同一カテゴリ内での二値判別を許容する符号表に基づく ECOC 多値判別法に関する研究

情報数理応用研究　5215C012-3　鈴木玲央奈
指導教員　後藤正幸

A Study of ECOC Multi-Category Classification Approach Based on Code Table Considering Binary Classification in Same Category

SUZUKI Leona

1　研究背景・目的

カテゴリ数が 3 以上の分類を扱う多値判別問題は，広く適用範囲を持つ重要な課題である．多値判別問題の解法には，様々なアプローチが存在するが，本研究では強力な二値判別器を組み合わせることで新規データのカテゴリを推定する Error-Correcting Output Coding 多値判別法 [1]（以下，ECOC 法）に着目する．ECOC 法は，符号表生成と復号の二つのステップから構成される．符号表生成のステップにおいては，各行にカテゴリ，各列に二値判別器の構成を表した符号表と呼ばれる数値表を生成する．判別を意味する復号のステップでは，新規データに対する複数の二値判別器の出力を統合したベクトルを用いて，新規データの所属カテゴリを推定する．

符号表生成に対するアプローチには，適応的に符号表を生成する方法と学習前に符号表の全構成を与える方法（非適応的な方法）の二通りが存在する．前者では，判別器の学習結果に応じて適応的に符号表の生成を行う．一方，後者の非適応的な方法では，判別器の学習を行う前に符号表を生成し，生成された符号表を固定する．この方法には，個々の二値判別器の学習に対して並列処理を行うことが可能となるという利点がある．

一方，判別精度の向上を目的とした符号表生成の手法として，サブクラスと呼ばれるカテゴリをデータの特徴により分割した部分集合を活用する手法 [2] がある．カテゴリのサブクラスへの分割は，各二値判別器における分類を容易とし，分類精度の向上を可能とする．しかしながら，サブクラスを用いる手法の多くは，二値判別器の学習を行いながら符号を構成していく適応的な手法であるため各判別器の学習に対して並列処理ができない．また，同一カテゴリ内に属する異なるサブクラス間では二値判別を行わないという制約を持つ．

本研究では，分類精度の向上のために，符号表生成と復号の双方のステップに着目した，新たな ECOC 多値判別法を提案する．まず，符号表生成では，データに対して適応的にサブクラスを形成しつつ，得られたサブクラスを基にし，符号表を判別器の学習を行う前に生成する．これにより，データに対して適応的であり，なおかつ二値判別器の学習に対して並列処理が可能となる適応的な生成法と非適応的な生成法の双方の長所を有する生成アルゴリズムを実現する．その際，各行が 1 つの学習データに対応する符号表を用い，同一カテゴリ内での二値判別を許容することで，上記の同一カテゴリ内で二値判別を行わないという制約を解消する．これにより，分類が容易である部分データを判別する精度の高い二値判別器を多数生成し，全体としての多値判別の精度向上が期待できる．さらに，復号のステップにおいて，提案する各行が 1 つの学習データに対応する符号表に適した推定方法を提案する．上記二つの提案の有効性についてベンチマークデータを用いた多値判別問題に適用し，検証を行う．

2　ECOC 法

本研究では，入力 $\boldsymbol{x}$ に対し，その所属カテゴリ $c_k(1 \le k \le K)$ を推定する問題を扱う $(K \ge 3)$．ECOC 法は，符号理論で用いられる誤り訂正技術を多値判別問題に応用した手法であり，カテゴリが未知の新規データに対し，複数の二値判別器を組み合わせることにより，所属カテゴリを推定する．複数の二値判別器の構成は符号表と呼ばれる {1,0} の二値を要素とする数値表により表現される．いま，二値判別器の個数を R とすると，符号表 $\boldsymbol{W}$ は $K \times R$ 行列で与えられる．ここに，符号表 $\boldsymbol{W}$ の各列ベクトルは二値判別器の構成を表現しており，要素が 1 のカテゴリ集合と要素が 0 のカテゴリ集合を二値判別するための R 個の判別器が学習される．そのため，0 と 1 が反転した列ベクトルは，等価な二値判別器を意味する．また，符号表 $\boldsymbol{W}$ の k 行目の行ベクトルをカテゴリ c_k の符号語と呼び $\boldsymbol{w}_k$ で表す．新規データの所属カテゴリを推定する際には，新規データに対する二値判別器の出力結果ベクトルと各カテゴリの符号語の比較により，分類を行う．

