SVMs for pattern classification, an n-class problem is converted into n two-class problems. For the ith two-class problem we determine the optimal decision function $D_i(\mathbf{x})$ so that class i is separated from the remaining classes. In classification, a datum $\mathbf{x}$ is classified into class i only when $D_i(\mathbf{x}) > 0$. In this architecture, the datum is unclassifiable if the values of more than two decision functions are positive or all the values are negative. To avoid this, in [3], a pairwise classification method, in which $n(n-1)/2$ decision functions are determined, is proposed. By this method, however unclassifiable regions remain.

In this paper, to overcome this problem, we propose fuzzy support vector machines (FSVMs). Using the decision functions obtained by training the SVM, we define truncated polyhedral pyramidal membership functions [4] and resolve unclassifiable regions.

In Section 2, we summarize support vector machines for pattern classification. And in Section 3 we discuss the problem of the multiclass support vector machines. In Section 4, we discuss the method of defining the membership functions using the SVM decision functions. Finally, in Section 5, we evaluate our method for three benchmark data sets and demonstrate the superiority of the FSVM over the SVM.

2 Two-Class Support Vector Machines

In training the support vector machines, an n-class problem is converted into n two-class problems. For each two-class problem, the decision function that maximizes the generalization ability is determined. For a two-class problem, the m-dimensional input $\mathbf{x}$ is mapped into the l-dimensional ($l \geq m$) feature space $\mathbf{z}$. Then in the feature space $\mathbf{z}$ the quadratic optimization problem is solved to separate two classes by the optimal separating hyperplane. In this section we discuss the support vector machine for a two-class problem.

2.1 The Optimal Hyperplane

Let m-dimensional input $\mathbf{x}_i\,(i = 1, \ldots, M)$ belong to class I or class II and the associated labels be $y_i = 1$ for class I and -1 for class II. 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 and b is a scalar. The separating hyperplane satisfies:

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

The separating hyperplane that has the maximum distance between the hyperplane and the nearest data, i.e., the maximum margin, is called optimal hyperplane (see Fig. 1). The generalization ability is maximized by the optimal hyperplane [1].

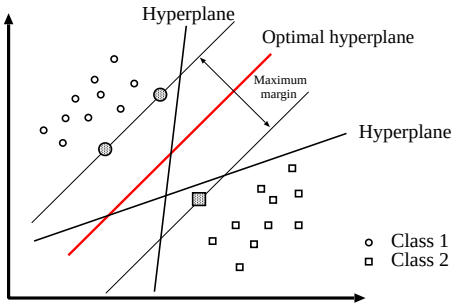

Figure 1: Optimal hyperplane

The optimal hyperplane can be obtained by solving the following convex quadratic optimization problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}||\mathbf{w}||^2 \\ \text{subject to} & y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1. \end{array} \tag{3}$$

When the number of features is small, we can solve this by the quadratic programming technique. When the number of features is large, we can convert (3) into the following equivalent dual problem whose number of variables is the number of training data:

$$\begin{aligned} \text{maximize} \quad & Q(\alpha) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=0}^{M} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j \\ \text{subject to} \quad & \sum_{i=1}^{M} y_i \alpha_i = 0, \\ & \alpha_i \geq 0 \quad \text{for} \quad i = 1, .., M, \end{aligned} \tag{4}$$

where $\alpha = (\alpha_1, \ldots, \alpha_M)$ is the Lagrange multiplier.

Let the optimal solution be α^* and b^*. According to the Kuhn-Tucker theorem, in (2) the equality condition holds for the training input-output pair $(\mathbf{x}_i, y_i)$ only if the associated α^* is not 0. In this case the training data $\mathbf{x}_i$ are support vectors. Solving (4) for $\alpha = (\alpha_1, \ldots, \alpha_M)$, we can obtain the support vectors for classes I and II. Then optimal hyperplane is placed at the equal distances from the support vectors for classes I and II, and b^* is given by

$$b^* = -\frac{1}{2} \sum_{k=1}^{M} y_k \alpha_k^* (\mathbf{s}_1^t \mathbf{x}_k + \mathbf{s}_2^t \mathbf{x}_k), \tag{5}$$

where $\mathbf{s}_1$ and $\mathbf{s}_2$ are, respectively, arbitrary support vectors for class 1 and class 2.

