Multi-Category Classification Based on ECOC Approach Using Sub-categories

Leona Suzuki†
Graduate School of Creative Science and Engineering, Waseda University.
3-4-1, Okubo, Shinjuku-ku, Tokyo 169-8555, JAPAN
Tel:(+81) 3-5286-3290, Email: davinci.1031@suou.waseda.jp

Haruka Yamashita ††, Masayuki Goto††
School of Creative Science and Engineering, Waseda University.
3-4-1, Okubo, Shinjuku-ku, Tokyo 169-8555, JAPAN
Tel:(+81) 3-5286-3290, Email: h.yamashita@aoni.waseda.jp, masagoto@waseda.jp

Abstract. Due to the development of information technology, it is recently available to use a huge number of document data accumulated on various databases. Because the data size is usually enormous, automatic multi-category classification becomes more important. We focus on multi-category classification based on Error-Correcting Output Codes (ECOC) approach which combines plural binary classifiers. The ECOC approach consists of two steps: coding and decoding. In the step of coding, a numerical table called a code table whose row represents each category and column represents configuration of each binary classifier is generated. In the decoding step, the category of new input data is predicted by integrating outputs of all binary classifiers designed by the code table.
In this study, in order to enhance classification accuracy, we improve both steps of the ECOC method, i.e. coding and decoding. For the coding step, we introduce the code tables considering sub-categories of each original category. For the decoding step, we propose a new decoding method which is suitable for code tables based on the sub-category setting. We verify the effectiveness of our proposed method by conducting classification experiments with actual document data.

Keywords: ECOC, multi-category classification, text data

1. INTRODUCTION

Due to the development of information technology, a huge number of document data has been accumulated on various databases in several levels of business processes. In the field of industrial management, various kinds of digital document data are treated for many purposes and the analysis of the text data can show new findings for management purposes. Because the number of such digital text data is enormous, the technology of the automatic multi-valued classification becomes more important. The multi-valued classification is the classification problem on which the size of category labels is more than 3. Usually, the documents are classified into one in many categories, so that the multi-valued classification task is necessary in document classification. There are two approaches for dealing with multi-valued classification problems. First one is to construct a unique multi-valued classifier, and second one is to formulate a combination of binary classifiers (Dietterich & Bakiri.1995, Ikeada.2010, Huang *et al*. 2006). We focus on the multi-valued classification based on Error-Correcting Output Codes (ECOC) approach which is one of the effective methods combining several binary classifiers (Dietterich & Bakiri. 1995). The ECOC approach consists of two steps: coding and decoding steps. In the step of coding, a numerical table called a *code table* whose row represents each category and column represents configuration of each binary classifier is generated. In the decoding step, the category of a new input data is predicted by integrating outputs of all binary classifiers designed by the code table.

There are two approaches for constructing a code table. First one is the adaptive coding method (Pujol *et al.* 2006, Escalera *et al.* 2008, Zhong & Cheriet. 2013, Zhang. 2015,) and second one is the non-adaptive method (Oyama *et al*. 2008, Ogihara *et al*. 2013). The former approach generates a code table adoptively with learning the training data. The latter

generates a code table firstly and the table is fixed before the learning of binary classifiers. This means that the configuration of all binary classifiers can be obtained before the training data is given. Therefore, this method has the advantage of possibility of parallel computation for each binary classifier. In this study, we focus on the code table construction of the latter approach.

On the other hand, the methods considering sub-categories that are divided from an original set of categories are proposed (Pujol *et al.* 2008, Bouzas *et al.* 2010, Zhang 2015). Dividing categories into sub-categories makes the binary classification for each classifier easy. However, most of these methods adaptively divide into sub-categories, and these methods have the limitation that the data belonging to the same category are not divided into different sub-categories so that these are not classified by a binary classifier. This limitation disturbs the configuration of various binary classifiers.

In this study, in order to enhance classification accuracy, we improve both steps of the ECOC method, i.e. coding and decoding methods. For the coding step, we propose a new non-adaptive coding methods considering sub-categories of each original category and overcoming the limitation that the sub-categories belonging to the same category are not classified. For the decoding step, we propose a new decoding method which is suitable for code tables considering the sub-category. We verify the effectiveness of our proposed method by conducting classification experiments with actual document data.