符号表の中には {1,0} の二値で表される二元符号表の他に，判別に用いないカテゴリを許容した三元符号表がある．ここでは，判別に用いないカテゴリを $*$ で表す．$\boldsymbol{w}_k$ の r 番目の値を w_k^r とすると，w_k^r が $*$ の場合には r 番目の判別器を学習する際に，カテゴリ c_k の学習データは除外される．そのため，三元符号表は各二値判別器で用いる学習データ数を減少させ，二元符号表に対して学習計算量の低減を可能とする．

3　従来手法

3.1　従来の符号表生成

本研究で扱う非適応的な符号表生成手法の代表的なものとして，一対他法と Exhaustive 符号 [1] がある．一対他法とは，1 つのカテゴリとそれ以外を二値判別する判別器をカテゴリ数分用いる符号表構成であり，$R = K$ となる．Exhaustive 符号とは，Dietterich らによって示された符号表であり，$2^{K-1} - 1$ 個の考えられる全ての 2 群分類に対する判別器を用意する判別器構成となっている．

表 1 は，$K = 4$ の時の Exhaustive 符号の例である．判別器数は $2^{4-1} - 1 = 7$ となる．Exhaustive 符号は判別器数が非常に多いため高い分類精度となる一方で，K が大きいと判別器の学習に膨大な時間を要してしまう．

表 1: Exhaustive 符号（$K = 4$）

		判別器						
		1	2	3	4	5	6	7
カ	c_1	1	1	1	1	1	1	1
テ	c_2	0	0	0	0	1	1	1
ゴ	c_3	0	0	1	1	0	0	1
リ	c_4	0	1	0	1	0	1	0

3.2　従来の分類基準

従来の復号における分類基準として，類似度最大に基づく分類基準が存在する．入力 $\boldsymbol{x}$ に対する $r(1 \le r \le R)$ 番目の二値判別器の出力を $g_r(\boldsymbol{x})$ としたとき，類似度最大に基づく分類基準では，符号語 $\boldsymbol{w}_k$ と $\boldsymbol{g} = (g_1(\boldsymbol{x}), ..., g_R(\boldsymbol{x}))$ の類似度 $S(\boldsymbol{w}_k, \boldsymbol{g})$ が最大となるカテゴリ $c_{\hat{k}}$ の番号 $\hat{k}$ を式 (1) により導出し，分類する．

$$\hat{k} = \arg\max_k S(\boldsymbol{w}_k, \boldsymbol{g}) \tag{1}$$

また，類似度 S の代わりにハミング距離を用いて，ハミング距離が最小のカテゴリへと分類する，最小距離復号と呼ばれる手法も存在する．類似度最大に基づく分類基準，最小距離復号のどちらの手法も，二値判別器の出力例から，最も類似性の高いまたは最も距離が小さい符号語を一つ選ぶ操作となっている．

4　提案手法

4.1　提案手法の着眼点

提案手法では，符号表生成と分類基準の双方に着目し，新たな ECOC 法のアプローチを提案する．まず，符号表生成に関しては，サブクラスを用いた符号表生成を考える．サブクラスとは，カテゴリをデータの特徴により分割した部分集合であり，例えば，新聞記事における「スポーツ」カテゴリには，「プロ野球」,「J リーグ」,「オリンピック」などの異なる話題を持ったサブクラスが含まれると考えられる．従来の符号表ではこのサブクラスの存在は考慮されず，カテゴリ情報のみを用いて符号表の生成が行われている．

しかしながら，上記の例のように同一カテゴリ内に性質が異なる複数のサブクラスが存在する場合に加えて，異なるカテゴリに属するサブクラス同士の特徴が類似している場合が考えられる．例えば,「スポーツ」カテゴリの中でも,「オリンピック」の話題を持ったサブクラスは「国際」カテゴリにおける「オリンピック」関連のサブクラスと類似した特徴を持っているものと思われる．このような場合に，従来のカテゴリ情報のみを用いた符号表生成法では，同一カテゴリ内でのサブクラス間の特徴の違いだけでなく，異なるカテゴリ間でのサブクラスの特徴の類似性を考慮できず，多値判別の精度向上に貢献するような二値判別器の構成が困難となる可能性がある．そこで，カテゴリをデータの特徴からいくつかのサブクラスへと分割し，そのサブクラスを用いた符号表を生成することにより，上記のような複雑な構造を持つデータに対応しうるアプローチを提案する．

ここでカテゴリ c_k から分割された $j(1 \leq j \leq C)$ 番目のサブクラスを $c_{k,j}$ とすると，サブクラスを用いた符号表は表 2 のようになる．このように，各サブクラスに 1 つの符号語を対応させることで，同一カテゴリ内においても，サブクラスごとに異なるカテゴリ集合とみなして二値判別を行うための判別器を構成することが可能となる．

