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

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

Shigeichi Hirasawa
Center for Data Science
Waseda University
Tokyo, Japan
hira@waseda.jp

Gendo Kumoi
Center for Data Science
Waseda University
Tokyo, Japan
kumoi@aoni.waseda.jp

Hideki Yagi
Dept. of Computer & Network Eng.
The University of Electro-Communications
Tokyo, Japan
h.yagi@uec.ac.jp

Manabu Kobayashi
Center for Data Science
Waseda University
Tokyo, Japan
mkoba@waseda.jp

Masayuki Goto
Dept. of Ind. & Mgmt. Systems Eng.
Waseda University
Tokyo, Japan
masagoto@waseda.jp

Hiroshige Inazumi
Dept. of Social Informatics
Aoyama Gakuin University
Tokyo, Japan
inazumi@si.aoyama.ac.jp

***Abstract*—Consider M-valued ($M \geq 3$) classification systems realized by combination of N ($N \geq \lceil \log_2 M \rceil$) binary classifiers. Such a construction method is called an Error Correcting Output Code (ECOC). First, focusing on a Reed-Muller (RM) code, we derive a modified RM (mRM) code to make it suitable for the ECOC. Using the mRM code and the Hadamard matrix, we introduce a simplex code which is one of the powerful equidistant codes. Next, from the viewpoint of system evaluation model, we evaluate the ECOC by using constructive coding described above. We show that they have desirable properties such as *Flexible*, *Elastic*, and *Effective Elastic* as M becomes large, by employing analytical formulas and experiments.**

***Index Terms*—multi-valued classification, ECOC, error correcting code, Reed-Muller code, simplex code, Hadamard matrix, trade-off, system evaluation**

I. Introduction

For binary classifiers, there have been studies on such as Support Vector Machines (SVM) [1]–[3], Relevance Vector Machines (RVM) [4] and Regularized Least-Squares Classification [5]. 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. Although the former uses a high-performance classifier such as the SVM, it requires the large amount of space and time calculation as M becomes large, where M is the number of categories for the multiple classification problems. If the accuracy is remained in higher; as a result, the cost increases. The latter is realized by combining N binary classifiers in parallel, where N is the number of binary classifiers. It is a powerful method for large M, since it can be constructed systematically by using the concept of error correcting codes. Hence, it is called Error Correcting Output Code (ECOC) [6].

In this paper, first we focus on a Reed-Muller (RM) code which is efficient in low-rate, where the RM code is also one of the practically noteworthy codes which is used for imagetransmission of the Marina 6 in the 1960s. We derive a modified RM (mRM) code [7] by modifying the RM code to be suitable for the ECOC method, and further clarify the relationship between it and a Hadamard matrix [8]–[10]. The obtained codes are equidistant codes which are a class of the simplex codes, and the performance will be shown when these codes are used for the ECOC. The simplex code is an excellent code in the sense that it satisfies the Plotkin bound by the equality.

Next, for the number of categories (system scale)[1] M, using artificial data and benchmark data, we investigate the trade-off relationships between code length (system investment cost) N and the probability of classification error P_{ce} between categories (system performance degradation) from the standpoint of system evaluation using the normalized trade-off functions. It is shown that as the number of categories M becomes large, the ECOC system has *elastic* property and *effective elastic* property.

Throughout this paper, we shall evaluate the average probability of the worst classification error between categories for the ECOC methods with binary classifiers. In Section II, we define a codeword table, which decides the performance of the ECOC method. Construction methods for valid codeword tables for the ECOC which can improve the performance

[1]Terminology A of (A) stands for which used in general system evaluation model.

are clarified, and a modified RM code is derived from an RM code. In Section III, introducing a Hadamard matrix, we lead the simplex code. A mathematical derivation on the performance of constructive coding is also shown in Section IV. Section V discusses the system evaluations of the ECOC using constructive coding. Discussions on the properties of the ECOC methods are given in Section VI. Section VII describes concluding remarks.

