



Analysis of Multiclass Support Vector Machines

Abe, Shigeo

(Citation)

Proc. International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA' 2003):385-396

(Issue Date)

2003-02

(Resource Type)

conference paper

(Version)

Accepted Manuscript

(URL)

<https://hdl.handle.net/20.500.14094/90000226>



Analysis of Multiclass Support Vector Machines

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

Abstract

Since support vector machines for pattern classification are based on two-class classification problems, unclassifiable regions exist when extended to problems with more than two classes. In our previous work, to solve this problem, we developed fuzzy support vector machines for one-against-all and pairwise classifications, introducing membership functions. In this paper, for one-against-all classification, we show that fuzzy support vector machines are equivalent to support vector machines with continuous decision functions. For pairwise classification, we discuss the relations between decision-tree-based support vector machines: DDAGs and ADAGs and compare classification performance of fuzzy support vector machines with that of ADAGs.

1 Introduction

Unlike neural networks or fuzzy systems, support vector machines directly determine the decision function for two-class problems so that the generalization ability is maximized while minimizing the training error. Because of this formulation, extension to multiclass problems is not unique. Original formulation by Vapnik [1] is one-against-all classification, in which one class is separated from the remaining classes. By this formulation, however, unclassifiable regions exist. There are roughly three ways to solve this problem: one-against-all, pairwise, and simultaneous classifications.

In one-against-all classification, instead of discrete decision functions, Vapnik [2, p. 438] proposed to use continuous decision functions. Namely, we classify a datum into the class with the maximum value of the decision functions. Inoue and Abe [3] proposed fuzzy support vector machines, in which membership functions are defined using the decision functions.

In pairwise classification, the n -class problem is converted into $n(n-1)/2$ two-class problems. Kreßel [4] showed that by this formulation, unclassifiable regions reduce, but still they remain. To resolve unclassifiable regions for pairwise classification, Platt, Cristianini, and J. Shawe-Taylor [5] proposed decision-tree-based pairwise classification called Decision Directed Acyclic Graph (DDAG). Pontil and Verri [6] proposed to use rules of a tennis tournament to resolve unclassified regions. Not knowing their work, Kijssirikul and Ussivakul [7] proposed the same method and called it Adaptive Directed Acyclic Graph (ADAG). Abe and Inoue [8] extended one-against-all fuzzy support vector machines to pairwise classification.

In simultaneous formulation we need to determine all the decision functions at once [9, 10], [2, pp. 437–440], which results in simultaneously solving a problem with larger number of variables than the above mentioned methods.

The aim of our paper is to clarify the relationships between different multiclass architectures. First we prove that one-against-all fuzzy support vector machines are equivalent to one-against-all support vector machines with continuous decision functions. Then we show that ADAGs are a subset of DDAGs and that the generalization ability depends on the structure of decision trees. Finally, we show the difference of generalization regions for ADAGs, DDAGs, and fuzzy support vector machines.

In Section 2, we explain two-class support vector machines, and in Section 3 we prove the equivalence of one-against-all fuzzy support vector machines and one-against-all support vector machines with continuous decision functions. In Section 4 we define DDAGs, ADAGs, and fuzzy support vector machines for pairwise classification, clarify their relations, and compare performance of the fuzzy support vector machine with that of ADAGs.

2 Two-class Support Vector Machines

Let m -dimensional inputs $\mathbf{x}_i$ ($i = 1, \dots, M$) belong to Class 1 or 2 and the associated labels be $y_i = 1$ for Class 1 and -1 for Class 2. Let the decision function be

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

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

$$y_i D(\mathbf{x}_i) \geq 1 - \xi_i \quad \text{for } i = 1, \dots, M. \quad (2)$$

Here ξ_i are nonnegative slack variables.

The distance between the separating hyperplane $D(\mathbf{x}) = 0$ and the training datum, with $\xi_i = 0$, nearest to the hyperplane is called the margin. The hyperplane $D(\mathbf{x}) = 0$ with the maximum margin is called the optimal separating hyperplane.

