情報処理学会第84回全国大会
2022年3月3-5日，愛媛大学，愛媛
ハイブリッド開催

構成的符号化を用いたECOCの一構成法（続）

早稲田大学　平澤　茂一
早稲田大学　雲居　玄道
電気通信大学　八木　秀樹
湘南工科大学　小林　学
早稲田大学　後藤　正幸
青山学院大学　稲積　宏誠

目次

雲居玄道，八木秀樹，小林学，後藤正幸，平澤茂一，“構成的符号化を用いたECOCの一構成法，”
日本経営工学会 2021年 春季大会，予稿集，F10, pp.371-372, 2021年5月15-16日.

1 はじめに

多値($M \geq 3$)分類問題 M : カテゴリ数

・2値分類器の多値化・・・Support Vector Machine (SVM),
Relevance Vector Machine (RVM)など

・Error-Correcting Output Codes (ECOC)

構成的符号化・・・Reed-Muller (RM) 符号
・修正RM符号
・Hadamard 行列
・Simplex 符号

トレードオフモデルによるシステム評価 [5][6]

2 符号語表

2.1符号語表の構成と性質

符号語表（Codeword Table）:

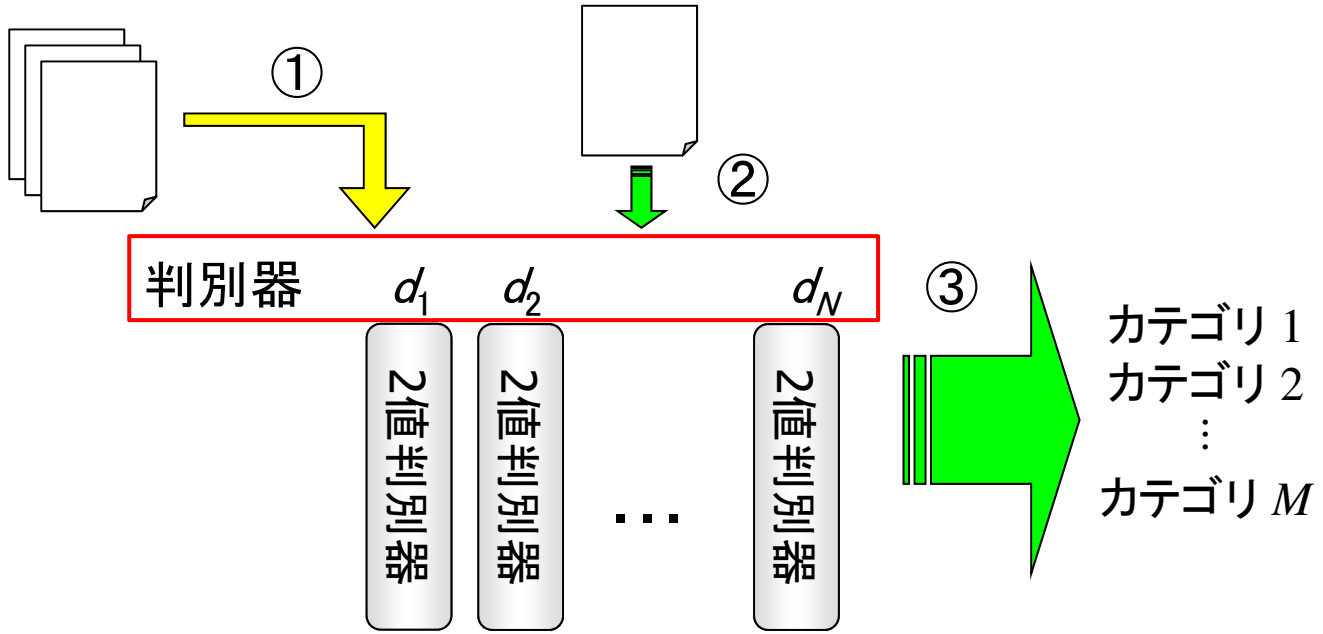
$$W = [w_{ij}] \in \{0, 1\}^{M \times N}$$

W の第i 行を$\boldsymbol{c}_i = (w_{i1}, w_{i2}, \cdots, w_{iN})$, 第 j 列を$\boldsymbol{d}_j = (w_{1j}, w_{2j}, \ldots, w_{Mj})^{\mathrm{T}}$ で表す. ここで, Tはベクトルの転置を示す.

