Fuzzy support vector machines for multilabel classification

Abe, Shigeo

(Citation)
Pattern Recognition,48(6):2110-2117

(Issue Date)
2015-06

(Resource Type)
journal article

(Version)
Accepted Manuscript

(URL)
https://hdl.handle.net/20.500.14094/90003293

Fuzzy Support Vector Machines for Multilabel Classification

Shigeo Abe

Kobe University
Rokkodai, Nada, Kobe, Japan

Abstract

The problem of one-against-all support vector machines (SVMs) for multilabel classification is that a data sample may be classified into a multilabel class that is not defined or it may not be classified into any class. To solve this problem, in this paper we propose fuzzy SVMs (FSVMs) for multilabel classification, in which for each multilabel class, a region with the associated membership function is defined and a data point is classified into a multilabel class whose membership function is the largest. By computer experiments, we show that the accuracy is improved by the FSVM over the conventional one-against-all SVM.

Keywords: Multilabel Classification, pattern classification, support vector machines, training

1. Introduction

In pattern classification, usually a data sample is classified into a single class. But in real world applications there may be cases where a sample belongs to more than one class. For instance, for classification of facial expression, a person may express happiness and relaxation at the same time. Classification of this type is called multilabel classification in contrast to single-label classification.

Extensive work has been done to handle multilabel classification [1, 2]. Multilabel classification methods are classified into three categories: algo-

Email address: abe@kobe-u.ac.jp (Shigeo Abe)
URL: http://www2.kobe-u.ac.jp/~abe (Shigeo Abe)

rithm adaptation methods, problem transformation methods, and ensemble methods.

In algorithm adaptation methods, conventional classification methods such as support vector machines, decision trees, and boosting are adapted to multilabel classification [3, 4, 5].

Problem transformation methods convert multilabel classification into single-label classification [6, 1]. One of the widely used methods converts multilabel classification into single-label classification defining a new class to each multilabel. This method is called a label power-set method. The converted classification problem is usually solved by one-against-one classification. One of the problems with this method is that the number of classes may be increased greatly if many multilabels are used. Another method uses one-against-all classification. In determining a decision function that separates class i from the others, we place the data with multilabels that include the class i label on the positive side of the decision function and place the remaining data on the negative side. In classification, a data sample is classified into a single-label or multilabel class associated with positive decision functions. This method is sometimes called a binary relevance method.

In ensemble methods, each classifier in ensemble is based on either problem transformation or algorithm adaptation methods [7, 8].

By the binary relevance method, the number of classes does not increase but a data sample is unclassifiable if there is no positive decision function, and a data sample may be classified into a multilabel that is not included in the multilabels contained in the training set. We can implement the binary relevance method using any classifier, but because SVMs realize high generalization ability for a wide range of applications, SVMs are often used for implementing the binary relevance method. In [6, 9], the unclassifiable region is resolved by classifying a sample to the class associated with the maximum decision function value. This is the heuristics used in single-label classification.

To improve the generalization ability of one-against-all SVMs, in [9], one-against-all SVMs are extended to enforce the slack variables to be zero.

There are several approaches to handle uncertainty in classification such as belief function theory (or Dempster-Shafer theory) [10, 11] and fuzzy logic. In fuzzy logic, there has been much work in developing trainable fuzzy classifiers, in which fuzzy rules are extracted from training data [12]. Each class region is defined by fuzzy rules with membership functions. According to the shape of membership functions, fuzzy regions defined by the mem-

bership functions can be classified into hyperboxes [13, 14], hyper-ellipsoids [15, 16, 17], and hyper-polyhedrons [18, 19, 20, 21].

In fuzzy SVMs for single-label classification [19, 20, 21], the decision functions determined by SVMs are used to generate membership functions with hyper-polyhedral regions. Using the membership functions, the existence of unclassifiable regions and multilabel regions caused by multiclass SVMs is resolved. It is proved that the classification result by the fuzzy SVM is equivalent to the above heuristics [22].

In [23], based on the membership function discussed in [20, 21], unclassifiable regions obtained by the one-against-one SVM for multilabel classification are resolved. However, the undefined multilabel classes are not resolved.

In this paper, we propose fuzzy SVMs (FSVMs) for multilabel classification that resolve unclassifiable regions and undefined multilabel classes. For each single-label or multilabel class that is defined in the training data set, we define a fuzzy region using the decision functions. The degree of membership of a data sample to the fuzzy region is determined by the decision hyperplane that is nearest to the data sample. The data sample is classified into the single-label or multilabel class with the highest degree of membership.

This classification strategy is simplified for an unclassifiable region. If no decision function is positive for a data sample, it is classified into a class with the maximum degree of membership. This is the same as the fuzzy SVM for single-class classification. This explains the validity of the heuristics used in [6, 9] from the standpoint of fuzzy membership functions.

We demonstrate the effectiveness of the proposed fuzzy SVMs using several benchmark data sets.

In Section 2, we overview conventional two-class SVMs. Then in Section 3, we explain the conventional one-against-all and one-against-one SVMs for multilabel classification, and in Section 4 we propose the fuzzy SVM. In Section 5, we compare the fuzzy SVM with the conventional one-against-all and one-against-one SVMs using several benchmark data sets.

2. Support Vector Machines

In this section, according to [24], we briefly summarize the two-class L1 SVM, which is widely used among several SVM variants.