2. PRELIMINARIES

2.1 Classification using Relevance Vector Machine

On the ECOC approach, it is possible to choose various binary classifiers. For example, the support vector machine can be used as a strong classification model. In this study, the Relevance Vector Machine (RVM) is applied to make a binary classifier used to the ECOC method.

The RVM is the method based on the probabilistic approach for solving regression and classification (Tipping. 2001). Here we describe the RVM method based on (Tipping. 2001). In the classification problem, the RVM whose output means the belonging probability to a category has the relatively high accuracy of category prediction. Let $\boldsymbol{x}$ be a feature vector on the defined feature space and $c \in \{c_1, c_2\}$ be a binary category label. A set of N training document samples is denoted by $\{(\boldsymbol{x}_n, t_n)\}_{n=1}^{N} (\boldsymbol{x}_n = (x_{n,1}, x_{n,2}, \dots, x_{n,d}))$, where $t_n \in \{c_1, c_2\}$. The probability of category label $c_k (k = 1,2)$ conditioned by $\boldsymbol{x}$ is expressed by using logistic regression as follows.

$$p(c = c_k|\boldsymbol{x}) = \frac{1}{1 + \exp(-f(\boldsymbol{x}))}, \qquad (1)$$

$$f(\boldsymbol{x}) = \sum_{i=1, t_i = c_k}^{N} a_i K(\boldsymbol{x}, \boldsymbol{x}_i), \qquad (2)$$

Where γ_i denotes the variance parameter of the weight parameter $a_i \sim N(0, \gamma_i^{-1})$, and $\gamma_1^{-1}, \gamma_2^{-1}, \cdots, \gamma_N^{-1}$ are obtained by maximizing a posterior probability of $\boldsymbol{\gamma} = (\gamma_1^{-1}, \gamma_2^{-1}, \cdots, \gamma_N^{-1})$. $K(\cdot,\cdot)$ is a kernel function which calculates the inner product of two input data mapping to a higher dimensional space. By maximizing a posterior probability, almost all γ_i^{-1} become 0. We call that $\boldsymbol{x}_i$ having non-zero γ_i^{-1} value is a relevance vector (RV). The function $f(\boldsymbol{x})$ is decided by using the relevance vectors (RVs). The RVM has high performance of classification accuracy and other several desirable properties.

It should be noted that the RVM needs a lot of computational complexity to learn a set of training data. The computational complexity for the estimation is known to be $O(N^3)$ (Tipping. 2001). Here, let K be the number of categories. For a multi-valued classification by using a unique RVM classifier, there is a disadvantage that the computational complexity for learning is K^3 times larger than that of the binary RVM.

2.2 ECOC approach

In this study, we focus on the multi-valued classification problem on which the size of category labels is more than 3. Let K be the number of categories, that is, $\mathbb{C} = \{c_1, c_2, \dots, c_K\}$ is the set of category labels. The ECOC approach is a multi-valued classification method based on the error correcting technique in the field of code theory. This approach estimates the category $c_k (k = 1, \dots, K)$ of a new input data by combining the outputs of all binary classifiers. Each binary classifier classifies the data into a category set based on a code table which is configured with $\{0,1\}$, where the category set means a set of several categories which is used for binary classification on this binary classifier. Here, a code table is denoted by $\boldsymbol{W}$ where $\boldsymbol{W}$ is a $K \times R$ matrix and R is the number of binary classifiers. Each column vector of $\boldsymbol{W}$ represents the configuration of each binary classifier which defines the classification rule between the set of categories corresponding to 1 in the column vector of $\boldsymbol{W}$ and the set of categories corresponding to 0 in the same vector. Thereby the two columns that 0 and 1 are reversed are meaning an identical binary classifier. Moreover, the vector of i-th row is called the *code word* of the category c_i and the code word of c_i is denoted by $\boldsymbol{w}_i$. In the predicting step, the code words are compared with the outputs for the new data, and the category of new data is decided by most similar or closest code word in $\boldsymbol{W}$.