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

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

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

$$\boldsymbol{d}_j = (w_{1j}, w_{2j}, \ldots, w_{Mj})^{\mathrm{T}}, \tag{3}$$

where T indicates the transpose of the vector. Here, the i-th row vector $\boldsymbol{c}_i$ and the j-th column vector $\boldsymbol{d}_j$ of the codeword table indicate the representative of the i-th category c_i, and the boundary region of the j-th binary classifier d_j, respectively.

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.*

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*.

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

$$N_{\max} = 2^{M-1} - 1. \tag{4}$$

Here, the code of length N, the number of information symbols K, and the minimum design distance D is denoted as the (N, K, D) code. Hereafter, we use shortened versions of the exhaustive code denoted by a shortened exhaustive code. The codes composed by selecting $N\,(< N_{\max})$ column vectors from the exhaustive codes of length $N_{\max}$ are called shortened exhaustive codes, where the number of such codes is $\binom{N_{\max}}{N}$.

Example 1. *For* $M = 5$*, the codeword table of an exhaustive code is shown in Table I.* □

TABLE I
CODEWORD TABLE OF EXHAUSTIVE CODE $(M = 5, N_{\max} = 15, D = 8)$

	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$	$\boldsymbol{d}_3$	$\boldsymbol{d}_4$	$\boldsymbol{d}_5$	$\boldsymbol{d}_6$	$\boldsymbol{d}_7$	$\boldsymbol{d}_8$	$\boldsymbol{d}_9$	$\boldsymbol{d}_{10}$	$\boldsymbol{d}_{11}$	$\boldsymbol{d}_{12}$	$\boldsymbol{d}_{13}$	$\boldsymbol{d}_{14}$	$\boldsymbol{d}_{15}$
$\boldsymbol{c}_1$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\boldsymbol{c}_2$	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
$\boldsymbol{c}_3$	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
$\boldsymbol{c}_4$	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
$\boldsymbol{c}_5$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

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.* □

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

	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$	$\boldsymbol{d}_3$	$\boldsymbol{d}_4$	$\boldsymbol{d}_5$	$\boldsymbol{d}_6$	$\boldsymbol{d}_7$	$\boldsymbol{d}_8$
$\boldsymbol{c}_1$	1	1	1	1	1	1	1	1
$\boldsymbol{c}_2$	1	0	1	0	1	0	1	0
$\boldsymbol{c}_3$	1	1	0	0	1	1	0	0
$\boldsymbol{c}_4$	1	0	0	1	1	0	0	1
$\boldsymbol{c}_5$	1	1	1	1	0	0	0	0
$\boldsymbol{c}_6$	1	0	1	0	0	1	0	1
$\boldsymbol{c}_7$	1	1	0	0	0	0	1	1
$\boldsymbol{c}_8$	1	0	0	1	0	1	1	0
$\boldsymbol{c}_9$	0	0	0	0	0	0	0	0
$\boldsymbol{c}_{10}$	0	1	0	1	0	1	0	1
$\boldsymbol{c}_{11}$	0	0	1	1	0	0	1	1
$\boldsymbol{c}_{12}$	0	1	1	0	0	1	1	0
$\boldsymbol{c}_{13}$	0	0	0	0	1	1	1	1
$\boldsymbol{c}_{14}$	0	1	0	1	1	0	1	0
$\boldsymbol{c}_{15}$	0	0	1	1	1	1	0	0
$\boldsymbol{c}_{16}$	0	1	1	0	1	0	0	1

[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.

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

	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$	$\boldsymbol{d}_3$	$\boldsymbol{d}_4$	$\boldsymbol{d}_5$	$\boldsymbol{d}_6$	$\boldsymbol{d}_7$
$\boldsymbol{c}_1$	1	1	1	1	1	1	1
$\boldsymbol{c}_2$	0	1	0	1	0	1	0
$\boldsymbol{c}_3$	1	0	0	1	1	0	0
$\boldsymbol{c}_4$	0	0	1	1	0	0	1
$\boldsymbol{c}_5$	1	1	1	0	0	0	0
$\boldsymbol{c}_6$	0	1	0	0	1	0	1
$\boldsymbol{c}_7$	1	0	0	0	0	1	1
$\boldsymbol{c}_8$	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, and it is an excellent code, since it has the largest distance among them. As shown in *C.* of Section II, the 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 [4].