To determine the optimal separating hyperplane, we minimize

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^M \xi_i \quad (3)$$

subject to the constraints

$$y_i (\mathbf{w}^t \mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{for } i = 1, \dots, M, \quad (4)$$

where C is the upper bound that determines the tradeoff between the maximization of the margin and minimization of the classification error. We call the obtained hyperplane soft margin hyperplane. The data that satisfy the equality in (4) are called support vectors.

To enhance separability, the input space is mapped into the high-dimensional dot-product space called feature space. Let the mapping function be $\mathbf{g}(\mathbf{x})$. If the dot product in the feature space is express by $H(\mathbf{x}, \mathbf{x}') = \mathbf{g}(\mathbf{x})^t \mathbf{g}(\mathbf{x}')$, $H(\mathbf{x}, \mathbf{x}')$ is called kernel function, and we do not need to explicitly treat the feature space. To simplify notations, in the following we discuss support vector machines with the dot product kernel. The extension to the feature space is straightforward.

3 One-against-all Support Vector Machines

3.1 One-against-all Classification with Continuous Decision Functions

For the conventional support vector machine, let the i th decision function, with the maximum margin, that classifies class i and the remaining classes be

$$D_i(\mathbf{x}) = \mathbf{w}_i^t \mathbf{x} + b_i, \quad (5)$$

where $\mathbf{w}_i$ is the m -dimensional vector and b_i is a scalar.

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

$$D_i(\mathbf{x}) > 0 \quad (6)$$

satisfies for one i , $\mathbf{x}$ is classified into class i . Since only the sign of the decision function is used, the decision is discrete.

If (6) is satisfied for plural i 's, or there is no i that satisfies (6), $\mathbf{x}$ is unclassifiable (the shaded regions in Fig. 1 are unclassifiable regions). To avoid this, datum $\mathbf{x}$ is classified into the class:

$$\arg \max_i D_i(\mathbf{x}). \quad (7)$$

Since the continuous value of the decision function determines classification the decision is continuous.

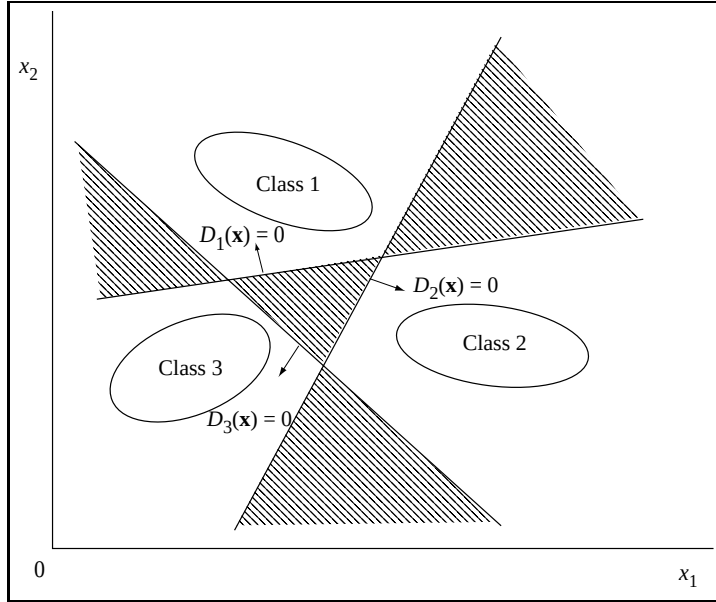


Figure 1: Unclassifiable regions by the one-against-all formulation.

3.2 Fuzzy Support Vector Machines

Here we summarize fuzzy support vector machines discussed in [3]. We introduce the membership function to resolve unclassifiable regions, while realizing the same classification results for the data that satisfy (6) for one i . To do this, for class i we define one-dimensional membership functions $m_{ij}(\mathbf{x})$ in the directions orthogonal to the optimal separating hyperplanes $D_j(\mathbf{x}) = 0$ as follows:

1. For $i = j$

$$m_{ii}(\mathbf{x}) = \begin{cases} 1 & \text{for } D_i(\mathbf{x}) \geq 1, \\ D_i(\mathbf{x}) & \text{otherwise.} \end{cases} \quad (8)$$