There is also a ternary code table which permits the existence of categories which are not used for a classification

in addition to the conventional binary code table which uses only $\{0,1\}$ (Escalera *et al.* 2010). Here, the categories which are not used for classification are represented by an asterisk symbol '$*$'. Let W_i^r be an r-th$(r = 1, \dots, R)$ element of a vector $\boldsymbol{w}_i$. If W_i^r is $*$, then the training data of category c_i is not used for learning binary classifier r. Hence, when a ternary code table with $\{0,1,*\}$ is used, the number of training data and the computational complexity can be decreased compared with a binary code table.

In general, the classification accuracy becomes high in the ECOC approach when the hamming distances between code words are large. The main reason of this property is that a code word is a vector corresponding to a category on the R-dimensional space. The two categories with large hamming distance are easily distinguished because the hamming distance represents differences between two vectors. Therefore, when the number of classifiers become large, the classification accuracy rises because of enlarging the hamming distance. However, the computational complexity increases at the same time.

In the following, we introduce the detail of the ECOC approach focusing on coding (section 2) and decoding methods (section 3).

3. CODING METHODS

The various kinds of methods for code table construction were proposed for the ECOC approach. In this section, we describe the methods for generating code table (we call methods "coding methods") which were proposed in past studies (e.g., Dietterich. & Bakiri. 1995, Crammer & Singer. 2002). As we mentioned above, there are mainly two approaches for coding methods. First one is adaptive methods and second one is non-adaptive methods. The former generates a code table adaptively with learning the binary classifiers. The latter generates a code table firstly and the table is fixed before the learning binary classifiers. In the following, we describe the adaptive methods in the section 3.1 and the non-adaptive methods are described in the section 3.2.

3.1. Adaptive coding methods

There are lots of adaptive methods. These methods construct the code tables without considering the hamming distances between code words, and considering the structure of the data adoptively. Crammer & Singer. (2002) proposed the method searching the code table whose norm is small with minimizing empirical loss. Another type of adaptive methods is based on tree structure. Pujol *et al.* (2006) proposed Discriminant ECOC, which is based on the discriminant tree structures. This method constructs a hierarchical code table which maximizes a discriminative criterion based on mutual information (Cover. 2012). It should be noted that, there are lots of methods of ECOC using this tree structure (Escalera *et al.* 2006, Escalera *et al.* 2007, Pujol *et al.* 2008, Xue *et al.* 2015). Moreover, the methods using sub-categories which are divided from original set of categories are proposed. (Pujol *et al.* 2008, Bouzas *et al.* 2010, Zhang 2015). In these methods, the sub-category information is used in order to simplify the discrimination on the binary classifiers. In this way, a lot of adaptive methods are proposed (Escalera *et al*. 2009, Escalera *et al*. 2011), however the methods generate code table adaptively with learning the classifiers. Therefore, this method cannot compute each binary classifier on parallel process and the computational complexity of learning become high.

3.2 Non-adaptive coding methods

In this study, we focus on non-adaptive coding methods because these methods enable the parallel computation for each binary classifier. In the following section, we show typical non-adaptive coding methods.

3.2.1 One-vs-the rest and one-vs-one

The one-vs-the rest and the one-vs-one are the standard non-adaptive methods. The one-vs-the rest learn the K binary classifiers where K denotes the number of categories. Each of the K classifiers classifies the data into a category or others. That is, it discriminates between the data belonging to a single category and others belonging to the remaining categories. The one-vs-one used the binary classifiers, which classify all of every conceivable pair of the categories.

For example, there is a set of category labels $\{c_1, c_2, c_3\}$. One-vs-the rest learns the three classifiers: $\{c_1\}$ vs $\{c_2, c_3\}$, $\{c_2\}$ vs $\{c_1, c_3\}$, and $\{c_3\}$ vs $\{c_1, c_2\}$. On the other hand, one-vs-one leans 3 classifiers: $\{c_1\}$ vs $\{c_2\}$, $\{c_2\}$ vs $\{c_3\}$, and $\{c_3\}$ vs $\{c_1\}$. Although, these methods are so simple, it is known that these methods are effective (Rifkin. & Klautau. 2004).

