IEEE International Conference on Systems, Man, and Cybernetics,
October 9-12, 2022, Prague, Czech Republic

Construction Methods for Error Correcting Output Codes Using Constructive Coding and Their System Evaluations

Shigeichi Hirasawa (Waseda University)
Gendo Kumoi (Waseda University)
Hideki Yagi (The University of Electro-Communications)
Manabu Kobayashi (Waseda University)
Masayuki Goto (Waseda University)
Hiroshige Inazumi (Aoyama Gakuin University)

There are two methods for solving the multi-valued classification problem:

(i) a method of directly extending a single binary classifier to a multi-valued classifier and

(ii) a method of constructing a multi-valued classifier using multiple binary classifiers.

→ Error Correcting Output Code (ECOC)

Out line of this paper

Continued

I. Introduction

[Code construction]

a Reed-Muller (RM) code

↓・・・・ modifying the RM code to be suitable for the ECOC

a modified RM (mRM) code [7]

↓↑・・・relationship

Hadamard matrix [8]–[10]

⇩

a class of the simplex codes・・・ satisfy the Plotkin bound by the equality

[System Evaluations]

the trade-off relationships between :

- code length (system investment cost) N and
- the probability of classification error P_{ce} between categories (system performance degradation)

- for the number of categories (system scale) M,

using (a) artificial data and (b) benchmark data

It is shown that as the number of categories M becomes large, the ECOC system has “elastic property” and “effective elastic property”.

Appendix A

TABLE VI
CORRESPONDENCE TABLE BETWEEN RATE-DISTORTION, GENERAL SYSTEM EVALUATION MODEL AND ECOC.

Rate-Distortion Theory [18]	System Evaluation Model [18], [20]	ECOC
Rate (R)	Investment Cost (r)	Number of Binary Classifiers (n)
Distortion (D)	Performance Degradation (d)	Probability of Classification Error between Categories (p_{ce})
	Scale of System (G)	Number of Categories (M)

Trade-off relationships

II. Codeword Tables

A. Configuration of Codeword Tables and Their Properties

The performance of the ECOC is determined by the codeword table W with M rows and N columns, where

$$W = [w_{ij}], \quad w_{ij} \in \{0, 1\} \quad (i = 1, 2, \ldots, M; j = 1, 2, \ldots, N). \tag{1}$$

The i-th row of W, $\boldsymbol{c}_i$, and the j-th column of W, $\boldsymbol{d}_j$ are represented by

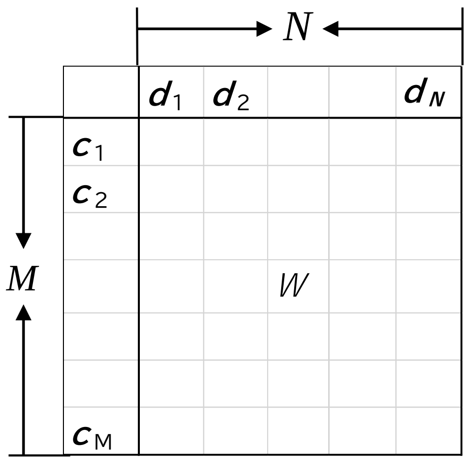
$$\boldsymbol{c}_i = (w_{i1}, w_{i2}, \ldots, w_{iN}), \tag{2}$$

and

$$\boldsymbol{d}_j = (w_{1j}, w_{2j}, \ldots, w_{Mj})^{\mathrm{T}},$$

where T indicates the transpose of the vector. Here, the i-th

Codeword Table



	ECOC	constructive code
N	# of binary classifiers	code length
M	# of categories	# of codewords
		$(N, \log_2 M, D)$ code

Definition 1. *Letting a binary vector of length L be $\boldsymbol{u} = (u_1, u_2, \ldots, u_L)$, we call the binary vector $\boldsymbol{u}^{\mathrm{C}} = (u_1^{\mathrm{C}}, u_2^{\mathrm{C}}, \ldots, u_L^{\mathrm{C}})$, the complement vector of $\boldsymbol{u}$, where $u_\ell \oplus u_\ell^{\mathrm{C}} = 1$ $(\ell = 1, 2, \ldots, L)$ holds, and the symbol $\oplus$ denotes the exclusive OR operation.*

Examples

d_j	d_j^{C}
0	1
0	1
1	0
0	1
1	0
1	0

(M=6)

c_i	1	0	1	1	1	0	1
c_i^{C}	0	1	0	0	0	1	0

(N=7)

0 ⇄ 1 : interchanged

Note that obviously for the column vectors, valid codeword tables *do not* contain:

(i) identical column vectors,
(ii) the all 0's and the all 1's column vectors, and
(iii) the column vector $\boldsymbol{d}_j^{\mathrm{C}}$, if $\boldsymbol{d}_j$ exists for any j. [2]
These are called the *column operation for the ECOC*.