[補 元] 任意の長さ L の 2 元 ベ クト ル $\boldsymbol{u} = (u_1, u_2, \ldots, u_L)$ に対し, $u_\ell \oplus u_\ell^{\mathrm{C}} = 1$ を満たす要素 u_ℓ^{C} $(\ell = 1, 2, \ldots, L)$ を持つベクトル $\boldsymbol{u}^{\mathrm{C}}$ $= (u_1^{\mathrm{C}}, u_2^{\mathrm{C}}, \ldots, u_L^{\mathrm{C}})$ をベクトル $\boldsymbol{u}$ の補元 (Complement) と呼ぶことにする. ここで, 演算 $\oplus$ は排他的論理和を 示す

多値分類システム構成部のモデル

訓練データ（文書集合）$(\boldsymbol{x},\ \mathrm{c}_i)^D$　　未知データ（文書）$\boldsymbol{y}$



［例］

符号語表

	d_1	d_2	d_3	d_4	d_5	d_6	d_7
C_1	0	0	0	0	0	0	0
C_2	0	0	0	1	1	1	1
C_3	0	1	1	0	0	1	1
C_4	1	0	1	0	1	0	1

カテゴリC_1・C_2
と
カテゴリC_3・C_4
の2値判別器

2.2 Exhaustive 符号

① 長さ M の 2^m $(m \geq 2)$ 個の列ベクトル $\boldsymbol{d}_j$ に対し，補元 $\boldsymbol{d}_j^{\mathrm{C}}$ を除去する.
② 全 0（または全 1）の列ベクトルを除去する.

得られた $N_{\max} = 2^{m-1} - 1$ の $M \times N$ の符号語表は

$(N_{\max}, \log_2 M, (M+1)/2)$ Exhaustive 符号

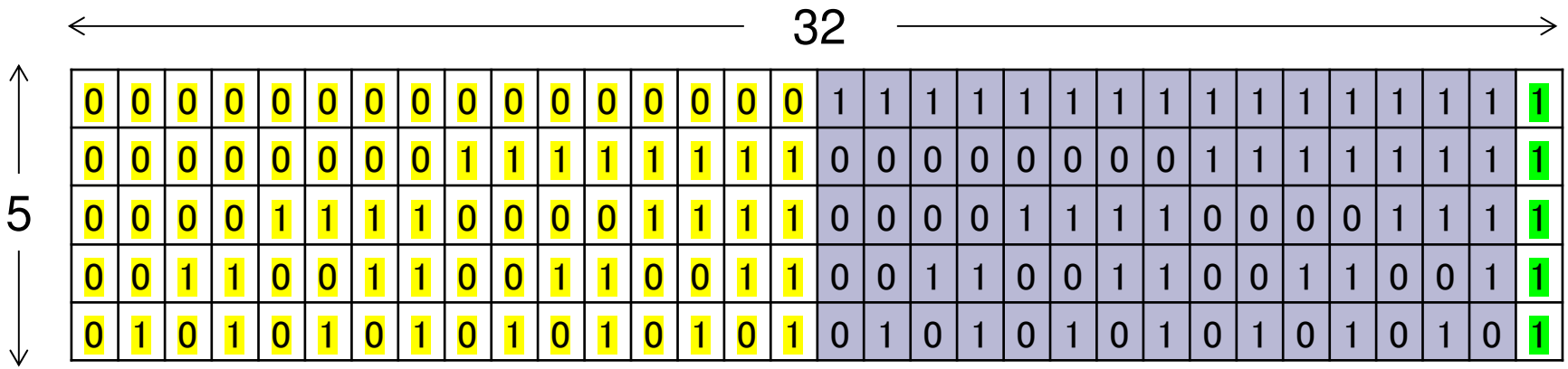
を与える.

ただし，符号長 N, 情報記号数 K, 最小設計距離 D の符号を (N, K, D) 符号と 示す. ここで,

① は負例 (0) と正例 (1) を入れ替えれば，同一 の判別領域を持つ,
② は明らかに判別に寄与していない という意味で冗長である.