2. For $i \neq j$

$$m_{ij}(\mathbf{x}) = \begin{cases} 1 & \text{for } D_j(\mathbf{x}) \leq -1, \\ -D_j(\mathbf{x}) & \text{otherwise.} \end{cases} \quad (9)$$

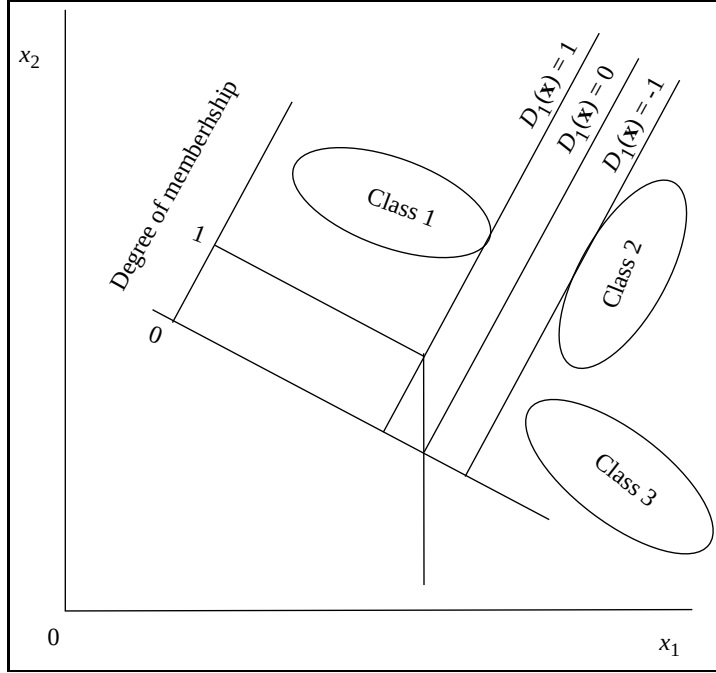


Figure 2: Definition of one-dimensional membership function.

Fig. 2 shows the membership function $m_{ii}(\mathbf{x})$. Since only the class i training data exist when $D_i(\mathbf{x}) \geq 1$, we assume that the degree of class i membership is 1, and otherwise, $D_i(\mathbf{x})$.

Here, we allow the negative degree of membership so that any data that are not on the boundary can be classified.

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 class i degree of membership is 1 and otherwise, $-D_j(\mathbf{x})$.

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

$$m_i(\mathbf{x}) = \min_{j=1, \dots, n} m_{ij}(\mathbf{x}). \quad (10)$$

The shape of the resulting membership function is the truncated polyhedral pyramid [11, pp. 76–78] in the feature space.

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

$$\arg \max_{i=1, \dots, n} m_i(\mathbf{x}). \quad (11)$$

According to the above formulation, the unclassifiable regions shown in Fig. 1 are resolved as shown in Fig. 3.

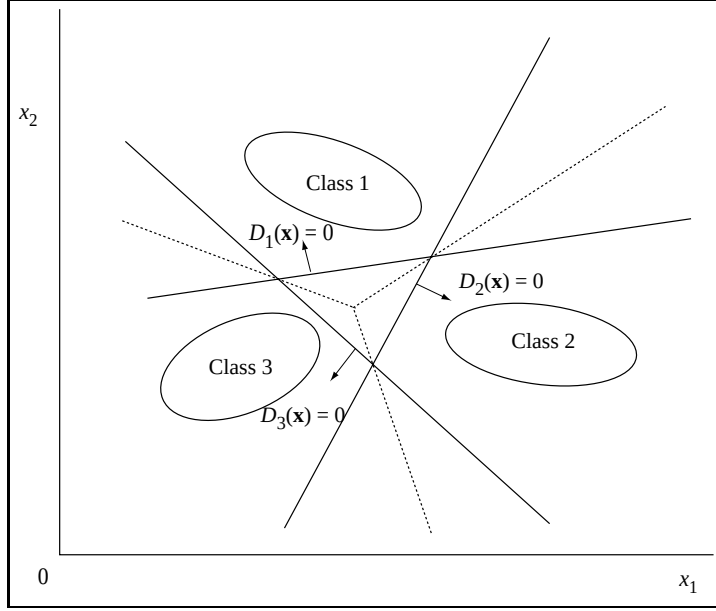


Figure 3: Extended generalization regions.