Similarly, for the row vectors, they *do not* contain:

(i) identical row vectors, and
(ii) the row vector $\boldsymbol{c}_i^{\mathrm{C}}$, if $\boldsymbol{c}_i$ exists for any i. [3]
These are called the *row operation for the ECOC*.

[2] This is because they have the same classification boundary and the outputs of them are highly correlated.

[3] This is because the category c_i and the category c_i^{C} are always classified into separate groups, even if the all binary classifiers which divide into two groups are used.

B. Exhaustive Codes [6]

For a given M, generate the all 2^M column vectors of length M. Then the column operation described above is performed on these column vectors. The resultant codeword table gives that for the $(N_{\max}, \log_2 M, (N_{\max}+1)/2)$ exhaustive code with M rows and $N_{\max}$ columns, where

$$N_{\max} = 2^{M-1} - 1. \qquad (4)$$

Example : Case of M=5, N=15

32

15

5

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

d_j … d_j^C

C. *Modified Reed-Mullar Codes [7]*

For any positive integer $m\,(\geq 2)$, there is a first order $(2^m, m+1, 2^{m-1})$ linear Reed-Muller (RM) code. Here, generate an RM code such that $2M = 2^{m+1}$, then the row operation for the ECOC is performed. The resulting codeword table gives that of an $(M-1, \log_2 M, M/2)$ modified RM (mRM) code with M rows and $N\,(= M-1)$ columns.

Example 2. *For* $M = 8$*, codeword table of* $(8,4,4)$ *RM code, and that of* $(7,3,4)$ *mRM code are shown in Tables II and III, respectively.* □

c_i

c_i^C

TABLE II
CODEWORD TABLE OF THE $(8,4,4)$ RM CODE.

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
c_1	1	1	1	1	1	1	1	1
c_2	1	0	1	0	1	0	1	0
c_3	1	1	0	0	1	1	0	0
c_4	1	0	0	1	1	0	0	1
c_5	1	1	1	1	0	0	0	0
c_6	1	0	1	0	0	1	0	1
c_7	1	1	0	0	0	0	1	1
c_8	1	0	0	1	0	1	1	0
c_9	0	0	0	0	0	0	0	0
c_{10}	0	1	0	1	0	1	0	1
c_{11}	0	0	1	1	0	0	1	1
c_{12}	0	1	1	0	0	1	1	0
c_{13}	0	0	0	0	1	1	1	1
c_{14}	0	1	0	1	1	0	1	0
c_{15}	0	0	1	1	1	1	0	0
c_{16}	0	1	1	0	1	0	0	1

TABLE III
CODEWORD TABLE OF THE $(7,3,4)$ MRM CODE.

	d_1	d_2	d_3	d_4	d_5	d_6	d_7
c_1	1	1	1	1	1	1	1
c_2	0	1	0	1	0	1	0
c_3	1	0	0	1	1	0	0
c_4	0	0	1	1	0	0	1
c_5	1	1	1	0	0	0	0
c_6	0	1	0	0	1	0	1
c_7	1	0	0	0	0	1	1
c_8	0	0	1	0	1	1	0

III. Codeword Table Based on Constructive Coding

A. Modified RM Code and Hadamard Matrix

The Plotkin bound for binary codes is given by the following formula [8]:

$$\text{(Plotkin Bound)} \qquad D \leq \frac{NM}{2(M-1)}, \tag{5}$$

where the RHS is the average value of the Hamming distance between any two different codewords. If the bound (5) is satisfied by equality, it is one of a few equidistant codes,

The $(M-1, \log_2 M, M/2)$ mRM code is superior compare to the $(M, \log_2 M, M/2)$ orthogonal code with relatively high rates, which achieve a linear equidistant code with the distance $M/2$, where $N = M - 1$ holds. However, an mRM code exists only the case where $N = 2^m - 1$.