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.*

B. Simplex Code

An $(N, \log_2(N + 1), (N + 1)/2)$ binary simplex code is 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 $N = 2^m - 1$.

If we add an overall parity check to the codeword of the simplex code described above, and replace 0's by $+1$'s, and

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

1's by -1's, the resulting $M \times M$ codeword table forms a Hadamard matrix H_M. This means that we can easily obtain the $(M - 1, \log_2 M, M/2)$ simplex code from a given H_M, since the all 0's column (or the all 1's column) exists, where $M = 2^m = N + 1$ holds.

For $M = 4\ell$ ($\ell \geq 3$) except $M = 2^m$, a codeword table of the simplex code can be constructed in a similar manner, i.e., by removing the all 0's (or the all 1's) columns of a given Hadamard matrix. As an example, in Table IV, the heavy square line shows the codeword table of a simplex code for $M = 12$ with replacing $+1$'s by 0's, and -1's by 1's.

Hereafter, we discuss the performance of the ECOC constructed by simplex codes and their related codes, since our interest is concentrated to them [5].

IV. PERFORMANCE OF ECOC USING 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).

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

[4]The ECOC using Hadamard matrix only in the case $M = 2^m - 1$ has been discussed as Hadamard ECOC in [11].

[5]It is very interesting to note that the Bayes optimal ECOC has been shown to be obtained if and only if equidistant codes are used [14].

estimated category $c_{\hat{i}}$ of the input $\boldsymbol{x}$ is given by the following equations [6], [17]:

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

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

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

and

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

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:*

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

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.

On the other hand, the performance evaluation variable P_{ce} described in *A.* of this section is defined by the following equation:

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

C. Comparison between RM Codes and mRM Codes

We will show that the mRM code improves the classification performance of the ECOC by removing the complement codewords of the RM code.

When $(\mu, \sigma^2) = (1.0, 0.1)$ for the artificial data, the computational result is shown in Fig. 1. Here for M = 8, the number of the complement codewords is s, and the probability of the worst classification error between the categories P_{ce}, given by (10). When $s = 0$, the $(7, 3, 4)$ mRM code (Table III) is used, and when $s > 0$, the upper eight codewords of the codeword table of the $(8, 4, 4)$ RM code (Table II) are focused which is an (8, 3, 4) subcode of the original RM code. If $s = 1$, then replace c_8 by c_9 since $c_1 = c_9^{\mathrm{C}}$. If $s = 2$, then in addition replace c_7 by c_{10}, since $c_2 = c_{10}^{\mathrm{C}}$. Sequentially increase to replace the complement codewords for $s = 3$ and 4, and calculate P_{ce}.

We also examined benchmark data as experimental data, where the EMNIST data [16] are used. The performance comparison method between the RM code and the mRM code for the benchmark data EMNIST is quite similar to that for the artificial data given by Fig. 1, and the results are as shown in Fig. 2. However, the method of selecting a pair of a codeword and its complement is performed by all combinations of possible selection methods, and the average is taken to give P_{ce}. The reason is that the distribution of characteristic vectors for real data generally differs depending on the categories.

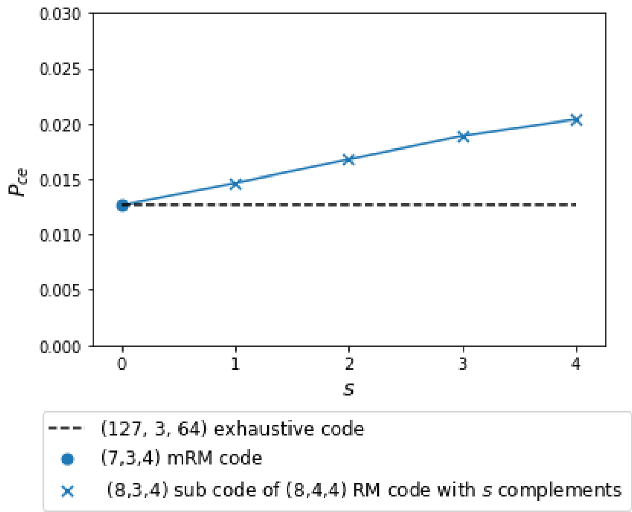

Fig. 1. Performance (P_{ce}) comparison between RM codes and mRM codes for artificial data $(\mu, \sigma^2) = (1.0, 0.1)$.