In the above discussion, we assumed that the training data are linearly separable. In the case where the training data are not linearly separable, we introduce nonnegative slack variables ξ_i to (2) and add, to the objective function given by (4), the sum of the slack variables multiplied by the parameter C. This corresponds to adding the upper bound C to α. In both cases, the decision functions are the same and are given by

$$D(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i^* y_i \mathbf{x}_i^t \mathbf{x} + b^*. \tag{6}$$

Then unknown datum $\mathbf{x}$ is classified as follows:

$$\mathbf{x} \in \begin{cases} \text{Class 1} & \text{if } D(\mathbf{x}) > 0, \\ \text{Class 2} & \text{otherwise.} \end{cases} \tag{7}$$

2.2 Mapping to a High-dimensional Space

In a support vector machine for a two-class problem the optimal hyperplane is determined to maximize the generalization ability. However, 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 hyperplane is determined optimally. To enhance linear separability, in support vector machines, the original input space is mapped into a high-dimensional dot product space called feature space.

Now using the nonlinear vector function $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_l(\mathbf{x}))^t$ that maps the m-dimensional input vector $\mathbf{x}$ into the l-dimensional feature space, the linear decision function in dual form is given by

$$D(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i y_i \mathbf{g}(\mathbf{x}_i)^t \mathbf{g}(\mathbf{x}). \tag{8}$$

According to the Hilbert-Schmidt theory the dot product in the feature space can be expressed by a symmetric kernel function $H(\mathbf{x}, \mathbf{x}')$:

$$H(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{l} g_j(\mathbf{x}) g_j(\mathbf{x}'), \tag{9}$$

if

$$\int\int H(\mathbf{x},\mathbf{x}')h(\mathbf{x})h(\mathbf{x}')d\mathbf{x}d\mathbf{x}' \geq 0 \tag{10}$$

is satisfied for all the square integrable functions $h(\mathbf{x})$ in the compact subset of the input space. This condition is called Mercer's condition.

Using kernel functions, without treating the high dimensional data explicitly, we can construct a nonlinear classifier using the method discussed above. Then unknown data are classified using the kernel function as follows.

$$\mathbf{x} \in \begin{cases} \text{Class 1} & \text{if } f(\mathbf{x}) = +1, \\ \text{Class 2} & \text{if } f(\mathbf{x}) = -1, \end{cases} \tag{11}$$

where

$$f(\mathbf{x}) = \text{sign}\left(\sum_{\text{support vectors}} y_i\alpha_i^* H(\mathbf{x},\mathbf{x}_i)\right). \tag{12}$$

3 Multiclass Support Vector Machines

For the conventional support vector machines, an n-class problem is converted into n two-class problems and for the ith two-class problem, class i is separated from the remaining classes. Let the ith decision function that classifies class i and the remaining classes be

$$D_i(\mathbf{x}) = \mathbf{w}_i^t\mathbf{x} + b_i. \tag{13}$$

The hyperplane $D_i(\mathbf{x}) = 0$ forms the optimal separating hyperplane and the support vectors belonging to class i satisfy $D_i(\mathbf{x}) = 1$ and to those belonging to the remaining class satisfy $D_i(\mathbf{x}) = -1$. For conventional support vector machine, if for the input vector $\mathbf{x}$

$$D_i(\mathbf{x}) > 0 \tag{14}$$

is satisfied for one i, $\mathbf{x}$ is classified into class i.

But if (14) is satisfied for plural i's, or there is no i that satisfies (14), $\mathbf{x}$ is unclassifiable (see Fig. 2). To solve this problem, pairwise classification [3] is proposed. In this method, we convert the n-class problem into $n(n-2)/2$ two-class problems, which cover all pair of classes. Let the decision function for class i against class j, with the maximum margin, be

$$D_{ij}(\mathbf{x}) = \mathbf{w}_{ij}^t\mathbf{x} + b_{ij}, \tag{15}$$

where $D_{ij}(\mathbf{x}) = -D_{ji}(\mathbf{x})$. For the input vector $\mathbf{x}$ we calculate

$$D_i(\mathbf{x}) = \sum_{i=1,\ldots,n} \text{sign}(D_{ij}(\mathbf{x})) \tag{16}$$

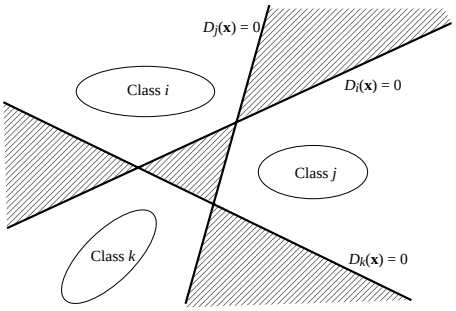

Figure 2: Unclassifiable region by the two-class formulation

and classify $\mathbf{x}$ into the class

$$\arg\max_{i=1,\ldots,n} D_i(\mathbf{x}). \tag{17}$$

In this formulation, however, unclassifiable regions remain, where some $D(\mathbf{x}_i)$ have the same values.