In the SVM, nonlinear separation is realized by using the nonlinear vector function $\boldsymbol{\phi}(\mathbf{x})$ that maps the m-dimensional input vector $\mathbf{x}$ into the l-

dimensional feature space. In the feature space, we determine the decision function that separates Class 1 data from Class 2 data:

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

where $\mathbf{w}$ is the l-dimensional vector and b is the bias term. Then the L1 SVM is formulated in the primal form as follows:

$$\text{minimize} \qquad Q(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^{M} \xi_i \tag{2}$$

$$\text{subject to} \qquad y_i \, D(\mathbf{x}_i) \geq 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, M, \tag{3}$$

where C is the margin parameter that controls the trade-off between the training error and the generalization ability, $\mathbf{x}_i$ are M m-dimensional training inputs and belong to Class 1 or 2 and the associated labels are $y_i = 1$ for Class 1 and -1 for Class 2, and $\xi_i \, (\geq 0)$ are the slack variables for $\mathbf{x}_i$.

Because in some cases the dimension of the feature space is infinite, usually we solve the following dual form, instead of solving (2) and (3):

$$\text{maximize} \qquad Q(\boldsymbol{\alpha}) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j \, y_i \, y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{4}$$

$$\text{subject to} \qquad \sum_{i=1}^{M} y_i \, \alpha_i = 0, \quad 0 \leq \alpha_i \leq C \quad \text{for} \quad i = 1, ..., M, \tag{5}$$

where α_i are Lagrange multipliers associated with $\mathbf{x}_i$ and $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}^\top(\mathbf{x}) \, \boldsymbol{\phi}(\mathbf{x}')$ is a kernel function. Using a kernel function, we can avoid treating the feature space directly.

Among several kernels, polynomial kernels and radial basis function (RBF) kernels are often used for pattern classification, but because in most cases RBF kernels perform better [24], in the following study we use RBF kernels:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2 / m), \tag{6}$$

where m is the number of inputs for normalization and γ is a spread of a radius.

3. Multilabel Classification

First we explain multilabel classification using an example [1] shown in Table 1. In the table, for example, Sample 1 belongs to Classes 1 and 4.

Multilabel classification can be converted into single-label classification. One way is to define a new class label for the multilabel for each data sample. This method is called a label power-set method. For Sample 1, we define a new class label 1&4, for Sample 3, 1&3, and for Sample 4, 2&4. Table 2 shows the resulting data set, which is converted to a single-label classification problem. The converted classification problem can be classified by any classifier, but in the following we consider classifying it by the one-against-one (OAO) SVM and thus we call this method OAO.

Another approach is to adopt the one-against-all (OAA) strategy: we train the binary classifier for class i so that the class i training samples are separated from the remaining samples. In classification, $\mathbf{x}$ is classified into a multilabel class [1]

$$L_{\mathrm{c}} = \{k \,|\, D_k(\mathbf{x}) > 0 \text{ for } k = 1, \ldots, n\}, \tag{7}$$

where n is the number of classes. This method is called a binary relevance method, but in the following we call it OAA. This is an extension of single-label one-against-all classification. By this formulation, however, $\mathbf{x}$ may not be classified into any class when all the decision functions are negative or may be classified into a multilabel class that is not defined in the training set.

Consider a two-class problem shown in Fig. 1. In the figure, the filled circle, the filled square, and the filled ellipsoid belong to Classes 1, 2, and 1 & 2, respectively. Then the decision functions $D_1(\mathbf{x})$ and $D_2(\mathbf{x})$ are obtained

Table 1: Example of a multilabel data set

Sample	Class			
	1	2	3	4
1	1	0	0	1
2	0	1	0	0
3	1	0	1	0
4	0	1	0	1

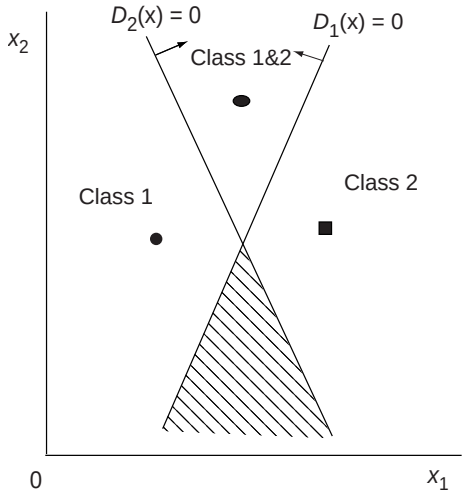

Figure 1: An unclassifiable region by one-against-all classification

as in the figure. According to (7), the data samples that satisfy $D_1(\mathbf{x}) > 0$ and $D_2(\mathbf{x}) > 0$ are classified into Class 1&2, and those that satisfy $D_1(\mathbf{x}) > 0$ and $D_2(\mathbf{x}) < 0$ or $D_1(\mathbf{x}) < 0$ and $D_2(\mathbf{x}) > 0$ are classified into Class 1 or Class 2. But those that satisfy $D_1(\mathbf{x}) < 0$ and $D_2(\mathbf{x}) < 0$ (the shaded region in the figure) are not classified into any class.

Reconsider the four-class problem shown in Table 1. Assuming that the dimension of the input variable is two and that each sample is placed as in Fig. 2, the obtained separating lines are as shown in the figure. According to (7), the regions separated by the lines are labeled as shown in the figure. Because $D_1(\mathbf{x})$ and $D_2(\mathbf{x})$ are the same, there is no unclassifiable region.

Table 2: Conversion to single-label classification by defining new classes

Sample	Class			
	1&4	2	1&3	2&4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

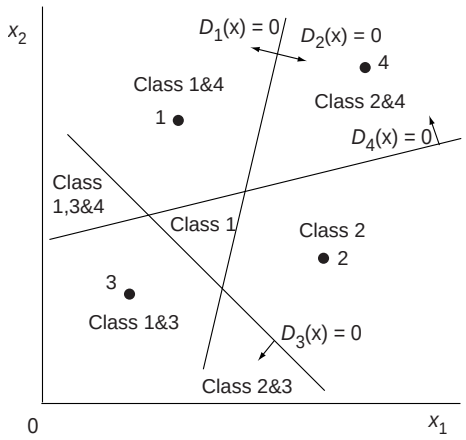

Figure 2: Undefined multilabel regions by one-against-all classification

Notice that single-label Classes 3 and 4 do not exit and multilabel classes 1,3&4 and 2&3 are generated according to (7) although they are not included in the training set.