3.2.2 Random code

Generating a code table by random coding, the set of category labels is split into two category sets at random under the certain rules. The random codes do not have the guarantee that the codes have high performance because these are based on random strategy. Therefore, these codes are used for benchmark methods or initial matrix of code tables' coding method by iterative processing (Chmielnicki. 2015).

There are two types of random codes; dense random codes and sparse random codes (Allwein. 2001). The first step of coding of the dense random is making high number of binary random code tables whose elements of $\{0,1\}$ chosen at random. The code tables have $\lceil 10\log_2 K \rceil$ columns, where $\lceil x \rceil$ is the smallest integer not less than x. The second step is

that choosing the code table from these binary random code tables which has the highest minimum hamming distance between code words.

The coding of sparse random is the same to the coding of dense random. The sparse random codes have three elements $\{0,1,*\}$ and the elements are designed at random. Each element in the sparse random codes is $*$ with probability 0.5 and 0 or 1 with probability 0.25, and the code tables have $\lceil 15\log_2 K \rceil$ binary classifiers.

3.2.3 BCH code

The BCH code is one of the effective codes which are known in the field of coding theory (Cover. & Thomas. 2012). This code is generated uniquely based on two factors; error correction capability and code length. Here, ν, κ, τ denote the code word length, the number of information bits and the number of error correction bits respectively. The BCH code with these parameters is denoted by (ν, κ, τ). When the BCH code is used for the ECOC approach, ν and $2\tau + 1$ correspond to R and the minimum hamming distance among the code words. Therefore, the minimum hamming distance can be set freely in the BCH code.

As mentioned above, in the ECOC approach, two categories with large hamming distance is more easily distinguishable. Therefore, the BCH code is suitable for creating a code table because the minimum hamming distance of the code words can be set freely. Table 1 is the example of $(15, 5, 3)$ BCH code in case of $K = 8$.

Table 1. $(15, 5, 3)$ BCH code ($K = 8$)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
c_1	1	0	1	0	0	1	1	0	1	1	1	0	0	0	0
c_2	0	1	0	1	0	0	1	1	0	1	1	1	0	0	0
c_3	0	0	1	0	1	0	0	1	1	0	1	1	1	0	0
c_4	0	0	0	1	0	1	0	0	1	1	0	1	1	1	0
c_5	0	0	0	0	1	0	1	0	0	1	1	0	1	1	1
c_6	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0
c_7	1	0	0	0	1	1	1	1	0	1	0	1	1	0	0
c_8	1	0	1	1	0	0	1	0	0	0	1	1	1	1	0

3.2.4 Exhaustive code

The exhaustive code is a binary code table containing all of every conceivable binary classifier (Dietterich. & Bakiri. 1995). The configuration method of the exhaustive code is given as follows. Here, $\mathbf{0}_L = (0,0,\ldots,0)$ denotes the vector whose the number of elements "0" is L. By the same manner to $\mathbf{0}_L$, $\mathbf{1}_L = (1,1,\ldots,1)$ is the vector whose the number of elements "1" is L.

(1) All elements of the vector $\boldsymbol{w}_1$ are set as $\boldsymbol{w}_1 = 1$.
(2) The vector $\boldsymbol{w}_2$ is set as $\boldsymbol{w}_2 = (\mathbf{0}_{2^{K-2}}, \mathbf{1}_{2^{K-2}-1})$.
(3) The vector $\boldsymbol{w}_3$ is set as $\boldsymbol{w}_3 = (\mathbf{0}_{2^{K-3}}, \mathbf{1}_{2^{K-3}}, \mathbf{0}_{2^{K-3}}, \mathbf{1}_{2^{K-3}-1})$.
(4) Similar to the above, in $\boldsymbol{w}_k$, there are alternating runs of 2^{K-k} zeros and ones.

Table 2 is the example of the exhaustive code in case of $K = 4$. The number of binary classifiers R is $2^{4-1} - 1 = 7$. Because the number of classifiers is the highest in all binary code tables, the classification performance is relatively high for a given category number $K = 4$. However, because of the number of classifiers is large, the computational cost is the highest of the ECOC approach.