3.3 Equivalence of FSVMs and SVMs with Continuous Decision Functions

Here we show that one-against-all fuzzy support vector machines with the minimum operator are equivalent to one-against-all support vector machines with continuous decision functions.

If $\mathbf{x}$ satisfies

$$D_k(\mathbf{x}) \begin{cases} > 0 & \text{for } k = i, \\ \leq 0 & \text{for } k \neq i, k = 1, \dots, n, \end{cases} \quad (12)$$

from (8) and (9), $m_i(\mathbf{x}) \geq 0$ and $m_j(\mathbf{x}) < 0$ ($j \neq i, i = 1, \dots, n$) hold. Thus, $\mathbf{x}$ is classified into class i . This is equivalent to the condition that (6) is satisfied for one i and is equivalent to (7).

Now we assume, without loss of generality, that (6) is satisfied for $k = 1, \dots, l$ ($l > 1$). We also assume that $D_s(\mathbf{x})$ ($s \in \{1, \dots, l\}$) takes the maximum value among $D_k(\mathbf{x})$ ($k = 1, \dots, n$). Then,

$$D_k(\mathbf{x}) > 0 \quad \text{for } k = 1, \dots, l, \quad (13)$$

$$D_k(\mathbf{x}) \leq 0 \quad \text{for } k = l+1, \dots, n. \quad (14)$$

Thus from (8) to (10), $m_k(\mathbf{x})$ is given as follows.

1. For $k = 1, \dots, l$

Since

$$m_{kj}(\mathbf{x}) = \begin{cases} \min(1, D_k(\mathbf{x})) > 0 & \text{for } j = k, \\ -D_j(\mathbf{x}) < 0 & \text{for } j = 1, \dots, l, j \neq k, \\ \min(1, -D_j(\mathbf{x})) \geq 0 & \text{for } j = l+1, \dots, n, \end{cases}$$

$$m_k(\mathbf{x}) = \min_{j=1, \dots, l, j \neq k} -D_j(\mathbf{x}) < 0. \quad (15)$$

Thus, $m_s(\mathbf{x})$ is the largest among $m_k(\mathbf{x})$ ($k = 1, \dots, l, k \neq s$):

$$m_s(\mathbf{x}) = \min_{j=1, \dots, l, j \neq s} -D_j(\mathbf{x}) > m_k(\mathbf{x}). \quad (16)$$

2. For $k = l+1, \dots, n$

Since

$$m_{kj}(\mathbf{x}) = \begin{cases} D_k(\mathbf{x}) \leq 0 & \text{for } j = k, \\ -D_j(\mathbf{x}) < 0 & \text{for } j = 1, \dots, l, \\ \min(1, -D_j(\mathbf{x})) \geq 0 & \text{for } j = l+1, \dots, n, j \neq k, \end{cases}$$

$$m_k(\mathbf{x}) = \min \left(D_k(\mathbf{x}), \min_{j=1, \dots, l} -D_j(\mathbf{x}) \right) < 0. \quad (17)$$

Thus if

$$D_k(\mathbf{x}) < \min_{j=1, \dots, l} -D_j(\mathbf{x})$$

$m_k(\mathbf{x}) = D_k(\mathbf{x})$ holds. And if

$$D_k(\mathbf{x}) \geq \min_{j=1, \dots, l} -D_j(\mathbf{x}),$$

$m_k(\mathbf{x}) = -D_s(\mathbf{x})$ holds.

Thus from Cases 1 and 2, $m_s(\mathbf{x})$ takes the maximum value among $m_k(\mathbf{x})$ ($k = 1, \dots, n$); $\mathbf{x}$ is classified into class s . This is equivalent to (7).