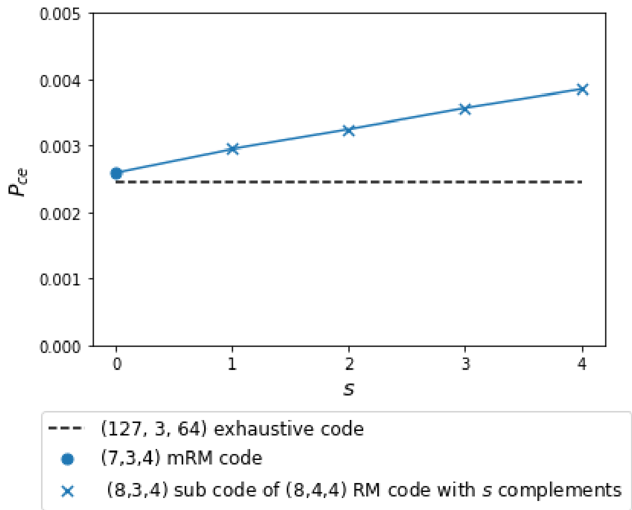

Fig. 2. Performance comparison (P_{ce}) between RM codes and mRM codes for benchmark data (EMNIST).

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

In order to compare the performance of simplex codes and exhaustive codes, we conducted analysis and experiments using artificial data $(\mu, \sigma^2) = (1.0, 0.1)$, (6)-(9), and benchmark data (EMNIST). The results are summarized in Table V for each M. Here, in the case of artificial data, they are used to compute the probability of error by the error function. We also summarized the results obtained by the benchmark data for each M by selecting a subset of them in Table V, where a DNN (Deep Neural Network) is used [6].

[6]A type of convolutional neural network with 5 layers is applied.

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

V. SYSTEM EVALUATION USING A TRADE-OFF FUNCTION

A. Trade-off Relationship

Generally speaking, 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) (See Appendix A, Table VI). Here, 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

In terms of trade-offs, pay attention to "If we tolerate a slight increase in investment cost, we can decrease the performance degradation drastically" [18]. This is called *flexible* (See Appendix A, Fig. 6 (1)). When the trade-off relationship is given by a convex downward curve of the ECOC for the system scale M, normalize it as $n = N/N_{\max}$ and $p_{ce} = P_{ce}/P_{ce,\max}$, and enable to relatively compare with any M. Here, $N_{\max}$ ($N_{\min}$) is the maximum value (minimum value) of the domain of the code length N of shortened exhaustive codes, $N_{\max} = 2^{M-1} - 1$ ($N_{\min} = \lceil \log_2 M \rceil$), and $P_{ce,\max} = 1/2$.

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

Hereafter, we note that N and P_{ce} are normalized by $N_{\max}$ and $P_{ce,\max}$, respectively. The performance of simplex codes for $M = 4$, $M = 12$, and $M = 16$, are illustrated by ■ together with the trade-off curves between n and p_{ce} for shortened exhaustive codes in Fig. 3. Regarding the values p_{ce}'s for the ■ of simplex codes be a constant for any M in Fig. 3, we have the relationships between M and n are as shown in Fig. 5.

D. Experimental Results by Benchmark Data

The performance of simplex codes for $M = 4, M = 12$, and $M = 16$, are also illustrated by ■ together with the trade-off curves between n and p_{ce} for shortened exhaustive codes in Fig. 4. By the similar assumption, we have the relationships between M and n are as shown in Fig. 5 which is the same as that for the artificial data, since the relationship between M and n is derived when the values of $p_{ce}{}'s$ for different n are regarded to be a constant for any M.