4. Fuzzy Support Vector Machines

We now discuss how to resolve unclassifiable regions and undefined multilabel regions. Resolution of undefined multilabel regions is based on the assumption that all the multilabels for a given classification problem appear in the targets of the training set. If this assumption is violated, the correct prediction of multilabels that do not appear in the targets of the training set may be changed to one of the existing multilabels.

For an n class problem with classes 1 to n, we define new classes from class $n+1$ to class o for the distinct multilabels in the targets of the training data set, where $o-n$ is the number of newly defined classes. We call these classes multilabel classes and classes 1 to n, single-label classes. Multilabel class k consists of single-label classes k_1 to k_{e}, where $L_{k_{\mathrm{e}}} = \{k_1, \ldots, k_{\mathrm{e}}\} \subseteq L_n = \{1, \ldots, n\}$.

We assume that the optimal separating hyperplane $D_i(\mathbf{x}) = 0$ for class i ($i = 1, \ldots, n$) separates class i training samples from the remaining class

samples. Then, the convex region R_k

$$R_k = \{\mathbf{x} \,|\, D_i(\mathbf{x}) > 0 \text{ for } i \in L_{k_\mathrm{e}}, D_i(\mathbf{x}) < 0 \text{ for } i \in L_n - L_{k_\mathrm{e}}\} \quad \text{for} \quad k = 1, \ldots, o \tag{8}$$

is the region for class k and includes all the class k training samples, where if class k is a single-label class, i.e., for $1 \leq k \leq n$, $k_1 = k_\mathrm{e} = k$. We denote the region associated with multilabel L_c given by (7), R_c.

Regions R_k $(k = 1, \ldots, o)$ do not overlap. Therefore, if $\mathbf{x}$ is in R_k, it is classified into class k. But in general, the union of the regions does not occupy the whole feature space. In Fig. 1, R_1 and R_2 are given by $D_1(\mathbf{x}) > 0, D_2(\mathbf{x}) < 0$ and $D_1(\mathbf{x}) < 0, D_2(\mathbf{x}) > 0$, respectively. And R_3 for Class 1&2 is $D_1(\mathbf{x}) > 0, D_2(\mathbf{x}) > 0$. Therefore, the region given by $D_1(\mathbf{x}) < 0, D_2(\mathbf{x}) < 0$ does not belong to any class.

Consider classifying $\mathbf{x}$ into one of o classes. To do this we define a membership function of $\mathbf{x}$ to Region R_k $(k = 1, \ldots, o)$. We define the membership of $\mathbf{x}$ to the decision function $D_i(\mathbf{x})$ for $i \in L_{k_\mathrm{e}}$ by

$$m_{ki}(\mathbf{x}) = \min(1, D_i(\mathbf{x})) \text{ for } i \in L_{k_\mathrm{e}}. \tag{9}$$

If $D_i(\mathbf{x}) \geq 1$, $m_{ki}(\mathbf{x}) = 1$, namely, the membership function saturates to 1 but we allow negative membership if $D_i(\mathbf{x}) < 0$.

Likewise, we define the membership of $\mathbf{x}$ to the decision function $D_i(\mathbf{x})$ for $i \in L_n - L_{k_\mathrm{e}}$ by

$$m_{ki}(\mathbf{x}) = \min(1, -D_i(\mathbf{x})) \text{ for } i \in L_n - L_{k_\mathrm{e}}. \tag{10}$$

If $D_i(\mathbf{x}) < 0$, $\mathbf{x}$ is on the negative side of $D_i(\mathbf{x}) = 0$, and if $D_i(\mathbf{x}) < -1$, $m_{ki}(\mathbf{x}) = 1$. We also allow negative degree of membership if $D_i(\mathbf{x}) > 0$, namely, $\mathbf{x}$ is on the positive side of the hyperplane.

We consider the membership of $\mathbf{x}$ to R_k as the minimum of $m_{ki}(\mathbf{x})$ as follows:

$$m_k(\mathbf{x}) = \min_{i=1,\ldots,o} m_{ki}(\mathbf{x}). \tag{11}$$

The above membership function means that the degree of membership of $\mathbf{x}$ to R_k is measured by the nearest hyperplane from $\mathbf{x}$ among n separating hyperplanes. If the value is positive, $\mathbf{x}$ is in R_k and if negative, it is outside of R_k. Thus, if $m_k(\mathbf{x}) > 0$, $m_j(\mathbf{x}) < 0$ $(j \neq k, j = 1, \ldots, o)$. Therefore, we classify $\mathbf{x}$ into class k.

If all the membership functions are negative, namely, $\mathbf{x}$ is outside of any R_k $(k = 1, \ldots, o)$, we classify $\mathbf{x}$ into the class with the maximum membership value:

$$\arg \max_{i=1,\ldots,o} m_i(\mathbf{x}). \tag{12}$$

This means that $\mathbf{x}$ is classified into the class with the nearest R_k.

Classification by (11) and (12) is more complicated than classification by (7). Therefore, in the following we consider improving classification by (11) and (12) using (7).

We consider the following three cases:

Case 1: $L_c = L_k$ for $k \in \{1, \ldots, o\}$.

Sample $\mathbf{x}$ is in R_k. Thus we classify $\mathbf{x}$ into class k.

Case 2: $L_c \neq L_k$ for $k = 1, \ldots, o$.