On the other hand, for the Hadamard matrix H_M whose $M \times M$ elements are from $\{-1, +1\}$, the codeword table of the equidistant code is also obtained by replacing $+1$'s by 0's, and -1's by 1's and removing the all 0's column (or the all 1's column), the resultant codeword table gives an $(M-1, \log_2 M, M/2)$ code which has the same code parameter as that given by mRM code, where $N = 2^m - 1$ holds.

In addition, any positive integer ℓ ($\ell \geq 3$), the Hadamard matrix H_M is expected to exist, when $M = 4\ell$ [8], [9], [12]. Hence there is a hypothesis that it exists [8], [13], and examples are known with their construction methods [10].

Example 3. *Example of Hadamard matrix H_{12} [8] is shown in Table IV.*

TABLE IV
HADAMARD MATRIX H_{12} ($M = 12$) [8]

+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
+1	-1	-1	+1	-1	-1	-1	+1	+1	+1	-1	+1
+1	-1	+1	-1	-1	-1	+1	+1	+1	-1	+1	-1
+1	+1	-1	-1	-1	+1	+1	+1	-1	+1	-1	-1
+1	-1	-1	-1	+1	+1	+1	-1	+1	-1	-1	+1
+1	-1	-1	+1	+1	+1	-1	+1	-1	-1	+1	-1
+1	-1	+1	+1	+1	-1	+1	-1	-1	+1	-1	-1
+1	+1	+1	+1	-1	+1	-1	-1	+1	-1	-1	-1
+1	+1	+1	-1	+1	-1	-1	+1	-1	-1	-1	+1
+1	+1	-1	+1	-1	-1	+1	-1	-1	-1	+1	+1
+1	-1	+1	-1	-1	+1	-1	-1	-1	+1	+1	+1
+1	+1	-1	-1	+1	-1	-1	-1	+1	+1	+1	-1

Replace +1's by 0's, and −1's by 1's, then we obtain the (11, log212, 6) simplex code.

B. Simplex Code

An $(N, \log_2(N+1), (N+1)/2)$ binary simplex code

↓ ・・・known to be generated by

a $(2^m - 1, m, 2^{m-1})$ dual code of the $(2^m - 1, 2^m - 1 - m, 3)$ Hamming code [9] .

・ In that sense, the modified RM code gives another method for generating the simplex code. This is, however, only the case when $N = 2^m - 1$.

The Hadamard matrix H_M ($M = 4\ell$ for ℓ ($\ell \geq 3$)))is expected to exist

↓・・・ replace +1's by 0's, and −1's by 1's, and

↓・・・ by removing the all 0's (or the all 1's) columns of a given H_M

$(N, \log_2(N+1), (N+1)/2)$ binary simplex code =

$(M-1, \log_2 M, M/2)$ binary simplex code

IV. Performance of ECOC Uing Constructive Coding

A. Performance Evaluation Methods for ECOC

Let us give data discussed here as follows:

- Artificial Data: M-dimensional multi-valued classification data are generated from an M-dimensional Gaussian distribution with mean $\boldsymbol{\mu}$ and variance Σ^2, denoted by $\mathcal{N}(\boldsymbol{\mu}, \Sigma^2)$ [15], where we set $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_M)$, $\mu_i = \mu$ for $i = 1, 2, \ldots, M$, and $\Sigma^2 = \sigma^2 I_M$, where I_M is the $M \times M$ identity matrix.
- Benchmark Data: The handwritten numbers and English characters, EMNIST data [16].

The performance evaluation of the ECOC is taken by the average probability of the worst classification error between categories P_{ce}, which is often discussed in the classification problems for given N and M, and will be later defined by (10).

$$P_{ce} = \sum_{i=1}^{M} P(c_i) \sum_{i' \neq i} \Pr\{g(c_{i'}|\boldsymbol{x}) \geq g(c_i|\boldsymbol{x})\}/(M-1). \quad (10)$$