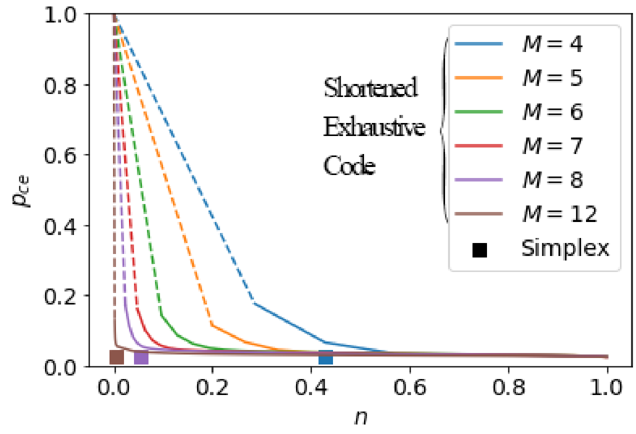

Fig. 3. Trade-off relationship between n and p_{ce} by the simplex codes and the shortened exhaustive codes for artificial data $(\mu, \sigma^2) = (1.0, 0.1)$.

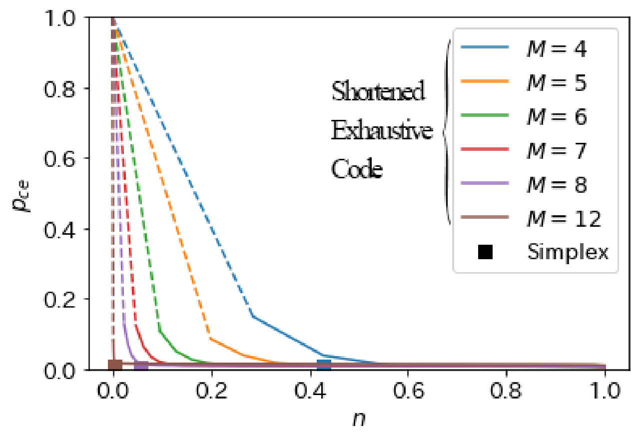

Fig. 4. The trade-off relationship between n and p_{ce} of the simplex codes and the shortened exhaustive codes for benchmark data (EMNIST).

VI. DISCUSSIONS

A. Performance Improvement of RM code by mRM code

As shown in Fig. 1 and Fig. 2, in both cases, the number of complements is increased from $s = 0$ (the case of the mRM code) to $s = 1, 2, 3$, and 4, by replacing the s codeword(s) of the mRM code with its (their) complement(s) for $M = 8$. The subcode of the RM code is constructed by this method with adding an overall parity, and P_{ce} is evaluated. As a result, it can be concluded that conversely the mRM code is obtained by removing the complement codeword from the RM code, thus the performance of the RM codes is improved by the mRM codes.

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

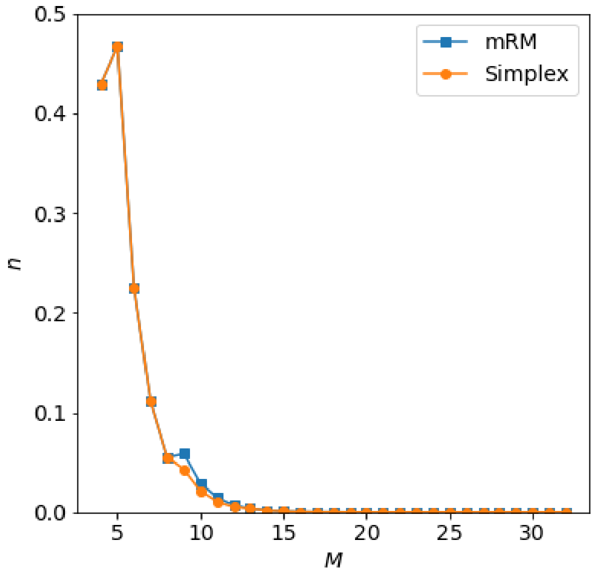

Fig. 5. The relationship between n and M of the mRM codes and the simplex codes for both the artificial data $(\mu, \sigma^2) = (1.0, 0.1)$ and the benchmark data (EMNIST) .

values for the simplex codes also very close to those of the exhaustive codes except in the case of the DNN.