Because L_c does not match any multilabel in the training data set, L_c is an undefined multilabel. Thus, we need to find R_k nearest to $\mathbf{x}$. To search for such R_k, we impose that $L_k \cap L_c \neq \emptyset$, namely, R_k and R_c overlap. In addition, we allow a single-label class k as a candidate if $k \in L_c$ and no multilabel including k exists in the target labels in the training data set. Then the candidate class set F_c is defined by

$$\begin{aligned} F_c \;=\; & \{k \,|\, k \in \{1, \ldots, n\} \text{ for } k \notin L_j, j = n+1, \ldots, o \\ & \text{or } L_k \cap L_c \neq \emptyset \text{ for } k = n+1, \ldots, o\}. \end{aligned} \tag{13}$$

Then (12) is reduced to

$$\arg \max_{k \in F_c} m_k(\mathbf{x}). \tag{14}$$

By (14), $\mathbf{x}$ is classified into the single-label or multilabel class whose associated region is nearest to $\mathbf{x}$.

Now we consider calculating $m_k(\mathbf{x})$. Because $D_j(\mathbf{x}) > 0$ for $j \in L_k \cap L_c$, we need not consider them. For $j \in L_c - L_k$, $D_k(\mathbf{x}) > 0$. Because $\mathbf{x}$ is outside of R_k,

$$m_{kj}(\mathbf{x}) = -D_k(\mathbf{x}). \tag{15}$$

For $j \in L_k - L_c$, $\mathbf{x}$ is outside of R_c. Thus,

$$m_{kj}(\mathbf{x}) = D_k(\mathbf{x}). \tag{16}$$

Thus, the calculation of $m_k(\mathbf{x})$ is simplified as follows:

$$m_k(\mathbf{x}) = \min_{j \in L_k \bigoplus L_c} m_{kj}(\mathbf{x}), \tag{17}$$

where $\bigoplus$ is the exclusive-or operator.

Case 3: $L_c = \emptyset$.

Sample $\mathbf{x}$ is in the unclassifiable region. Thus, we can classify $\mathbf{x}$ using (12). In the following we consider simplifying (12). The unclassifiable region is defined by $D_i(\mathbf{x}) < 0$ for $i = 1, \ldots, n$ and each separating hyperplane $D_i(\mathbf{x}) = 0$ separates the unclassifiable region from the corresponding adjacent region. Now if we delete the hyperplane $D_i(\mathbf{x}) = 0$, the two regions are combined into one. Therefore, the adjacent region is expressed by $D_i(\mathbf{x}) > 0$, $D_j(\mathbf{x}) < 0$ for $j \neq i, j = 1, \ldots, n$, namely, the adjacent region is R_i. This is the same situation with that for single-label one-against-all classification. Therefore, $\mathbf{x}$ is classified into class

$$\arg\max_{i \in L_n} D_i(\mathbf{x}). \tag{18}$$

The above classification rule is the same as that discussed in [6, 9]. Thus, our method explains the validity of the heuristics from the standpoint of fuzzy logic.

Now by the fuzzy SVM, the filled-circle sample in the undefined region shown in Fig. 3, is classified into Class 1, which is nearer, and the unclassifiable region is resolved and the dotted line shows the class boundary.

Consider the filled-circle sample in the undefined multilabel class $\{1, 3, 4\}$ shown in Fig. 4. The candidate multilabel classes are 1, 1&3, 1&4, and 2&4. Calculating the membership functions, we find that the sample is nearest to the region for Class 1&3. Thus, it is classified into the multilabel class 1&3. The dotted lines in the figure are decision boundaries.

5. Computer Experiments

We evaluated the proposed FSVM using the 12 benchmark data sets downloaded from [25, 26]. They are from biology, multimedia, and text categorization and are widely used for evaluating multilabel classification methods. Eleven of the data sets were extensively used in [2] to compare 12 multilabel classification methods: problem transformation methods and

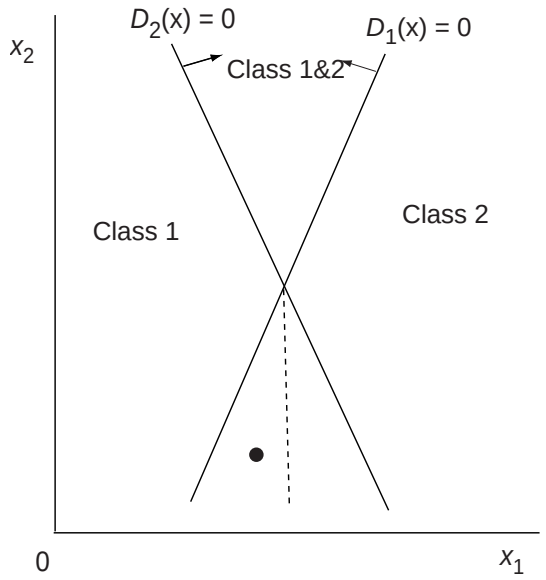

Figure 3: Resolving an unclassifiable region by one-against-all classification

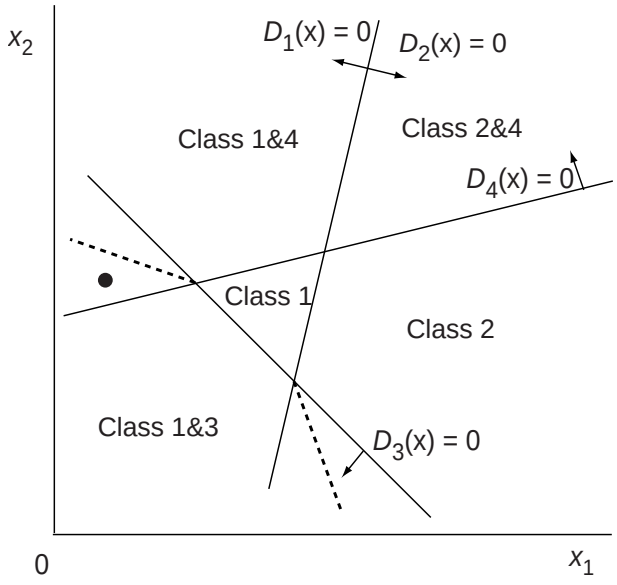

Figure 4: Resolving undefined multilabel regions by one-against-all classification

ensemble methods based on SVMs and decision trees. For comparison with the FSVM, we used the one-against-all SVM and the one-against-one SVM. We also compare performance of the FSVM with the best performance among 12 methods shown in [2]

The accuracy is one of the most important measures for single-label classification, and is also very important for multilabel classification. The accuracy A is defined by

$$A = \frac{1}{M}\sum_{i=1}^{M}\frac{|P_i \cap T_i|}{|P_i \cup T_i|}, \tag{19}$$

where P_i is the set of predicted labels, T_i is the set of target labels, and M is the number of training data.