Let (6) is not satisfied for any class. Then,

$$D_i(\mathbf{x}) \leq 0 \quad \text{for } j = 1, \dots, n. \quad (18)$$

Then (10) is given by

$$m_i(\mathbf{x}) = D_i(\mathbf{x}). \quad (19)$$

Thus the maximum degree of membership corresponds to the maximum value of $D_k(\mathbf{x})$. This is equivalent to (7).

Therefore, fuzzy support vector machines are equivalent to support vector machines with continuous decision functions.

4 Pairwise Support Vector Machines

4.1 DDAGs and ADAGs

In pairwise classification, we determine $n(n-1)/2$ decision functions for all pairs of classes. Let the decision function for class i against class j , with the maximum margin, be

$$D_{ij}(\mathbf{x}) = \mathbf{w}_{ij}^t \mathbf{x} + b_{ij}, \quad (20)$$

where $\mathbf{w}_{ij}$ is the m -dimensional vector, b_{ij} is a scalar, and $D_{ij}(\mathbf{x}) = -D_{ji}(\mathbf{x})$.

For the input vector $\mathbf{x}$ we calculate

$$D_i(\mathbf{x}) = \sum_{j \neq i, j=1}^n \text{sign}(D_{ij}(\mathbf{x})) \quad (21)$$

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

$$\arg \max_{i=1, \dots, n} D_i(\mathbf{x}). \quad (22)$$

But if (22) is satisfied for plural i 's, $\mathbf{x}$ is unclassifiable. In the shaded region in Fig. 4, $D_i(\mathbf{x}) = 1$ ($i = 1, 2$, and 3). Thus the shaded region is unclassifiable.

To resolve unclassifiable regions for the pairwise classification, Platt, Cristianini, and J. Shawe-Taylor [5] proposed decision-tree-based pairwise classification called Decision Directed Acyclic Graph (DDAG). Fig. 5 shows the decision tree for the three classes shown in Fig. 4. In the figure, $\bar{i}$ shows that associated class is not i . As the top-level classification, we can choose any pair of classes. And except for the leaf node if $D_{ij}(\mathbf{x}) > 0$, we consider that $\mathbf{x}$ does not belong to class j . If $D_{12}(\mathbf{x}) > 0$, $\mathbf{x}$ does not belong to Class 2. Thus it belongs to either Class 1 or 3 and the next classification pair is Classes 1 and 3. The generalization regions become as shown in Fig. 7. Unclassifiable regions are resolved but clearly the generalization regions depends on the tree formation.

Pontil and Verri [6] proposes to use rules of a tennis tournament to resolve unclassified regions. Not knowing their work, Kijssirikul and Ussivakul [7] proposes the same method

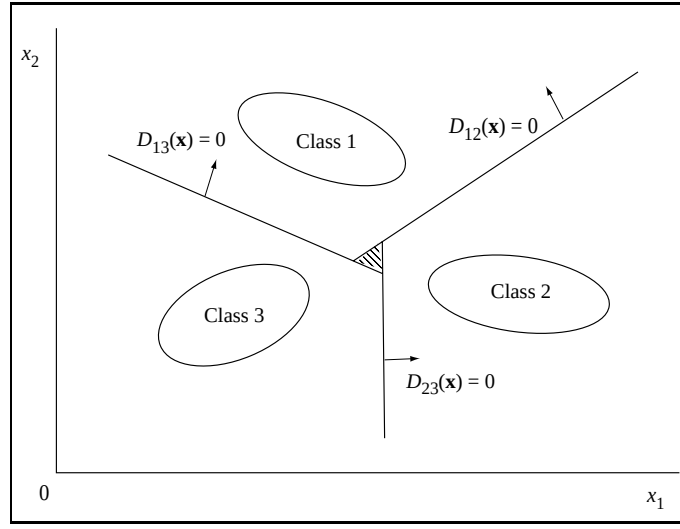


Figure 4: Unclassifiable regions by the pairwise formulation.