[例 2.1] Exhaustive 符号の生成 (M=5)

全てのM[ビット]の列ベクトル$\boldsymbol{d}_j$ (j=1,2,…, 2^M) を生成し，その中から
① 補元を取り除く.
② その後，全1（または全0）の列ベクトルを取り除く.



$$N = 2^{M-1} - 1$$

2.3 修正 Reed-Muller (RM) 符号 [1]

任意の正整数 m (≥ 2) に対し,

$(2^m, m+1, 2^{m-1})$ 1 次 Reed-Muller (RM) 符号

が存在する. ここで, $2M = 2^{(m+1)}$ となる RM 符号を生成し,

① 長さ N の行ベクトル $\boldsymbol{c}_i$ の補元 $\boldsymbol{c}_i^{\mathrm{C}}$ を除去する.
② 全 0(または全 1)を列ベクトル除去する.

得られる修正 RM 符号の M 行 $N(=M-1)$ 列の符号 語表は

$(M-1, \log_2 M, M/2)$ 修正 RM 符号

を与える. ここで,

① はどの d_j ($j = 1, 2, \ldots, N$) においても, $\boldsymbol{c}_i$ と $\boldsymbol{c}_i^{\mathrm{C}}$ は別々のカテゴリとして学習され性能を劣化させ,
② は分類に寄与しないという意味で冗長である [1].

[例　2.2] 修正RM符号 （M=8）

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

(8,4,4)RM符号の符号語表

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

(7,3,4)修正RM符号の符号語表

3 構成的符号化に基づく符号語表

3.1 修正 RM 符号と Hadamard 行列

- 修正 RM (mRM) 符号
  - ・等距離符号
    Plotkinの上界式: $D \leqq N M (q\text{-}1)/(M\text{-}1)q.$
    (q=2のとき, $NM/2(M\text{-}1) = M/2$, $N=M\text{-}1$)
  - ・$N=2^m\text{-}1$

- Hadamrd 行列
  - ・$H_M \in \{-1, +1\}^{M\times M}$
    -1→0
    +1→1
    ($M = 2^m$のとき) 全0列ベクトルを除去 → 修正RM符号
  - ・$M=4\ell$ (<1000, ただし668, 716, 892を除く) のとき, H_M が存在する.

3.2 Simplex 符号

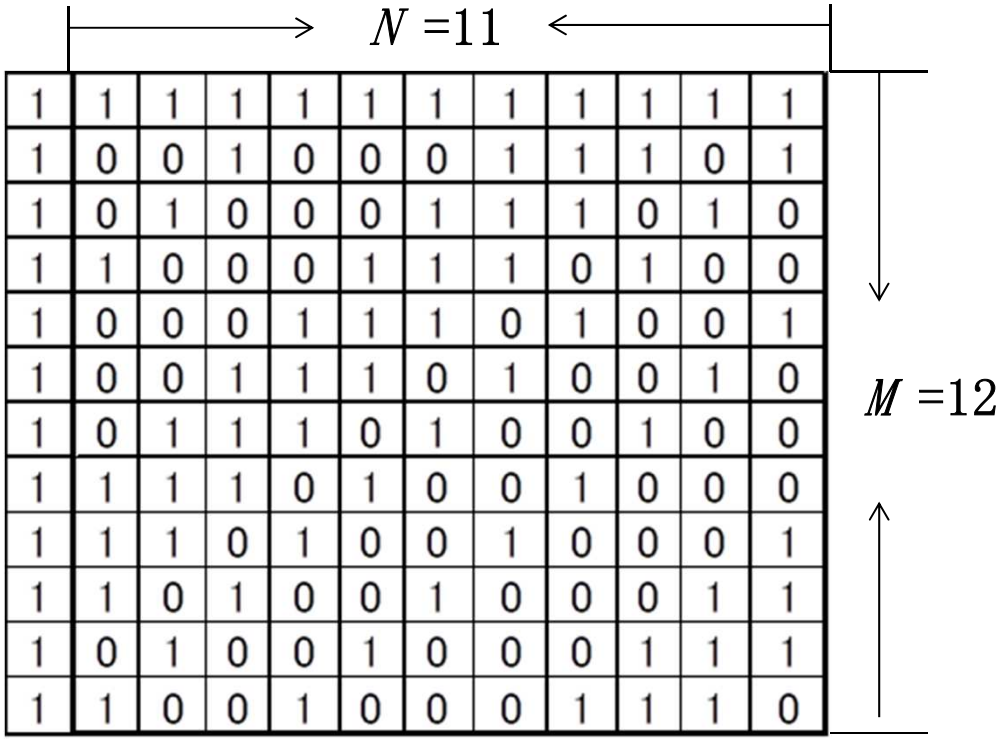
$(N, \log_2(N+1), (N+1)/2)$ Simplex 符号の生成法