But because multilabel classification is usually very difficult to classify and thus the accuracies are usually low, several other measures are considered. One such measure is the subset accuracy (exact match ratio) A_{S} defined by

$$A_{\mathrm{S}} = \frac{1}{M}\sum_{i=1}^{M} I(P_i = T_i), \tag{20}$$

where $I(P_i = T_i)$ is 1 when $P_i = T_i$ and 0, otherwise. For single-label classification, the accuracy and the subset accuracy are the same and reduce to the conventional accuracy.

Other measures include the microaverage F-measure and macroaverage F-measure. In our preliminary study, we evaluated the above four measures and found that they behave similarly: in almost all cases the best performance for each measure was obtained for the same parameter conditions. Therefore, in the following study, we only show the results using the accuracy and subset accuracy.

We implemented the proposed FSVM into the multilabel SVM classifier code, binary.py, downloaded from [27]. The multilabel processing is written in Python and LIBSVM [26] is used as an SVM tool. We modified the Python code and also the LIBSVM code written in c/c++. We also downloaded trans_class.py that converts multilabel classification into single-label classification and coded the one-against-one SVM for multilabel classification.

We used the L1 SVM with RBF kernels given by (6). We determined the γ value and the value of the margin parameter C, which controls the trade-off between the classification error and the generalization ability by

fivefold cross-validation. For the one-against-one SVM, we used conventional cross-validation for multiclass classification. But for the one-against-all and FSVMs, we carried out cross-validation for each binary classifier and took the average of cross-validation accuracies. This is a simplified measure but as will be shown later, it worked very well.

We selected the C value from $\{1, 2, \ldots, 2^{10}\}$. To speedup cross-validation we assumed that the optimum γ value is near $1/m$ and that around the optimum value the cross-validation error curve is convex for the change of the γ value. According to the assumption, first we carried out cross-validation with $\gamma = 1/m$ and $C = 1$ to 1024. Then we carried out cross-validation selecting the γ value that is near $1/m$ from $\{10^{-5}, 5 \times 10^{-4}, 10^{-4}, \ldots, 1, 5, 10\}$. We iterated cross-validation until the γ value for the minimum cross-validation error is within the minimum and the maximum γ values tested. Because cross-validation of the bookmarks data set listed in Table 3, which will be explained immediately, was very slow, we only carried out cross-validation for $\gamma = \{0.001, 0.000465, 0.001, 0.005\}$ and $C = \{1, 32, 1024\}$.

In [2], the γ value was selected from $2^{-15} = 3.05^{-5}, 2^{-13}, \ldots, 2^{1}, 2^{3}$, and the C value from $2^{-5}, 2^{-3}, \ldots, 2^{13}, 2^{15}$. Therefore, the ranges of the γ value of the both methods are comparable, but the range of the C value in [2] is wider but the increment was twice as large as that in our experiment.

Table 3 lists the data set specifications and the determined parameter values for 12 data sets. Because unlabeled data were included in the delicious and bookmarks data sets, we assigned a new label to them. For mediamill, delicious, and bookmarks data sets, we could not carry out cross-validation for the one-against-one SVM, because training was terminated because of errors. From the table, the parameter values determined by cross-validation for one-against-all (OAA) and one-against-one (OAO) were usually very similar.

Table 4 shows the accuracy, the subset accuracy, and elapsed time in training the SVM with the determined parameter values using the training data and classifying the test data. We used a windows 7 machine with 3.4 GHz processors and 16 GB memory. "FSVM_c" denotes the result that only the unclassifiable regions are resolved. We show the best (subset) accuracy among the FSVM, FSVM_c, and OAA in bold face. And "Best" shows the best value listed in [2] among 12 methods including the binary relevance method (the one-against-all SVM) [1], the classifier chaining method [8], the HOMER (hierarchy of multi-label classifiers) method [35], multi-label C4.5 [4], and ensembles of classifier chains [8]. Accuracies of some of the benchmark data sets are also shown in [8]. Comparing their best accuracies

Table 3: Data set specifications and parameter values

Data	Inputs	Classes	Train	Test	γ value		C value	
					OAA	OAO	OAA	OAO
emotions [28]	72	6	391	202	0.00005	0.00005	1024	1024
scene [6]	294	6	1211	1196	0.1	0.1	4	2
yeast [3]	103	14	1500	917	1	1	2	2
medical [29]	1449	45	333	645	0.005	0.01	64	32
enron [30]	1001	53	1123	579	0.01	0.0005	2	128
corel5k [31]	499	374	4500	500	0.1	0.05	2	4
tmc2007 [32]	30438	22	21519	7077	0.5	0.01	2	64
rcv1v2 [33]	47236	102	3000	300	0.005	0.001	256	1024
mediamill	120	102	30993	12914	10	—	4	—
bibtex [34]	1836	159	4880	2515	0.005	0.0001	8	1024
delicious [35]	500	984	12920	3185	0.05	—	4	—
bookmarks [34]	2150	208	60000	27856	0.0001	—	32	—

with those in Table 4, the accuracy of the medical data set in [8] is 0.7721, and is higher but others are lower. For the tmc2007 data set, the accuracy is 0.5492, which is much lower and is near to that by OAA.