Table 2. Exhaustive code $(K = 4)$

	1	2	3	4	5	6	7
c_1	1	1	1	1	1	1	1
c_2	0	0	0	0	1	1	1
c_3	0	0	1	1	0	0	1
c_4	0	1	0	1	0	1	0

4. DECODING METHODS

For deciding a predicted category of a new input data, we plug in the new data to the leaned classifiers and create a binary vector based on the output of classifiers. Then, we decide the estimated category to which the new data belongs. In this section, we describe the two decoding methods which were proposed in past studies. Section 4.1 describes the Hamming decoding (Dietterich. & Bakiri. 1995), and Section 4.2 describes the probabilistic-based decoding (Smith. & Windeatt. 2005)

4.1. Hamming decoding

The most generalized decoding method is the hamming decoding. The method is used when the outputs of binary classifiers are hard decision. Then, the hamming distances between the code words and outputs of binary classifiers are calculated, and the category of the new data is decided by the code word which has the smallest distance. Let G_r be the output of r-th binary classifier, the classification criterion is defined as follows.

$$\hat{k} = \arg\min_{i} \sum_{r=1}^{R} d_H(G_r, W_i^r) \quad (3)$$

$$d_H(G_r, W_i^r) = \begin{cases} 0, & \text{if } G_r = W_i^r \\ 1, & \text{if } G_r \neq W_i^r \end{cases} \quad (4)$$

4.2. Probabilistic-based decoding

The probabilistic-based decoding method uses the belonging probability to each category set. As mentioned above, in this study, we focus on the Relevance Vector Machine (RVM) as binary classifiers. The output of RVM means the belonging probability to category set, where the category set is described as {0,1}, written in the section 2.2. Therefore, the classification criterion based on belonging probability is defined as follows.

$$\hat{k} = \arg\max_{i} \prod_{r=1}^{R} G_r^{W_i^r} (1 - G_r)^{1 - W_i^r} \quad (5)$$

From above discussion, both the hamming decoding and the probabilistic-based decoding are the prediction rules selecting one code word $\boldsymbol{w}_{\hat{k}}$ for the decision of the category of the new data.

5. PROPOSED METHOD

5.1. Viewpoints of proposed method

In the proposed method, we focused on coding and decoding methods and propose new approach for each method. The conventional coding methods made a numerical table whose row represents the categories, where each category has one code word and the data belonging to the same categories have the same code word. Therefore, the conventional code table cannot construct configuration of the binary classifies which classify the training data belonging to same categories.

Therefore, we propose the code table whose row represents a representative training data in order to grasp the property of sub-categories, which are not observed. Our proposal allows us to classify the data belonging to the same category in the different category. This setting enables that the code words of training data belonging to the same sub-category is represented by same vectors, then the differences among the sub-categories can be represented.

Here, the j-th sub-categories of c_k is denoted by $c_{k,j}$, Table 3 shows an example of the code table whose row represents the training data which consider the sub-categories. In this way, we can consider the classifiers which classify the sub-categories belonging to the same category by using the code table whose row represents training data.

Next, for the decoding method, we consider the method considering the code table that each category is designed by multiple code words. The conventional decoding method assumes that each category has only one code word. However, as shown in Table 3, each category may have several code words in the code table using sub-categories. Therefore, we propose the new method which is robust to noise for the code table that each category has some plural code words.

Table 3. code table using sub-categories

			classifier			
c_k	$c_{k,j}$	training data	1	2	3	4
c_1	$c_{1,1}$	1	1	0	1	1
	$c_{1,1}$	2	1	0	1	1
	$c_{1,2}$	3	1	1	1	1
	$c_{1,2}$	4	1	1	1	1
c_2	$c_{2,1}$	1	1	1	0	0
	$c_{2,1}$	2	1	1	0	0
	$c_{2,2}$	3	0	1	0	0
	$c_{2,2}$	4	0	1	0	0
c_3	$c_{3,1}$	1	0	0	1	0
	$c_{3,1}$	2	0	0	1	0
	$c_{3,2}$	3	0	1	1	0
	$c_{3,2}$	4	0	1	1	0