In the next section to resolve unclassifiable regions, we propose fuzzy support vector machines for conventional one-to-$(n-1)$ formulation.

4 Fuzzy Support Vector Machines

To resolve unclassifiable regions, we introduce the fuzzy membership functions while realizing the same classification results for the data that satisfy (14). To do this, for class i we define one-dimensional membership functions $m_{ij}(\mathbf{x})$ on the directions orthogonal to the optimal separating hyperplanes $D_j(\mathbf{x}) = 0$ as follows:

1. For $i = j$

$$m_{ii}(\mathbf{x}) = \begin{cases} 1 & \text{for} \quad D_i(\mathbf{x}) > 1, \\ D_i(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{18}$$

2. For $i \neq j$

$$m_{ij}(\mathbf{x}) = \begin{cases} 1 & \text{for} \quad D_j(\mathbf{x}) < -1, \\ -D_j(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{19}$$

Since only the class i training data exist when $D_i \geq 1$, we assume that the degree of class i is 1, and otherwise, $D_i(\mathbf{x})$. Here we allow the negative degree of membership.

For $i \neq j$, class i is on the negative side of $D_j(\mathbf{x}) = 0$. In this case, support vectors may not include class i data but when $D_i(\mathbf{x}) \leq -1$, we assume that the degree of membership of class i is 1, and otherwise, $-D_j(\mathbf{x})$.

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

$$m_i(\mathbf{x}) = \min_{j=1,\ldots,n} m_{ij}(\mathbf{x}). \tag{20}$$

In this formulation, the shape of the membership function is a polyhedral pyramid (see Fig. 3).

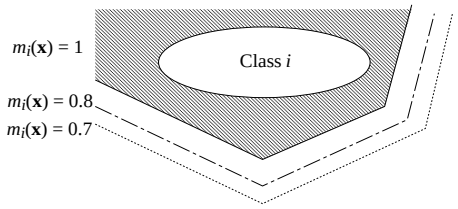

Figure 3: Contour lines of the class i membership function

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

$$\arg \max_{i=1,\ldots,n} m_i(\mathbf{x}). \tag{21}$$

If $\mathbf{x}$ satisfies

$$D_k(\mathbf{x}) \begin{cases} > 0 & \text{for} \quad k = i, \\ \leq 0 & \text{for} \quad k \neq i,\ k = 1, \ldots, n, \end{cases} \tag{22}$$

from (18) and (19), $m_i(\mathbf{x}) > 0$ and $m_j(\mathbf{x}) \leq 0\,(j \neq i, j = 1, \ldots, n)$ hold. Thus, $\mathbf{x}$ is classified into class i. This is equivalent to the condition that the condition that (14) is satisfied for only one i.

Now suppose (14) is satisfied for $i_1, \ldots, i_l\,(l > 1)$. Then, from (18) to (20), $m_k(\mathbf{x})$ is given as follows.

1. $k \in i_1, \ldots, i_l$

$$m_k(\mathbf{x}) = \min_{j=i_1,\ldots,i_l,\, j\neq k} -D_j(\mathbf{x}). \tag{23}$$

2. $k \neq j (j = i_1, \ldots, i_l)$

$$m_k(\mathbf{x}) = \min_{j=i_1,\ldots,i_l} -D_j(\mathbf{x}). \tag{24}$$

Thus the maximum degree of membership is achieved among $m_k(\mathbf{x}), k = i_1, \ldots, i_l$. Namely, $D_k(\mathbf{x})$ is maximized in $k \in \{i_1, \ldots, i_l\}$.