Comparing the FSVM, FSVM_c, and OAA, OAA shows the worst (subset) accuracy. In most cases the FSVM shows better accuracy than the FSVM_c and this tendency is more evident for the subset accuracy. The reason why sometimes the FSVM_c performs better than the FSVM is that resolution of undefined labels sometimes fails as will be explained later.

In most cases, the (subset) accuracies of OAO are better than those of OAA but those of the FSVM and OAO are comparable. So are those of the FSVM and Best.

The time for training and classification by OAO was the shortest. This is because the number of samples per binary classifier for OAO is much smaller than that for OAA, although the number of binary classifiers is larger. The time difference between the FSVM and OAA is caused by calculations of membership functions. Python is an interpretive list processing language and thus if the code is rewritten by c/c++, the overhead caused by membership calculations will be reduced.

Using the scene and medical data sets, we analyzed the behavior of the

Table 4: Evaluation results

Data	Accuracy					Subset Accuracy					Time (s)		
	FSVM	$FSVM_c$	OAA	OAO	Best	FSVM	$FSVM_c$	OAA	OAO	Best	FSVM	OAA	OAO
emotions	0.5351	**0.5363**	0.5025	0.5488	0.536	**0.2772**	**0.2772**	**0.2772**	0.3267	0.307	3.9	2.6	0.8
scene	**0.7699**	0.7678	0.6888	0.7701	0.735	**0.7224**	0.7166	0.6421	0.7316	0.694	9.9	7.2	2.3
yeast	**0.5208**	0.5190	0.5172	0.5503	0.559	**0.2246**	0.1985	0.1985	0.2737	0.239	17.6	12.7	2.8
medical	0.7822	**0.7893**	0.7056	0.7311	0.730	0.7008	**0.7039**	0.6264	0.6512	0.646	13.5	12.2	0.5
enron	**0.4613**	0.4479	0.4388	0.4148	0.478	**0.1675**	0.1399	0.1382	0.1710	0.149	40.7	37.5	6.0
corel5k	**0.1354**	0.1323	0.0568	0.1572	0.195	**0.0060**	**0.0060**	0.0020	0.0600	0.012	1160.0	992.3	149.7
tmc2007	0.5896	**0.5897**	0.5593	0.5538	0.914	**0.3510**	**0.3510**	0.3249	0.3442	0.816	4555.2	4203.2	1857.3
rcv1v2	**0.6668**	0.6533	0.6468	0.6748	—	**0.4433**	0.4107	0.4103	0.5163	—	272.3	212.7	54.8
mediamill	**0.4739**	0.4711	0.4671	—	0.441	**0.1415**	0.1370	0.1347	—	0.122	13921.7	12785.4	—
bibtex	**0.3869**	0.3847	0.3050	0.3451	0.352	**0.2231**	0.2195	0.1722	0.2243	0.202	445.4	370.4	236.9
delicious	**0.1702**	0.1557	0.1495	—	0.207	**0.0163**	0.0100	0.0097	—	0.018	8669.3	7156.2	—
bookmarks	**0.3401**	0.3400	0.2058	—	0.237	**0.2713**	0.2712	0.1867	—	0.209	80228.2	71019.3	—

FSVM. Figure 5 shows the accuracy change of the scene data set for the change of the γ value. For each γ value, the C value was set to the value selected by cross-validation. From the figure, the accuracy of the FSVM is higher than or equal to that of the $FSVM_c$. The accuracies of the FSVM, $FSVM_c$, and OAO around the peak accuracies are almost the same, but those of OAA are always much lower.

Figure 6 shows the subset accuracy change of the scene data set for the change of the γ value. The tendency is the same as that of the accuracy but difference between the subset accuracies of the FSVM and those of the $FSVM_c$ are much clearer.

Table 5 lists the results of the data that caused different (subset) accuracies between the FSVM and $FSVM_c$ for the parameter values as listed in Table 3. In the table, the "Predicted" column shows the predicted label set by OAA, "Modified", the modified label set by the FSVM, and "Target", the target label set. In the "Explanation" column, whether the accuracy is improved or not is explained. For example, for the 169th sample, the label set {0, 3} is changed to {0, 5}, which is exactly the same with the target label set. For the 565th sample, because the label set {1, 2} was changed to 1, which is not equal to the target label of 2, the accuracy was degraded. Accordingly, the number of exact matches increased by 7 by the FSVM.

Figure 7 shows the accuracy change of the medical data set for the change of the γ value. The accuracy for the $FSVM_c$ shows the best among the four methods, that for the FSVM, second best, and OAO, the worst. Unlike the scene data set, the accuracies of OAA were not so good compared to those of the FSVM and $FSVM_c$.