5.2. Coding method based sub-category setting

In this section, we describe the proposed coding method in detail. The overall flow of the proposed coding method is as follows, 1) randomized generation of sub-categories, 2) design of classifiers' configuration (allocation of $0,1,*$). First, the generation of sub-categories is described. In the generation of sub-categories, we introduce the concept of centroids. The centroids of sub-categories are selected randomly from training data in each category, and the sub-categories are generated in terms of the minimal distances between the centroids and training data. This approach makes it possible to divide the categories into plural sub-categories which include similar training data.

In the generation of classifiers' configuration, we select the sub-category groups' centroid from the centroid of each sub-category. The sub-categories are grouped considering distances from each centroid of sub-categories. The obtained sub-category groups can be regarded as categories, and new code table is generated. This handling enables to group the similar property sub-categories among different categories, and to generate a code table which considers the property of

each sub-category. If a new code table does not have enough classifiers, the accuracy rate becomes worse; therefore, the 2 steps are repeated and combining these code words until the combined code table contains the desired number of classifiers. On the other hand, in the code table generation using sub-category groups, the code table with K rows is prepared in advance and the code table is generated based on the prepared code table.

In the following discussion, let N_k be the number of training data in c_k, $N = \sum_{k=1}^{K} N_k$ be the number of all training data, S be the number of sub-categories of each category, the code table $\boldsymbol{H}$ be prepared code table having K rows and $\lfloor x \rfloor$ be the largest integer less than x.

Step1) Selection of sub-categories' centroids

In each category c_k, S training data $\boldsymbol{x}_n = (x_{n,1}, \ldots, x_{n,d})$ are selected randomly, and the training data are defined as the centroids of $c_{k,j}$.Here, let $\boldsymbol{y}_{k,j} = (y_{k,j}^1, \ldots, y_{k,j}^d)$ be the centroid of $c_{k,j}$.

Step2) Selection of sub-categories groups' centroids

In each category c_k, the distances between the centroids of sub-category $\boldsymbol{y}_{k,j}$ and the training data are calculated. The set of training data with small distances are divided into $c_{k,j}$ on condition that the number of training data belonging $c_{k,j}$ equal to $\lfloor N_k/S \rfloor$.

Step3) Selection of sub-category groups' centroids

In each sub-category $c_{k,j}$, one centroid of sub-category is randomly selected from the set of sub-categories' centroids, and the centroid is defined as the centroid of the sub-category group.

Step4) Grouping the sub-categories

The distances between centroids of the sub-category group and centroids of sub-categories are calculated. The sub-categories with small distances are divided into the sub-category groups on the condition that the number of sub-categories belonging the groups equal to S.

Step5) Generation of the code table based on sub-categories groups

A code table is generated based on $\boldsymbol{H}$, whose number of rows is K (the number of categories). In this generation, one sub-category is selected in each sub-category group, and the sub-category is not used for binary classification. The elements of the sub-category's code word are allocated $*$. The above processing is conducted until all sub-categories are selected once.

Step6) Combining of code table

Step1) to Step5) are repeated M times. The M code tables are combined to one code table.

5.3. Decoding method based on majority rule

In this section, we describe the proposed decoding method. In the conventional coding method, it is assumed that each category has only one code word. On the other hand, in the proposed coding method, each category has multiple code words, i.e. in each category, so that there can be multiple templates which are compared with the outputs of classifiers. Therefore, considering the multiple code words are expected to improve the accuracy rate for classification than considering only one code word.

In summary, the proposed decoding method calculates the similarity between all code words and the outputs of new data by the same way of conventional methods, and predicts the estimated category of new data by a majority rule of the categories to which the top l of high similarity code words belong.

6. EXPERIMENTS

6.1. Experimental conditions

In order to verify the performance of the proposed method, we conducted a simulation experiment by using Japanese newspaper articles. We used the Mainichi Newspapers published in 2010 with 9 categories. All articles belong to only one category. 200 articles in each category are used as the training data and 100 articles are used as test data at random. The accuracy rate and the computational complexity for training classifiers are applied as the evaluation criteria. The computational complexity was calculated for both case of the normal processing and parallel processing. These evaluation experiments are repeated 5 times and the average of 5 times are used for evaluation.