C. Properties of Simplex Codes by System Evaluation Model

It is useful to discuss the performance of the simplex code along with that of the shortened exhaustive code in order to find out where the performance of the simplex code is in that of the shortened exhaustive code. Figs. 3 and 4 are for that purpose.

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

(i) For a given M, the simplex code p_{ce} (indicated by ■) is almost on the trade-off curve for the shortened exhaustive code. Furthermore, it is close to the minimum n of the actually valid shortened exhaustive code, which indicate that p_{ce} is increasing rapidly at n smaller than this. Therefore, simplex code is one of the important codes for which shortened exhaustive code exhibits *flexible* property (See Appendix A, Fig. 6 (1)).

(ii) As M becomes larger, the p_{ce} of the simplex code goes toward the origin. Referencing the trade-off curve of the shortened exhaustive code, the *elastic* property is satisfied (See Appendix A, Fig. 6 (2)). If we do not prepare a graph with larger scale on the horizontal axis, unfortunately, for $M = 16$ or more, they will overlap with the result for $M = 12$.

(iii) As M increases, the interval between the positions (indicated by ■) of p_{ce} decreases. Letting the values of p_{ce} be regarded as almost the same for any M, then Fig. 5 is obtained. The relationship between n and M is illustrated by a convex down curve, hence the ECOC by simplex codes has *effective elastic* property (See Appendix A, Fig. 6 (3)).

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. If $M = 2^m$, then there are $2^{m+1} - 2^m - 1 = 2^m - 1$ cases, where the simplex code cannot be available. On the other hand, if $M = 4\ell$, only the $4(\ell + 1) - 4\ell - 1 = 3$ is sufficient. This value in the former increases exponentially with increasing m, and is always 3 in the latter. A simple way to interpolate between them is to remove codewords (to construct a subcode). For example, if $M = 8$, we can use the $(7, \log_2 7, 4)$, $(7, \log_2 6, 4)$, and $(7, \log_2 5, 4)$ codes for $M = 7, 6$, and 5, respectively. 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. This brings a non-convex curve in Fig. 5 only at $M = 4$ and at $M = 8$ for the mRM code.

VII. Concluding Remarks

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 $N = 2^m - 1$. From the ECOC's point of view, it is very interesting that simplex code is better than the RM code which is one of the excellent codes from the viewpoint of theory of error correcting code. 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]. 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.

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

Acknowledgment

One of the authors S. H. would like to thank Professor Shin'ichi Oishi of Waseda University for giving a chance to study this work. The authors would like to thank to Professor Kenta Mikawa of Tokyo City University for his helpful suggestions to this research. The research leading to this paper was partially supported by MEXT Kakenhi under Grant-in-Aids for Scientific Research (B) No. 19H01721, (C) No. 19K04914, (C) No. 18K11585, and (B) No. 22H01055.

APPENDIX

A. System Evaluation Model Based on Trade-off Relationship

1) Trade-off model for QA systems: In the latter decade of 1970's through early 80's, J. Pearl and A. Crolotte discussed the trade-off between the amount of memory and the error in QA (Question Answering) systems based on rate-distortion theory [18].

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)

Rate-distortion theory discusses data compression by the trade-off relationship between rate and distortion [18]. The rate-distortion function can be written as:

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

2) Trade-off model for system evaluation: Generally, the rate R corresponds to the investment cost of a system, and distortion D, the performance degradation of the system (See Table VI). By extending the rate-distortion model, we have proposed the trade-off model for system evaluation [20], where we have also introduced a parameter G as the scale of the system. 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:

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

For evaluation of the systems, we define the following properties to the normalized trade-off system evaluation function (12):

Definition 2. (1) ***Flexible*** *[20]: The system is "flexible", if $r = r(d;G)$ is a decreasing and convex downward function of d. And the system A with $r = r_A(d;G)$ is more flexible than the system B with $r = r_B(d;G)$, if $r_A(d;G) < r_B(d;G)$ for arbitrary $d\,(0 < d < 1)$, and $G\,(G > 1)$. (See Fig. 6 (1)).*