- $(2^m - 1, 2^m - 1 - m, 3)$ Hamming 符号の $(2^m - 1, m, 2^{m-1})$ 双対符号
  ↓
- 修正 RM 符号は Simplex 符号の構成法を与える.
- Hadamard 行列
  ↓
- $M = 2^m$ (2の冪乗) → $N = 2^m - 1$
- $M = 4\ell$ ($\ell \geq 3$) (4の倍数) → $N = 4\ell - 1$

- M＝4, 8, 16, 32, …
- M＝12, 16, 20, 24, 28, 32,…

M =4, 5, 6, 7, 8,　　M =12, 13, 14, 15, 16,　　← 部分符号で補間

例 3：Hadamard 行列（M=12）から得られるSimplex符号



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

(11, $\log_2 12$, 6) Simplex符号による符号語表

3.3 分類性能の解析 [4]

テストデータ: $\boldsymbol{x}$

$\boldsymbol{d}_j$ の出力: $f_j(\boldsymbol{x}) = \Sigma_i w_{ij} \Pr(\boldsymbol{c}_i|\boldsymbol{x})$ (2)

推定カテゴリ: $\boldsymbol{c}_{\hat{\imath}}$

$$\hat{\imath} = \underset{i}{\operatorname{argmax}}\, g(\boldsymbol{c}_i|\boldsymbol{x}) \quad (1)$$

(N, K, D) Simplex符号の場合

$$g(\boldsymbol{c}_i|\boldsymbol{x}) = (M - D)[1 - \Pr(\boldsymbol{c}_i|\boldsymbol{x})] \quad (3')$$

ここで，事後確率 $\Pr(\boldsymbol{c}_i|\boldsymbol{x})$ は測定可能と仮定している.

4 トレードオフモデルを用いたシステム評価[5][6]

4.1 トレードオフ関係

表：多値分類システムの評価（対応表）

情報縮約論	システム評価モデル	多値分類システム
レート (R)	投資コスト(r)	2値判別器数 (n)
歪 (D)	性能劣化(d)	分類誤り確率 (p_{ce})
	規模(L)	カテゴリ数 (M)

性能劣化の評価基準：

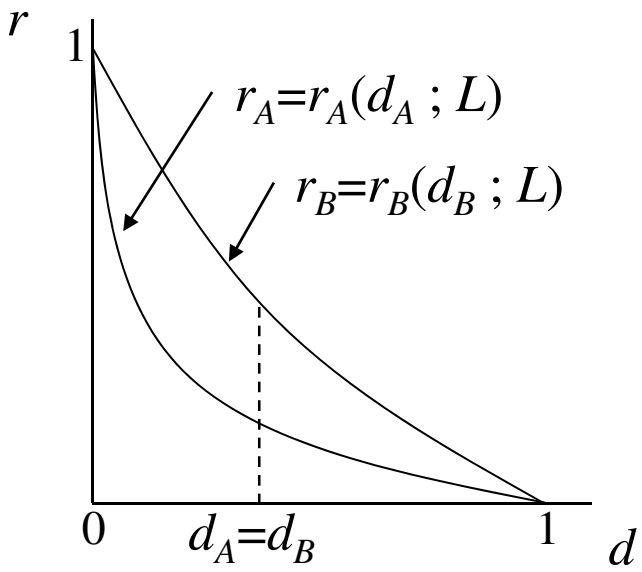
カテゴリ間の分類誤り確率（最悪値の下界，平均） P_{ce} (4)

トレードオフ評価モデル[5][6]