The setting parameter S was set as 3 and the prepared code table $\boldsymbol{H}$ is one-vs-the rest. The comparison approaches are the Exhaustive code and the one-vs-the rest.

6.2 The result of experiment and discussion

The numbers of classifiers are shown in Table 4. Table 5 shows the result for $M = 10$ and $l = 3$. Figure 1 shows the relation among the accuracy rate, the number of repentance M and the number of the vote in the majority rule for deciding category l.

Table 4. the number of classifiers

code table	number of classifiers	
	binary code table	ternary code table
Exhaustive	255	-
one-vs-the rest	9	-
proposed method	-	27×M

Table 5. the result for $M = 10$ and $l = 3$

code table	accuracy rate	Training computational complexity (second)	
		normal	parallel
Exhaustive	0.763	58,822	330
One-vs the rest	0.723	1,259	230
Proposed method	0.776	15,711	113

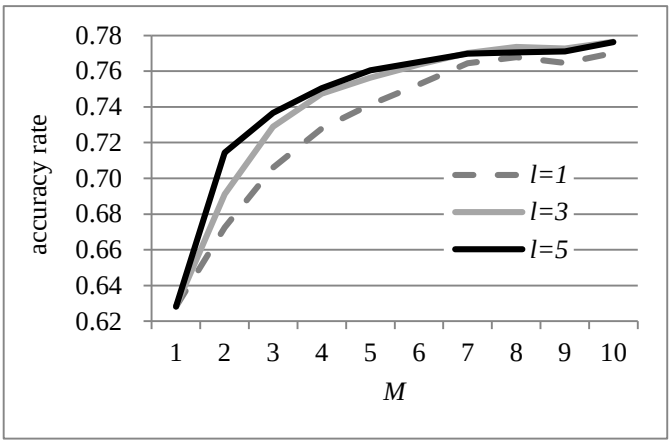

Figure1. The accuracy rate of each M and l

From Table 5, the proposed method has the highest accuracy rate. In the case of normal process of calculation, the one-vs-the rest has the smallest computational complexity. In the case of parallel processing, the proposed method has the smallest computational complexity. Focusing on the exhaustive code and the proposed method, both computational complexity and accuracy rate of the proposed method are better than that of the exhaustive code while the proposed method has the number of classifiers at the same level as exhaustive code. The reason why the proposed method has less computational complexity, especially in the case of parallel processing, is that the proposed method needs less training data in each classifier by using a ternary code table. The accuracy rate of the proposed method becomes high although this method is based on one-vs-the rest. From this result, the effectiveness of coding and decoding methods considering the sub-category setting is verified.

From Figure 1, the effectiveness of introducing the majority voting rule into the decoding method is verified. In particular, the effectiveness is enhanced when M is small. From this result, we see that the majority rule approach performs better when the number of classifiers is small. On the other hand, the accuracy rate is relatively row when M is small; there are two reasons. First one is that the number of classifiers is small. Second one is that there are not various code words among the data belonging to different categories in the setting of small M. This happens because the code table is generated by regarding the data belonging to the same sub-category group as the same category even if the data belong to different categories. On the other hand, the accuracy rates are equal in all of l when $M = 1$. It is because that the number of unique code words is the same to the number of sub-categories, and there are many same code words.

6. CONCLUSION AND FUTURE WORKS

In this study, we focused on the classification of digital document data based on the ECOC approach. We proposed a new coding method considering sub-categories and a decoding method based on majority rule. From the result of experiments, the effectiveness of our proposed method from the viewpoints of classification accuracy and computational complexity are shown. Future works are to construct the coding algorithm without randomness and iteration.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to Dr. Kenta Mikawa, and all members of Goto laboratory, Waseda University who support us for their helpful comments in this research. A part of this study was supported y JSPS KAKENHI Grant Numbers 26282090 and 26560167.