B. Analysis of Classification Performance for Simplex Codes

Input the data $\boldsymbol{x}$ into the j-th binary classifier d_j of the codeword table $W = [w_{ij}]$ given by constructive coding. Assuming that the output $f_j(w_{ij}|\boldsymbol{x})$ of d_j is an ideal binary classifier which outputs the true posterior probability, the estimated category $c_{\hat{i}}$ of the input $\boldsymbol{x}$ is given by the following equations [6], [17]:

$$\hat{i} = \arg\min_i g(c_i|\boldsymbol{x}), \tag{6}$$

where for $w \in \{0, 1\}$,

$$f_j(w|\boldsymbol{x}) = \sum_{i'=1|w_{i'j}=w}^{M} \Pr\{c_{i'}|\boldsymbol{x}\}, \tag{7}$$

and

$$g(c_i|\boldsymbol{x}) = \sum_{j=1}^{N} (1 - f_j(w_{ij}|\boldsymbol{x})). \tag{8}$$

Example 4. *In the case of codeword table of an* (N, K, D) *equidistant code, the function* $g(c_i|\boldsymbol{x})$ *is given by the following equation under the proper assumptions:*

$$g(c_i|\boldsymbol{x}) = D[1 - \Pr\{c_i|\boldsymbol{x}\}]. \tag{9}$$

Note that for equidistant codes, the function $g(\cdot|\cdot)$ depends only on category c_i and not on the other categories $c_{i'}$ $(i' \neq i)$, and is proportional to D.

C. Comparison between RM Codes and mRM Codes

The mRM code improves the Pce removing the complement codewords of the RM code

↓・・・ conversely

The number of complements is increased by replacing the s codeword(s) of the mRM code with its (their) complement(s) for $M = 8$

of the complements : $s = 0$ (mRM code),

1 (replace $\boldsymbol{c}_8$ by $\boldsymbol{c}_9$ since $\boldsymbol{c}_1 = \boldsymbol{c}_9^{C}$),

2 (in addition replace $\boldsymbol{c}_7$ by $\boldsymbol{c}_{10}$, since $\boldsymbol{c}_2 = \boldsymbol{c}_{10}^{C}$

3, and 4

(a) Artificial data : $(\mu, \sigma^2) = (1.0, 0.1)$ ← Numerical computation

(b) Benchmark data : EMNIST ← Experiment

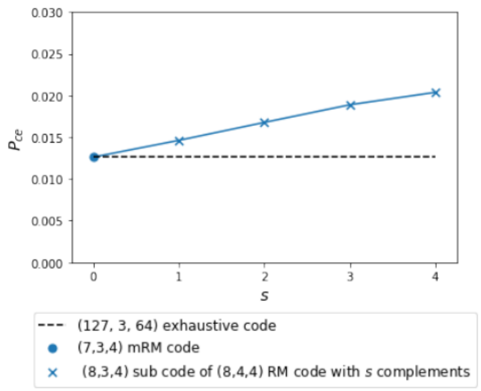


Fig. 1. Performance (P_{ce}) comparison between RM codes and mRM codes for artificial data $(\mu, \sigma^2) = (1.0, 0.1)$.

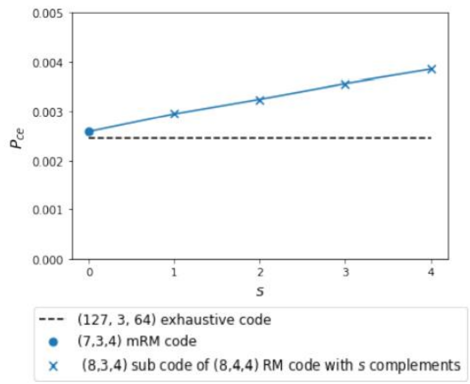


Fig. 2. Performance comparison (P_{ce}) between RM codes and mRM codes for benchmark data (EMNIST).

The P_{ce} is increased as s increased.

D. The Probability of Classification Error between Categories P_{ce} for ECOC using Simplex Codes

(a) Artificial data : $(\mu, \sigma^2) = (1.0, 0.1)$ ←Numerical computation
(b) Benchmark data : EMNIST ← Experiment

TABLE V
PERFORMANCE (P_{ce}) OF THE SIMPLEX CODES AND THE EXHAUSTIVE CODES.