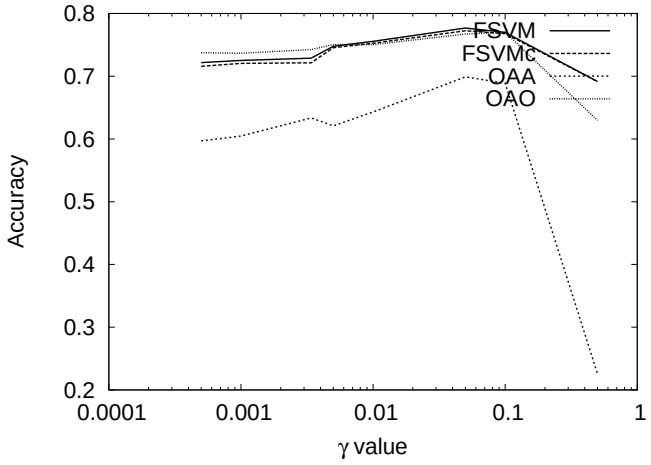

Figure 5: Accuracies for the scene data set

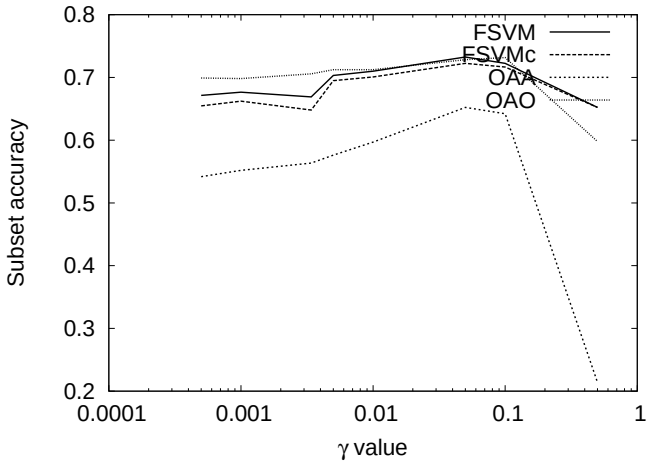

Figure 6: Subset accuracies for the scene data set

Table 5: Classification analysis for the scene data set

Sample	Predicted	Modified	Target	Explanation
169	0, 3	0, 5	0, 5	Exact match
229	1, 4	1	1	Exact match
242	1, 4	1	1	Exact match
283	1, 4	1	1	Exact match
328	1, 2	1	1	Exact match
339	1, 5	1	1	Exact match
565	1, 2	1	2	Degraded
668	0, 3	0, 4	3	Degraded
987	0, 3, 4	3, 4	3, 4	Exact match

Figure 8 shows the subset accuracy change for the change of the γ value. The tendency is similar to that in Fig. 7.

Now we analyze why the FSVM_c performed better than the FSVM. Table 6 lists the samples in which the classification results are different between the FSVM and FSVM_c for the determined parameter values. In the "Explanation" column, "Not defined" means that the target label set is not included in the label sets of the training data set. For example, for the 194th sample, the predicted label set of {32,44} was equal to the target label set. But because it is not included in the label sets of the training data set, the FSVM found the nearest label set of {3, 32, 34}, which resulted in mismatch. Accordingly, the number of exact matches is five but the number of mismatches is seven. Thus, the number of exact matches decreased by two by the FSVM. This means that to improve the generalization ability by the FSVM, possible label sets need to be defined in the training data set.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number 25420438.

6. Conclusions

In this paper we proposed fuzzy SVMs (FSVMs) for multilabel classification. Using the decision functions obtained by training the one-against-all SVM, we define a membership function for each label set included in the

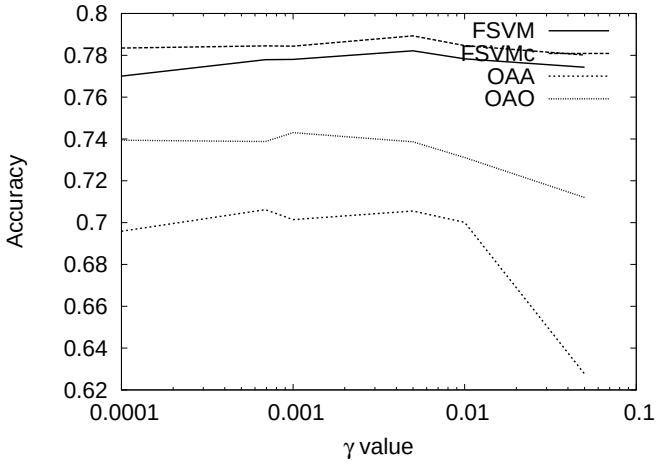

Figure 7: Accuracies for the medical data set

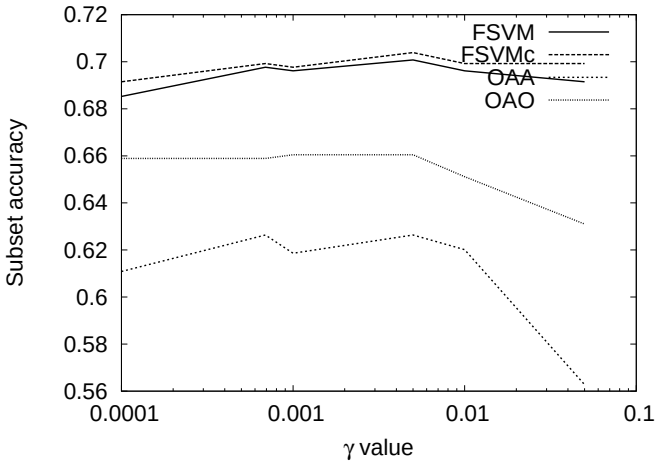

Figure 8: Subset accuracies for the medical data set

Table 6: Classification analysis for the medical data set

Sample	Predicted	Modified	Target	Explanation
19	4, 23	23	23	Exact match
71	4, 31	4, 44	31	Degraded
143	4, 31, 44	4, 44	4, 44	Exact match
161	32, 44	32, 34	44	Degraded
194	32, 44	3, 32, 34	32, 44	Not defined
224	9, 32	32, 34	9	Degraded
273	4, 32, 34	4, 34	4, 32, 34	Not defined
290	24, 41	36, 41	24, 41	Not defined
328	24, 38	38	38	Exact match
338	4, 32, 44	4, 32	4, 32, 44	Not defined
346	0, 36, 41	0, 41	0, 41	Exact match
349	9, 32	32, 34	9	Degraded
355	38, 43	38	38, 43	Not defined
360	32, 44	32, 34	32, 44	Not defined
367	9, 38	38	36, 38	Improved
379	31, 44	4, 44	4, 44	Exact match
421	0, 24	0, 11, 41	24, 41	Degraded
427	0, 38	38	0	Degraded
433	31, 32	10, 31, 44	32	Degraded
451	23, 32	23	32	Degraded
463	4, 31	4, 32	4, 32, 44	Improved
567	23, 32	32, 34	23, 32	Not defined

training data set. For a given test sample, we classify the sample into the multilabel associated with the largest membership. By the FSVM, we can resolve unclassifiable regions and classification to undefined label set that occur in the conventional one-against-all SVM. We also show that resolution of unclassifiable regions results in classifying a sample into the class associated with the maximum decision function. This is a heuristic to resolve unclassifiable regions. By computer experiments using 12 benchmark data sets, we showed that the accuracies were improved by the FSVM over the conventional one-against-all SVM and, in most cases, over the one-against-all SVM with the heuristic to resolve unclassifiable regions.