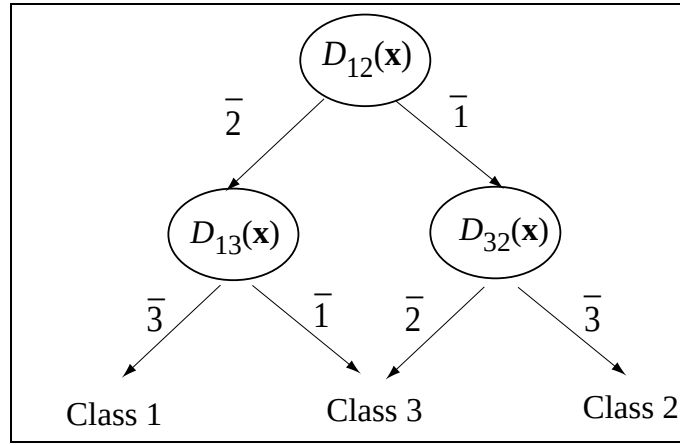


Figure 5: DDAG.

and called it Adaptive Directed Acyclic Graph (ADAG). For three-class problems, the ADAG is equivalent to the DDAG. Reconsider the example shown in Fig. 4. Let the first round matches be $\{\text{Class 1, Class 2}\}$ and $\{\text{Class 3}\}$. Then for an input $\mathbf{x}$, in the first match, $\mathbf{x}$ is classified into Class 1 or Class 2, and in the second match $\mathbf{x}$ is classified into Class 3. Then the second round match is either $\{\text{Class 1, Class 3}\}$ or $\{\text{Class 2, Class 3}\}$ according to the outcome of the first round match. The resulting generalization regions for each class are the same as those shown in Fig. 6. Thus for three-class problems there are three different ADAGs, each having an equivalent DDAG.

When the number of classes is more than three, the set of ADAGs is included in the set of DDAGs. In general, for an ADAG including n classes, we can generate an equivalent DDAG. But the reverse is not true for $n \geq 4$. The number of different ADAGs for an n class problem is given by $n!/(2^{\lfloor n/2 \rfloor} 2^{\lfloor n/2^2 \rfloor} \dots 2^0)$, where $\lfloor x \rfloor$ is the maximum integer that does not exceed x . According to the computer experiments [12, 7], classification performance of the two methods is almost identical. Thus in the following computer experiments, we compare ADAGs with fuzzy support vector machines.

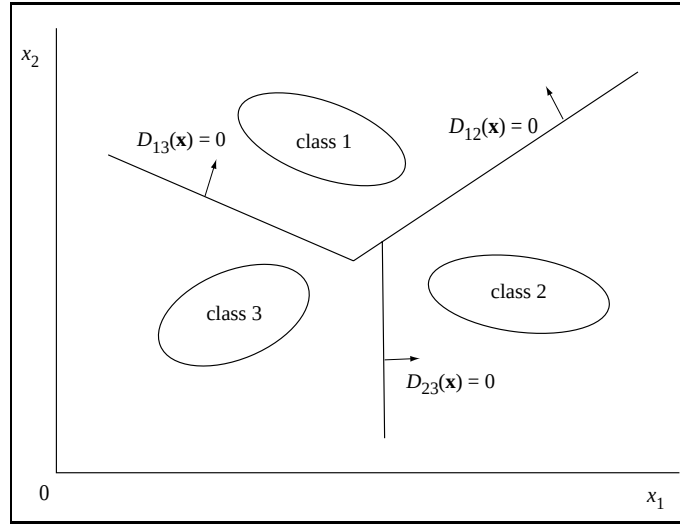


Figure 6: Generalization region by DDAG.

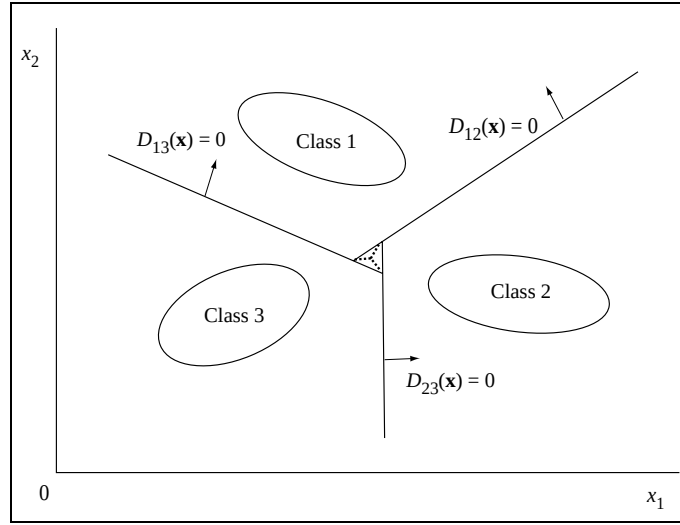


Figure 7: Extended generalization regions.