REFERENCES

Allwein, E. L., Schapire, R. E., & Singer, Y. (2001) Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research*, ***1***, 113-141.

Bouzas, D., Arvanitopoulos, N., & Tefas, A. (2010) Optimizing subclass discriminant error correcting output codes using particle swarm optimization. *The 2010 International Joint Conference on Neural Networks,* 1-7.

Chmielnicki, W. (2015) Creating Effective Error Correcting Output Codes for Multiclass Classification. In *Hybrid Artificial Intelligent Systems,* 502-514.

Cover, T.M. & Thomas, J.A. (2012) *Elements of information theory*. John Wiley & Sons.

Crammer, K., & Singer, Y. (2002) On the learnability and design of output codes for multiclass problems. *Machine learning*, ***47***, 201-233.

Dietterich, T.G. & Bakiri, G. (1995) Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research*, ***2***, 263-286.

Escalera, S., Pujol, O., & Radeva, P. (2006) ECOC-ONE: A novel coding and decoding strategy. *Proc. 18th International Conference on Pattern Recognition*, ***3***, 578-581.

Escalera, S., Pujol, O., & Radeva, P. (2007) Boosted Landmarks of Contextual Descriptors and Forest-ECOC: A novel framework to detect and classify objects in cluttered scenes. *Pattern Recognition Letters*, ***28***, 1759-1768.

Escalera, S., Tax, D. M., Pujol, O., Radeva, P., & Duin, R. P. (2008) Subclass problem-dependent design for error-correcting output codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ***30***, 1041-1054.

Escalera, S., Pujol, O., & Radeva, P. (2009) Recoding error-correcting output codes. In *Multiple Classifier Systems*, 11-21. Springer Berlin Heidelberg.

Escalera, S., Pujol, O. & Radeva, P. (2010) On the decoding process in ternary error-correcting output codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* ***32***, 120-134.

Escalera, S., Masip, D., Puertas, E., Radeva, P., & Pujol, O. (2011) Online error correcting output codes. *Pattern Recognition Letters*, ***32***, 458-467.

Huang, T.K., Weng, R. C. & Lin, C. J. (2006) Generalized Bradley-Terry models and multi-class probability estimates. *Journal of Machine Learning Research*, ***7***, 85-115.

Ikeda, S. (2010) Combining Binary Machines for Multi-class: Statistical Model & Parameter Estimation. *Proc. The Institute of Statistical Mathematics*, ***58***, 157-166. (in Japanese)

Ogihara, T., Mikawa, K. & Goto, M. (2013) Multi-valued classification of text data based on ECOC approach considering parallel processing. *Proc. The 14th Asia Pacific Industrial Engineering and Management Systems Conference*.

Oyama, Y., Takenouchi, T. & Ishii, S. (2008) A hierarchical multi−class classification method based on error−correcting output coding. *The Journal of the Institute of Electronics, Information and Communication Engineers*, ***197***, 337-342. (in Japanese)

Pujol, O., Radeva, P. & Vitria, J. (2006) Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* ***28***, 1007-1012.

Pujol, O., Escalera, S., & Radeva, P. (2008) An incremental node embedding technique for error correcting output codes. *Pattern Recognition*, ***41***, 713-725.

Rifkin, R., & Klautau, A. (2004) In defense of one-vs-all classification. *The Journal of Machine Learning Research*, ***5***, 101-141.

Smith, R.S., & Windeatt, T. (2005). Decoding rules for error correcting output code ensembles. *International Workshop on Multiple Classifier Systems* 53-63.

Tipping, M.E. (2001) Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, 211-244.

Xue, A., Wang, X., Song, Y., & Lei, L. (2015) Discriminant error correcting output codes based on spectral clustering. *Pattern Analysis and Applications*, 1-19.

Zhang, X.L. (2015) Heuristic Ternary Error-Correcting Output Codes Via Weight Optimization and Layered Clustering-Based. Approach. *IEEE Transactions on Cybernetics* **45**, 289-301.

Zhong, G., & Cheriet, M. (2013) Adaptive error-correcting output codes. *Proc. The Twenty-Third international joint conference on Artificial Intelligence* 1932-1938. AAAI Press.