References

[1] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

[2] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9):3084–3104, 2012.

[3] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 681–687. MIT Press, 2002.

[4] A. Clare and R. D. King. Knowledge discovery in multi-label phenotype data. In L. De Raedt and A. Siebes, editors, *Principles of Data Mining and Knowledge Discovery*, volume 2168 of *Lecture Notes in Computer Science*, pages 42–53. Springer Berlin Heidelberg, 2001.

[5] R. E. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2):135–168, 2000.

[6] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[7] G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In J. N. Kok, J. Koronacki, R. L. Mantaras, S. Matwin, D. Mladenič, and A. Skowron, editors, *Machine Learning: ECML 2007*, volume 4701 of *Lecture Notes in Computer Science*, pages 406–417. Springer Berlin Heidelberg, 2007.

[8] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.

[9] J. Xu. An extended one-versus-rest support vector machine for multi-label classification. *Neurocomputing*, 74(17):3114–3124, 2011.

[10] H. Laanaya, A. Martin, D. Aboutajdine, and A. Khenchaf. Credal classification rule for uncertain data based on belief functions. *Information Fusion*, 11(4):338–350, 2010.

[11] Z. Liu, Q. Pan, J. Dezert, and G. Mercier. Credal classification rule for uncertain data based on belief functions. *Pattern Recognition*, 47(7):2532–25415, 2014.

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

[13] P. K. Simpson. Fuzzy min-max neural networks—Part 1: Classification. *IEEE Transactions on Neural Networks*, 3(5):776–786, 1992.

[14] S. Abe and M.-S. Lan. A method for fuzzy rules extraction directly from numerical data and its application to pattern classification. *IEEE Transactions on Fuzzy Systems*, 3(1):18–28, 1995.

[15] S. Abe and R. Thawonmas. A fuzzy classifier with ellipsoidal regions. *IEEE Transactions on Fuzzy Systems*, 5(3):358–368, 1997.

[16] K. Kaieda and S. Abe. KPCA-based training of a kernel fuzzy classifier with ellipsoidal regions. *International Journal of Approximate Reasoning*, 37(3):189–217, 2004.

[17] M.-C. Su, C.-H. Chou, E. Lai, and J. Lee. A new approach to fuzzy classifier systems and its application in self-generating neuro-fuzzy systems. *Neurocomputing*, 69(4–6):586–614, 2006.

[18] V. Uebele, S. Abe, and M.-S. Lan. A neural network–based fuzzy classifier. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(2):353–361, 1995.

[19] 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, Washington, DC, 2001.

[20] S. Abe and T. Inoue. Fuzzy support vector machines for multiclass problems. In *Proceedings of the Tenth European Symposium on Artificial Neural Networks (ESANN 2002)*, pages 113–118, Bruges, Belgium, 2002.

[21] D. Tsujinishi and S. Abe. Fuzzy least squares support vector machines for multiclass problems. *Neural Networks*, 16(5–6):785–792, 2003.

[22] T. Kikuchi and S. Abe. Comparison between error correcting output codes and fuzzy support vector machines. *Pattern Recognition Letters*, 26(12):1937–1945, 2005.

[23] Tai-Yue Wang and Huei-Min Chiang. Solving multi-label text categorization problem using support vector machine approach with membership function. *Neurocomputing*, 74(17):3682–3689, 2011.

[24] S. Abe. *Support Vector Machines for Pattern Classification*. Springer-Verlag, London, UK, second edition, 2010.

[25] Mulan: A Java Library for Multi-Label Learning. http://mulan.sourceforge.net/datasets.html.

[26] C.-C. Chang and C.-J. Lin. LIBSVM–A library for support vector machines: http://www.csie.ntu.edu.tw/˜cjlin/libsvm/.

[27] LIBSVM tools: Multi-label classification: http://www.csie.ntu.edu.tw/˜cjlin/libsvmtools/multilabel/.

[28] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. Multi-label classification of music into emotions. In *Proceedings of Ninth International Conference on Music Information Retrieval (ISMIR 2008)*, pages 325–330, Philadelphia, PA, USA, September 2008.

[29] J. P. Pestian, C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, and W. Duch. A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing (BioNLP '07)*, pages 97–104, Stroudsburg, PA, USA, June 2007.

[30] B. Klimt and Y. Yang. The enron corpus: a new dataset for email classification research. In *Proceedings of the 15th European conference on Machine Learning (ECML 2004)*, pages 217–226, Pisa, Italy, September 2004.

[31] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of Seventh European Conference on Computer Vision*, volume IV, pages 97–112, Copenhagen, Denmark, May, June 2002.

[32] A. Srivastava and B. Zane-Ulman. Discovering recurring anomalies in text reports regarding complex space systems. In *Proceedings of 2005 IEEE Aerospace Conference*, pages 3853–3862, Big Sky, MT, USA, March 2005.

[33] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, (5):361–397, 2004.

[34] I. Katakis, G. Tsoumakas, and I. Vlahavas. Multilabel text classification for automated tag suggestion. In *Proceedings of ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, Antwerp, Belgium, October 2008.

[35] G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proceedings of ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, pages 30–44, Antwerp, Belgium, October 2008.