(2) ***Elastic*** *[18]: The system with $r = r(d;G)$ is* ***elastic****, if $r = r(d;G)$ is flexible and a decreasing function of G for arbitrary $d\,(0 < d < 1)$. (See Fig. 6 (2)).*

(3) ***Effective elastic*** *[20]: The system is* ***effective elastic****, if the system is elastic and r is a convex downward function of G. (See Fig. 6 (3)).*

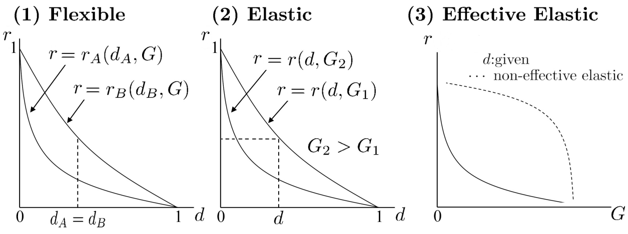

Fig. 6. Trade-off model for system evaluation

REFERENCES

[1] V. N. Vapnik and V. A. Vapnik, *Statistical Learning Theory*. Wiley, Sep. 1998.

[2] Q. Kang, L. Shi, M. Zhou, X. Wang, Q. Wu, and Z. Wei, "A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 9, pp. 4152–4165, 2017.

[3] H. Zhang, Y. Li, Z. Lv, A. K. Sangaiah, and T. Huang, "A real-time and ubiquitous network attack detection based on deep belief network and support vector machine," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 790–799, 2020.

[4] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[5] R. Rifkin, G. Yeo, and T. Poggio, "Regularized least-squares classification," *NATO Science Series Sub Series III*, 2003.

[6] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286, 1994.

[7] M. Goto and M. Kobayashi, *Introduction to Pattern Recognition and Machine Learning (in Japanese)*. Tokyo: Corona-Sha, 2014.

[8] E. R. Berlekamp, *Algebraic coding theory (revised edition)*. World Scientific, 2015.

[9] N. J. A. Sloane, *Short Course on Error Correcting Codes*. New York: Springer-Verlag, 1975.

[10] ——, "Library of Hadamard matrices," *http://neilsloane.com/hadamard/*.

[11] A.-R. Yin, X. Xie, and J. Kuang, "Using Hadamard ECOC in multi-class problems based on SVM," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[12] R. C. Bose and S. S. Shrikhande, "A note on a result in the theory of code construction," *Information and Control*, vol. 2, pp. 183–194, 1959.

[13] Z. Kiyasu, *Hadamrd Matrices and Their Applications (in Japanese)*. Tokyo: Corona-Sha, 1980.

[14] G. James and T. Hastie, "The error coding method and PICTs," *Journal of Computational and Graphical Statistics*, vol. 7, no. 3, pp. 377–387, 1998.

[15] E. J. Coyle, R. G. Roberts, E. G. Collins Jr., and B. A, "Synthetic data generation for classification via uni-modal cluster interpolation," *Autonomous Robots*, vol. 37, no. 1, pp. 27–45, Jun. 2014.

[16] G. Cohen, S. Afshar, J. Tapson, and A. V. Schaik, "EMNIST: Extending MNIST to handwritten letters," in *2017 Int. Joint Conf. on Neural Networks*, 2017.

[17] G. Kumoi, H. Yagi, M. Kobayashi, M. Goto, and S. Hirasawa, "Performance evaluation of ECOC considering estimated probability of binary classifiers," *Proc. 10th World CIST Conf. on Information Systems and Technologies*, 2022.

[18] J. Pearl and A. Crolotte, "Storage space versus validity of answers in probabilistic question answering systems," *IEEE Trans. Inform. Theory*, vol. IT-26, no. 6, pp. 633–640, Nov. 1979.

[19] G. Kumoi, H. Yagi, M. Kobayashi, and S. Hirasawa, "Effect of Hamming stance on performance of ECOC with estimated binary classifiers," *accepted for presentation at the 2022 Int. Conf. on Systems, Man, and Cybernetics*, 2022.

[20] S. Hirasawa and H. Inazumi, "A system evaluation model by using information theory," in *The 30th Joint National Meeting, MB35.3*, 1990.