Let (14) be not satisfied for any class. Then,

$$D_i(\mathbf{x}) < 0 \qquad \text{for } j = 1, \ldots, n. \tag{25}$$

Then (20) is given by

$$m_i(\mathbf{x}) = D_i(\mathbf{x}). \tag{26}$$

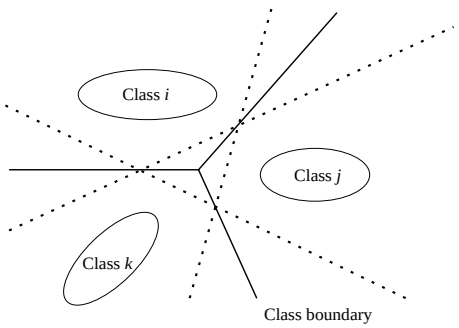

Figure 4: Class boundary with membership functions

According to above formulation, the unclassified regions shown in Fig. 2 are resolved as shown in Fig. 4 and generalization ability of FSVMs is the same with or better than that of the conventional SVMs.

In realizing the fuzzy pattern classification, we need not implement the membership functions $m_i(\mathbf{x})$ given by (20). The procedure of classification is as follows.

1. For $\mathbf{x}$, if $D_i(\mathbf{x}) > 0$ is satisfied for only one class, the input is classified into the class. Otherwise, go to Step 2.
2. If $D_i(\mathbf{x}) > 0$ is satisfied for more than one class $i\,(i = i_1, \ldots, i_l, l > 1)$, classify the datum into the class with the maximum $D_i(\mathbf{x})\,(i \in \{i_1, \ldots, i_l\})$. Otherwise, go to Step 3.
3. If $D_i(\mathbf{x}) \leq 0$ is satisfied for all the classes, classify the datum into the class with the minimum absolute value of $D_i(\mathbf{x})$.

5 Performance Evaluation

We evaluated the performance improvement of the fuzzy support vector machine over the conventional support vector machine using the thyroid data [5], blood cell data [6], and hiragana data [7] listed in Table 1.

Table 1: Feature of benchmark data

Data	Inputs	Classes	Train.	Test
Thyroid	21	3	3772	3428
Blood cell	13	12	3097	3100
Hiragana	50	39	4610	4610

We assumed that the data sets were not linearly separable and solved the optimization problem, using [8],

repeatedly reading 50 data at one time. We used the polynomial kernel functions and we set the value of the upper bound C so that the recognition rate using the dot product kernel was maximized.

For the hiragana data set, we normalize the kernel function by

$$H_{new}(\mathbf{x}, \mathbf{x}') = \frac{H(\mathbf{x}, \mathbf{x}')}{\max H(\mathbf{x}_t, \mathbf{x}_t')}$$

where $\mathbf{x}_t$ denotes the training data. Without this, the magnitudes of the solution became too small to continue calculations. We used a Pentium III 933 MHz personal computer.

Thyroid data. Tables 2 and 3 show the results of the conventional support vector machine for the training data and test data, respectively. "Plural" denotes that $D_i(\mathbf{x})$ are positive for plural classes and "Inactive" denotes that $D_i(\mathbf{x})$ are non-positive for all classes. Time denotes the training time. From these tables, as the degree of the polynomials increased, the numbers of the data belonging to unclassifiable regions were decreased. This meant that the linear separability in the feature space increased as the degree of the polynomials increased.

Table 4 lists the recognition rates of the conventional SVM and FSVM. The numerals in the brackets show the recognition rates of the training data. By the introduction of the membership functions, the recognition rates of the test and training data were improved for all the kernels.

Table 2: Performance for thyroid training data by SVM ($C = 5000$)

Kernel	Rate [%]	Plural	Inactive	Time [sec]
dot	94.06	7	153	20
Poly d =2	96.05	6	110	124
d =3	97.67	10	55	86
d =4	98.33	10	31	58
d =5	98.57	9	31	49

Table 3: Performance for thyroid test data by SVM

Kernel	Rate [%]	Plural	Inactive
dot	93.03	25	137
Poly d =2	93.61	40	121
d =3	94.60	50	80
d =4	95.01	54	60
d =5	95.19	49	64

Blood cell data. Tables 5 to 7 show the results for the blood cell data. From Tables 5 and 6 the recognition rates for the training and test data for the dot product kernel were very low due to a large number of unclassifiable data and training took more time than using other kernels. Though the number of unclassifiable data decreased as the degree of the polynomials increased, the recognition rates of the conventional support vector machine for the test data did not improved very much. As seen from Table 7, by introducing the membership function, the recognition rates of the test data were improved greatly. For $d = 4$ and $d = 5$, overfitting occurred.