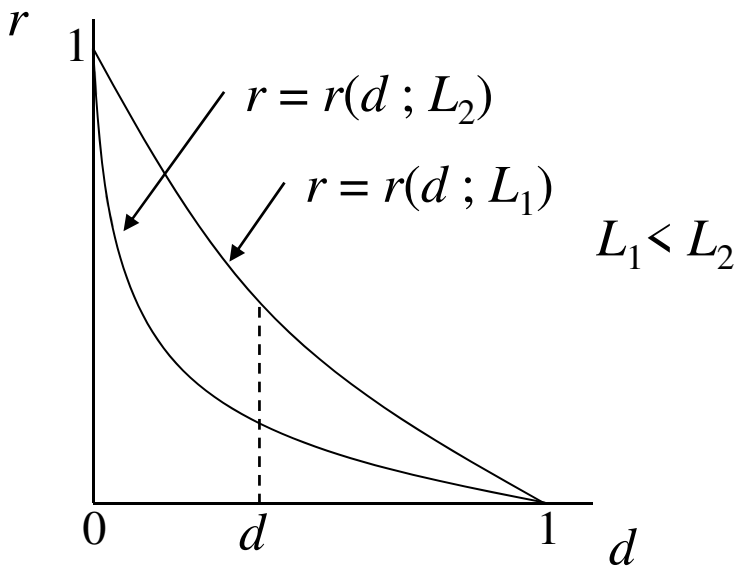
- 情報縮約論：レート歪関数　$R = R(D)$
- システム評価モデル：　$r = r(d\,;L)$
- $R\,(r)$　：レート
- $D\,(d)$：歪
- L：　システムの規模

（　）内は正規化した値

「フレキシブル」



「エラスティック」



4.2 システム評価

トレードオフ関係では「わずかな投資コスト増を許容 すれば，大きな性能劣化を改善できる」に注目する．こ れを「フレキシブル」という．

ECOC の与えられたシ ステムの規模 M に対しトレードオフ関係が下に凸な曲線で与えられるとき，

正規化： $n = N / N_{\max}$,

$p_{ce} = P_{ce} / P_{ce,\max}$

ここで， $N_{\max} = 2^{M-1} - 1$ $(N_{\min} = \lceil \log_2 M \rceil)$,

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

4.3 人工データ

M 次元多値分類データ：平均μ = 1.0, 分散σ^2 = 0.1（M 次元ガウス分布）

2値判別器：式(1)-(3')

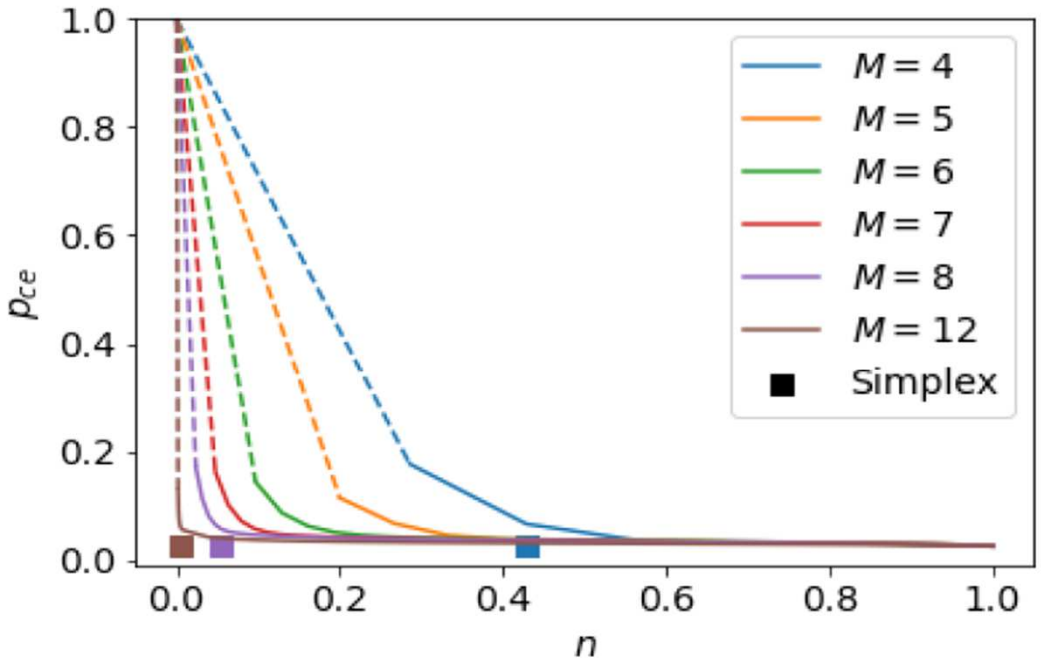


図１　人工データによるトレードオフ関係

4.4 実データ

実データ：書き数字・英文字（EMNIST）[7]