4.2 Fuzzy Support Vector Machines

Similar to one-against-all support vector machines, we introduce the membership function to resolve unclassifiable regions while realizing the same classification results with that of the conventional pairwise classification [8]. To do this, for the optimal separating hyperplane $D_{ij}(\mathbf{x}) = 0$ ($i \neq j$) we define one-dimensional membership functions $m_{ij}(\mathbf{x})$ in the directions orthogonal to $D_{ij}(\mathbf{x}) = 0$ as follows:

$$m_{ij}(\mathbf{x}) = \begin{cases} 1 & \text{for } D_{ij}(\mathbf{x}) \geq 1, \\ D_{ij}(\mathbf{x}) & \text{otherwise.} \end{cases} \quad (23)$$

Using $m_{ij}(\mathbf{x})$ ($j \neq i, j = 1, \dots, n$), we define the class i membership function of $\mathbf{x}$ using the minimum operator:

$$m_i(\mathbf{x}) = \min_{j=1, \dots, n} m_{ij}(\mathbf{x}). \quad (24)$$

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

$$\arg \max_{i=1,\dots,n} m_i(\mathbf{x}). \quad (25)$$

Equation (24) is equivalent to

$$m_i(\mathbf{x}) = \min \left(1, \min_{j \neq i, j=1,\dots,n} D_{ij}(\mathbf{x}) \right). \quad (26)$$

Since $m_i(\mathbf{x}) = 1$ holds for only one class, (26) reduces to

$$m_i(\mathbf{x}) = \min_{j \neq i, j=1,\dots,n} D_{ij}(\mathbf{x}), \quad (27)$$

assuming that $m_i(\mathbf{x})$ takes a value larger than 1.

The unclassifiable region shown in Fig. 4 is resolved as shown in Fig. 7.

In general, fuzzy support vector machines resolve unclassifiable regions in the way that does not favor any classes, but decision-tree-based support vector machines favor one or some classes.

4.2.1 Comparison of Decision-tree-based and Fuzzy Support Vector Machines

We compared the maximum, minimum, and average recognition rates of the test data for ADAGs and the recognition rate of the pairwise fuzzy support vector machine. Table 1 lists the data sets used in our study [11]. In the table, TRAIN. and TEST denote the numbers of training and test data, respectively. We used the dot product kernel, polynomial kernel: $(\mathbf{x}^t \mathbf{x}' + 1)^d$, and RBF kernel: $\exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$. We trained support vector machines by the steepest ascent method [13] with $C = 5000$.

Table 1: Benchmark data specification

DATA	INPUTS	CLASSES	TRAIN.	TEST
Thyroid	21	3	3772	3428
Blood cell	13	12	3097	3100
Hiragana	13	38	8375	8356

Table 2 shows the performance comparison for the conventional pairwise support vector machine (SVM), pairwise fuzzy support vector machine (FSVM), and ADAGs. For ADAGs, the maximum, minimum, and average recognition rates of the test data are shown. Poly4 denotes the polynomial kernel with degree 4 and RBF10 denotes the RBF kernel with $\gamma = 10$. The column NUM shows the number of ADAGs whose recognition rates are larger than or equal to that of the FSVM. The numeral in parentheses is the percentage of occurrence.

For the thyroid data since the number of classes is three, there are only three different ADAGs. The total number of ADAGs for the blood cell data is 467,775 and we generated

all the ADAGs and randomly selected 47,006 ADAGs. The total number of ADAGs for the hiragana data is 3.8×10^{38} and we randomly generated 40,000 ADAGs.