M	Artificial Data		Benchmark Data		
	Simplex	Exhaustive	Simplex	Exhaustive	DNN
4	0.0128	0.0128	0.00312	0.00271	0.00625
8	0.0124	0.0124	0.00259	0.00246	0.01000
12	0.0124	0.0124	0.00481	0.00506	0.01667

VI. DISCUSSIONS

B. Comparison Between Simplex Codes and Exhaustive codes

The performance P_{ce} of the exhaustive codes is theoretically considered to be the best value which can be achieved by the ECOC. From Table V, for the artificial data, the theoretical value of the simplex code matches that of the exhaustive code. In addition, it can be seen that the benchmark data shows the values for the simplex codes also very close to those of the exhaustive codes except in the case of the DNN.

V. System Evaluation Using a Trade-off Function

A. Trade-off Relationship

The ECOC has a trade-off relationship between

- The code length N (investment cost) and
- The probability of classification error P_{ce} (performance deterioration)]

for a given number of categories M (system scale)

The performance of the simplex code is evaluated by P_{ce}, referring that of the shortened exhaustive code of length N, where $N_{\min} \le N \le N_{\max}$, and $N_{\min} = \lceil \log_2 M \rceil$

B. System Evaluation of ECOC using Simplex Code

To relatively compare with any M, we normalize the variables as

- $n = N/N_{\max}$ and
- $p_{ce} = P_{ce}/P_{ce,\max}$,

where

$N_{\max} = 2^{M-1} - 1$

$N_{\min} = \lceil \log_2 M \rceil$, and

$P_{ce,\max} = 1/2$

Appendix

A. System Evaluation Model Based on Trade-off Relationship

The rate-distortion function can be written as:

$$R = R(D). \quad (11)$$

Let the rate R be normalized by the maximum of R, R_{max}, and the distortion D, by the maximum of D, D_{max}, then we have the following normalized function by $r = R/R_{max}$, and $d = D/D_{max}$, and introducing G:

$$r = r\,(d;\, G). \quad (12)$$

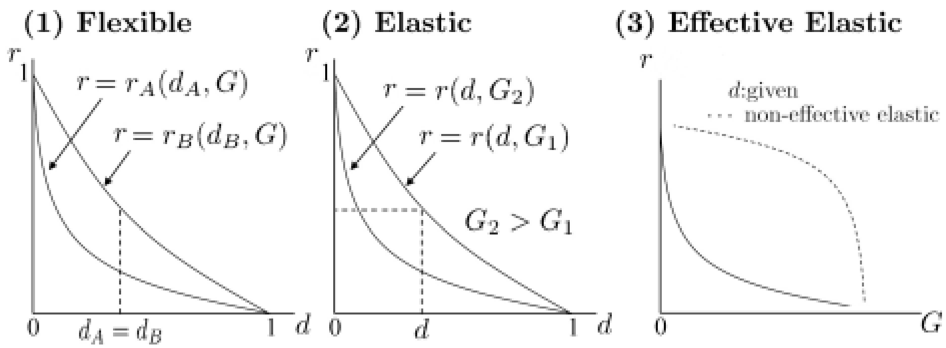


Fig. 6. Trade-off model for system evaluation

C. Computational Results by Artificial Data [17], [19]

D. Experimental Results by Benchmark Data

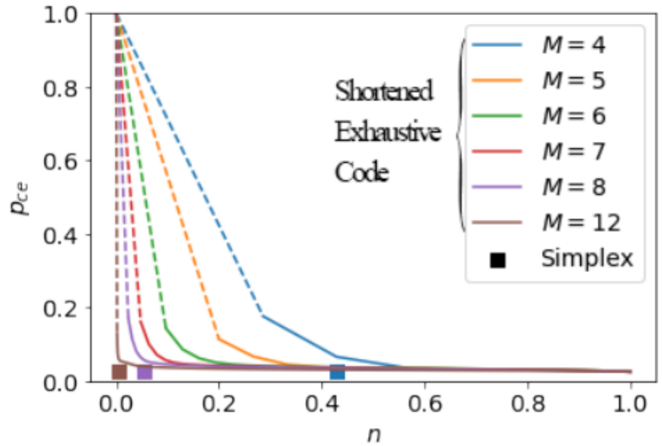


Fig. 3. Trade-off relationship between n and p_{ce} by the simplex codes and the shortened exhaustive codes for artificial data $(\mu, \sigma^2) = (1.0, 0.1)$.