2値判別器：Deep Convolutional Neural Network

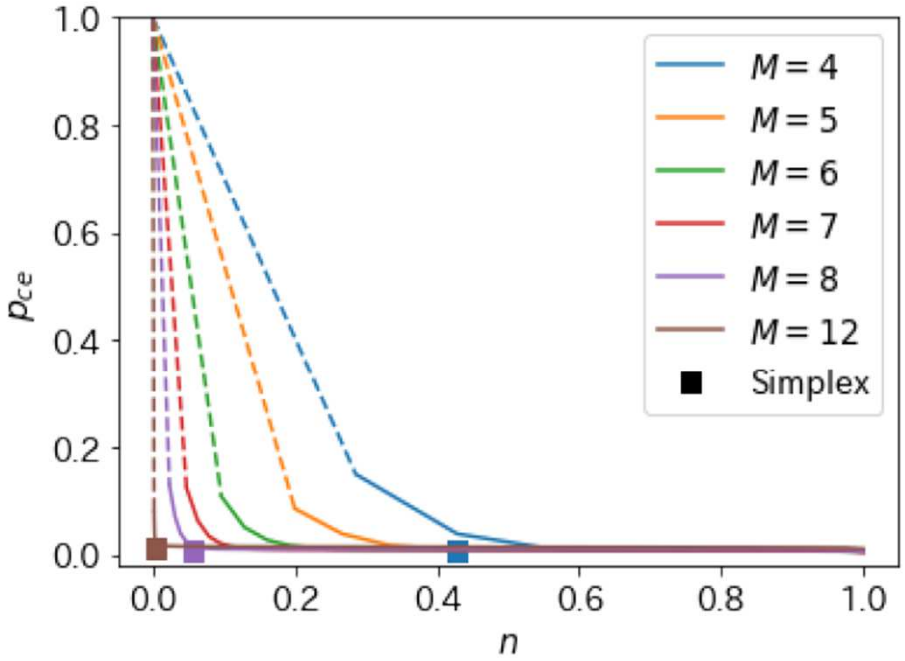


図 2　実データ EMNIST によるトレードオフ関係

5 考察

表 1　Simplex 符号と Exhaustive 符号による ECOC 法の性能（P_{ce}）

M	人工データ		実データ	
	Simplex	Exhaustive	Simplex	Exhaustive
4	0.0128	0.0128	0.00312	0.00271
8	0.0124	0.0124	0.00259	0.00246
12	0.0124	0.0124	0.00481	0.00506

表1より

1. いずれもExhaustive 符号に極めて近いp_{ce} を達成する.

次に図1, 2 より，最悪カテゴリ誤り確率の下界p_{ce} を示す短縮Exhaustive 符号の性質を参考に，■で示されたSimplex 符号について，以下の結果を得る.

2. Simplex 符号は，M の増加とともに原点方向に向かう「エラスティック」の性質を持つ.
3. Simplex 符号は，M によらずほぼ一定の小さなp_{ce}を持ち，その値はn が小とともに原点方向に向かう. その結果，n はM に対し下に凸な関数を与える. すなわち，「効果的エラスティック」の性質を持つ.

6 むすび

(1)修正RM符号（ECOC の性能改善）
線形距離符号であり，$N = 2^m - 1$のSimplex 符号の生成法の一つであることを示した.
(2)Simplex 符号はHadamard 行列からも生成できることが知られており，そのため符号語数が$M \leq 1000$（ただし，668, 716, 892 を除く）の例から多値分類問題の解決には十分実用化可能である.
(3) Simplex 符号の優れた性質をシステム評価の見方から明らかにし，「エラスティック」「効果的エラスティック」の性質を持つことを示した.

なお，Simplex 符号と他の優れた符号を組み合わせた符号語構成法については今後の課題である.