From the table, FSVM and ADAG performed better than SVM and FSVM performed better than or comparable to the average ADAG. For the thyroid data, the training data for Class 3 occupy 92% of the total training data and $\text{ADAG} = \{1, 2\}, \{3\}$, in which the unclassifiable region is labeled as Class 3, showed the best or second best performance. Thus the biased data distributions might make ADAG perform better than FSVM for the dot product and RBF kernels. For the blood cell and hiragana data, the rates that ADAG performed better or equal to FSVM are small. In addition, even if the maximum recognition rates of ADAG exceed the recognition rates of FSVM, the differences are small.

Table 2: Performance of pairwise SVMs in %

DATA	KERNEL	SVM	FSVM	ADAG			NUM
				Max	Min	Ave	
Thyroid	Dot	97.17	97.37	97.40	97.17	97.28	1
	Poly4	96.27	96.62	96.59	96.30	96.40	0
	RBF10	95.10	95.16	95.25	95.10	95.17	2
Blood Cell	Dot	90.55	91.32	91.55	90.58	91.01	1237 (2.6)
	Poly4	91.26	92.35	92.32	91.42	91.88	0
	RBF10	91.52	91.74	91.87	91.52	91.64	5290 (11)
Hiragana	Dot	99.23	99.44	99.53	99.15	99.34	2655 (6.6)
	Poly4	99.45	99.62	99.65	99.39	99.51	252 (0.6)
	RBF0.1	99.56	99.70	99.70	99.51	99.60	1

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).

5 Conclusions

To resolve or reduce unclassifiable regions caused by multiclass problems, one-against-all or pairwise support vector machines or their variants are usually used.

In this paper, we compared multiclass support vector machines theoretically and by computer experiments. For one-against-all support vector machines, we proved that fuzzy support vector machines with truncated polyhedral pyramidal membership functions in the feature space are equivalent to support vector machines with continuous decision functions. For pairwise support vector machines we showed that DDAGs, which are based on the decision tree, include ADAGs, which are based on rules of a tennis tournament.

Finally we demonstrated by computer experiments that pairwise fuzzy support vector machines performed better than average ADAGs.

References

- [1] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, London, UK, 1995.
- [2] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, NY, 1998.
- [3] T. Inoue and S. Abe. Fuzzy support vector machines for pattern classification. In *Proceedings of International Joint Conference on Neural Networks (IJCNN '01)*, volume 2, pages 1449–1454, July 2001.
- [4] 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, Cambridge, MA, 1999.
- [5] J. C. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 547–553. The MIT Press, 2000.
- [6] M. Pontil and A. Verri. Support vector machines for 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, 1998.
- [7] B. Kijssirikul and N. Ussivakul. Multiclass support vector machines using adaptive directed acyclic graph. In *Proceedings of International Joint Conference on Neural Networks (IJCNN 2002)*, pages 980–985, 2002.
- [8] S. Abe and T. Inoue. Fuzzy support vector machines for multiclass problems. In *Proceedings of 10th European Symposium on Artificial Neural Networks (ESANN'2002)*, pages 113–118, Bruges, Belgium, April 2002.
- [9] K. P. Bennett. Combining support vector and mathematical programming methods for classification. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 307–326. The MIT Press, Cambridge, MA, 1999.
- [10] J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *Proceedings of 7th European Symposium on Artificial Neural Networks (ESANN'99)*, pages 219–224, 1999.
- [11] S. Abe. *Pattern Classification: Neuro-fuzzy Methods and Their Comparison*. Springer-Verlag, London, UK, 2001.
- [12] C. Nakajima, M. Pontil, and T. Poggio. People recognition and pose estimation in image sequences. In *Proceedings of International Joint Conference on Neural Networks (IJCNN 2000)*, volume IV, pages 189–194, 2000.
- [13] S. Abe, Y. Hirokawa, and S. Ozawa. Steepest ascent training of support vector machines. In E. Damiani, R. J. Howlett, L. C. Jain, and N. Ichalkaranje, editors, *Knowledge-Based Intelligent Engineering Systems and Allied Technologies (KES 2002)*, volume Part 2, pages 1301–1305. IOS Press, 2002.