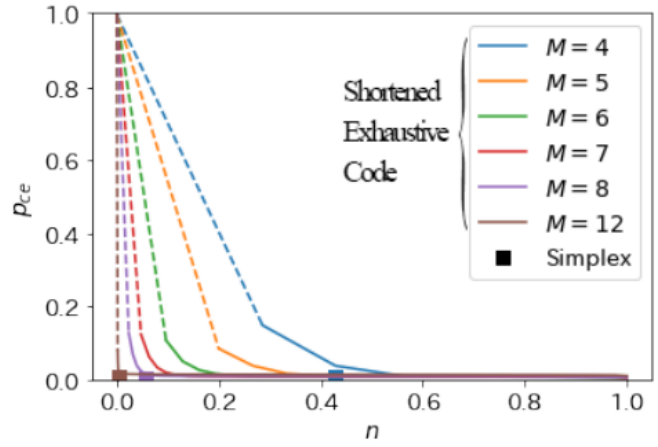


Fig. 4. The trade-off relationship between n and p_{ce} of the simplex codes and the shortened exhaustive codes for benchmark data (EMNIST).

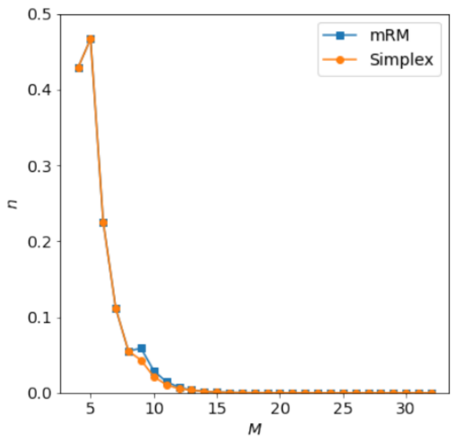


Fig. 5. The relationship between n and M of the mRM codes and the simplex codes for both the artificial data $(\mu, \sigma^2) = (1.0, 0.1)$ and the benchmark data (EMNIST) .

C. Properties of Simplex Codes by System Evaluation Model

From Figs. 3 and 4, for both artificial and benchmark data, the following results are clarified:

(i) Flexible property

For a given M, the p_{ce} of the simplex code (indicated by ■) is almost on the trade-off curve of the shortened exhaustive code. The shortened exhaustive code exhibits Flexible property.

(ii) Elastic property

As M becomes larger, the p_{ce} of the simplex code goes toward the origin.

(iii) Effective Elastic property

From Fig. 5, as M increases, the interval between the positions (indicated by ■) of p_{ce} decreases.

D. Further Remarks

In the above discussion so far, there exists a simplex code for $M = 2^m$ $(m \geq 2)$ or $M = 4\ell$ $(\ell \geq 3)$, where the obtained p_{ce} is shown by ■ in the figures.

(i) If $M = 2^m$, then there are $2^{m+1} - 2^m - 1 = 2^m - 1$ cases, where the simplex code cannot be available.

(ii) If $M = 4\ell$, only the $4(\ell + 1) - 4\ell - 1 = 3$ is sufficient.

eg. $M = 7$: $(7, \log_2 7, 4)$ code, $M = 6$: $(7, \log_2 6, 4)$ code, and $M = 5$: $(7, \log_2 5, 4)$ *.

(*) Strictly speaking, note that, however, there is a reverse in performance between the $(7, \log_2 4 = 2, 4)$ subcode of the $(7, 3, 4)$ simplex code and the $(3, \log_2 4 = 2, 2)$ simplex code.

VII. Concluding Remarks

(i) It has been shown that the modified RM code proposed by M. Goto and M. Kobayashi [7] for improving the performance of ECOC is a linear equidistant code, and gives one of the methods for generating the simplex code with $N = 2^m - 1$.

(ii) It is known that the simplex codes can be generated from the Hadamard matrices, which is conjectured to exist for multiples of 4. So it is applicable enough for required M from the example of the number of categories (codewords) $M \leq 1000$ (except 668, 716, and 892) [10].

(iv) It is sufficiently practical to solve the classification problem. Furthermore, in this paper, it is clarified by the system evaluation model that the excellent properties of ECOC using the simplex code have elastic and effective elastic. These properties imply that the relative performance degradation does not occur even if the number of categories M increases.

(v) The codeword construction methods which combines the simplex code and other good codes are remained as future works.