Table 4: Performance comparison for thyroid data

Kernel	SVM [%]	FSVM [%]
Dot	93.03 (94.06)	95.27 (96.00)
Poly d =2	93.61 (96.05)	96.30 (98.20)
d =3	94.60 (97.67)	97.08 (98.80)
d =4	95.01 (98.33)	97.32 (99.18)
d =5	95.19 (98.57)	97.26 (99.23)

Table 5: Performance for blood cell training data by SVM ($C = 2000$)

Kernel	Rate [%]	Plural	Inactive	Time [sec]
dot	71.23	78	738	578
Poly d =2	93.67	56	94	60
d =3	96.09	40	53	53
d =4	97.71	26	25	53
d =5	98.71	14	11	51

Table 6: Performance for blood cell test data by SVM

Kernel	Rate [%]	Plural	Inactive
dot	67.58	122	779
Poly d =2	88.77	96	128
d =3	89.06	122	108
d =4	86.97	188	105
d =5	86.13	214	103

Hiragana data. Tables 8 to 10 show the results for the hiragana data. From Tables 8 and 9, the recognition rates of the training data reached 100% for the polynomial kernels and unclassifiable test data decreased monotonically as the degree increased. From Table 10 by introducing the membership function the recognition rates for the test data were improved.

Table 7: Performance comparison for blood cell data

Kernel	SVM [%]	FSVM [%]
dot	67.58 (71.23)	85.38 (88.54)
Poly d =2	88.77 (93.67)	93.00 (96.67)
d =3	89.06 (96.09)	93.39 (98.19)
d =4	86.97 (97.71)	92.65 (98.87)
d =5	86.13 (98.71)	92.74 (99.32)

Table 8: Performance for hiragana training data by SVM ($C = 2000$)

Kernel	Rate [%]	Plural	Inactive	Time [sec]
dot	94.49	55	191	240
Poly d =2	100	0	0	152
d =3	100	0	0	151
d =4	100	0	0	148
d =5	100	0	0	155

Table 9: Performance for hiragana test data by SVM

Kernel	Rate [%]	Plural	Inactive
dot	82.86	367	355
Poly d =2	95.73	73	110
d =3	96.20	66	103
d =4	96.33	66	97
d =5	96.53	65	90

Table 10: Performance comparison for hiragana data

Kernel	SVM [%]	FSVM [%]
dot	82.86 (94.49)	93.32 (97.87)
Poly d =2	95.73 (100)	99.07 (100)
d =3	96.20 (100)	99.35 (100)
d =4	96.33 (100)	99.37 (100)
d =5	96.53 (100)	99.32 (100)

6 Conclusions

In this paper we proposed fuzzy support vector machines for classification that resolve unclassifiable regions caused by conventional support vector machines. In theory, the generalization ability of the fuzzy support vector machine is superior to that of the conventional support vector machine. By computer simulations using three benchmark data sets, we demonstrated the superiority of our method over the conventional support vector machine.

Acknowledgments

We are grateful to Professor N. Matsuda of Kawasaki Medical School for providing the blood cell data and to Mr. P. M. Murphy and Mr. D. W. Aha of the University of California at Irvine for organizing the data bases including the thyroid data (ics.uci.edu: pub/machine-learning-databases).

References

[1] V. Cherkassky and F. Mulier, "*Learning from Data: Concepts, Theory, and Method,*" John Wiley & Sons, 1998.

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

[3] 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*, pp. 255-268, The MIT Press, Cambridge, MA, 1999.

[4] S. Abe, "*Pattern Classification: Neuro-Fuzzy Methods and Their Comparison*," Springer-Verlag, London, 2001.

[5] S. M. Weiss and I. Kapouleas, "An Empirical Comparison of Pattern Recognition Neural Nets and Machine Learning Classification Methods," *Proc. IJCAI–89*, pp. 781–787, 1989.

[6] A. Hashizume, J. Motoike, and R. Yabe, "Fully Automated Blood Cell Differential System and Its Application," *Proc. IUPAC Third International Congress on Automation and New Technology in the Clinical Laboratory*, pp. 297–302, Kobe, Japan, 1988.

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

[8] R. J. Vanderbei, "LOQO: An Interior Point Code for Quadratic Programming," Princeton University SOR-94-15, 1998.