表 2: サブクラスを用いた符号表

		判別器				
c_k	$c_{k,j}$	1	2	3	$\cdots$	R
c_1	$c_{1,1}$	1	0	0	$\cdots$	$*$
	$c_{1,2}$	$*$	$*$	0	$\cdots$	0
	$c_{1,3}$	$*$	1	$*$	$\cdots$	0
c_2	$c_{2,1}$	0	0	1	$\cdots$	1
	$c_{2,2}$	0	$*$	1	$\cdots$	$*$
	$c_{2,3}$	1	1	$*$	$\cdots$	1
c_3	$c_{3,1}$	1	0	0	$\cdots$	$*$
	$c_{3,2}$	1	$*$	0	$\cdots$	0
	$c_{3,3}$	$*$	1	$*$	$\cdots$	0

このように，提案する符号表は，各カテゴリが複数の符号語を持つ．しかしながら，従来の符号表では，各カテゴリは単一の符号語を持ち，分類基準に関してもこれを前提とした分類基準が適用されている．そこで本研究では，各カテゴリが複数の符号語を持つ符号表に適した分類基準を考えることにする．

4.2　符号表生成アルゴリズム

この節では，提案する符号表生成アルゴリズムについて述べる．符号表生成アルゴリズムは，1) ランダム性のあるサブクラス生成，2) 二値判別器構成の決定（$0, 1, *$ の割り当て）で構成され，1) と 2) の繰り返しにより符号表を生成する．まず，サブクラスの生成では，各カテゴリごとに予め決められた数のサブクラスを構成するための代表ベクトルをランダムに選択し，代表ベクトルとの距離による学習データのクラスタリングにより，サブクラスを生成する．これにより，データの特徴からカテゴリをサブクラスへと分割することが可能となる．

次に，2 つ目の手順の二値判別器構成の作成について述べる．本研究では，個々の二値判別器の精度を向上させるために，同一カテゴリ内で二値判別を許容する符号表の生成を行う．しかし，同一カテゴリ内で二値判別を許容する符号表のみを用いると，カテゴリ間での識別の劣化が懸念される．そこで，同一カテゴリ内で二値判別を許容する符号表と（従来の考え方に基づく）同一カテゴリ内で二値判別を許容しない符号表との 2 つを作成し，その二つを適度にマージすることで分類精度の向上を図る．

4.2.1　同一カテゴリ内で二値判別を許容しない符号表の生成

まず，同一カテゴリ内で二値判別を許容しない判別器構成の決定では，$*$ を用いてサブクラス間で符号語に差異が生じるように判別器構成を表現する．

表 3 は，サブクラスを用いた同一カテゴリ内での二値判別を許容しない符号表の例である．c_1 に着目すると，各サブクラスごとで符号語に差異はあるものの，1 または $*$ が割り当てられており，$0, 1$ の割り付けによる二値判別は行われていない．一方で，c_1 と c_2 を比較すると，$0, 1$ の割り当てにより二値判別が行われていることがわかる．ここで述べた二値判別器構成の決定手順は，4.2.4 節で述べるアルゴリズムの Step2 にあたる．

表 3: 同一カテゴリ内での二値判別を許容しない符号表

		判別器			
c_k	$c_{k,j}$	1	2	3	$\cdots$
c_1	$c_{1,1}$	1	1	$*$	$\cdots$
	$c_{1,2}$	1	$*$	1	$\cdots$
	$c_{1,3}$	$*$	1	1	$\cdots$
c_2	$c_{2,1}$	0	0	$*$	$\cdots$
	$c_{2,2}$	0	$*$	0	$\cdots$
	$c_{2,3}$	$*$	0	0	$\cdots$
			$\vdots$		

4.2.2　同一カテゴリ内で二値判別を許容する符号表の生成

同一カテゴリ内で二値判別を許容する判別器構成の決定では，ランダム生成で得られるサブクラスを，サブクラス間の距離によりグルーピングをし，サブクラスグループを生成する．さらに生成されたサブクラスグループをそれぞれ別々のカテゴリとみなし，従来のカテゴリ集合間の二値判別による判別器構成と同様の割り当てを行い，符号表を生成する．これにより，異なるカテゴリに属するサブクラスであっても，サブクラス同士の特徴が類似したものは同一のグループ，同一カテゴリ内でも特徴が異なるものは違うグループとみなした符号表を生成することが可能となる．

表 4 は，同一カテゴリ内での二値判別を許容する符号表の例である．以下では，c' をサブクラスグループとする．ここで，c'_1 に着目すると c'_1 には，$\{c_{1,1}, c_{2,1}, c_{2,2}\}$ が含まれており，異なるカテゴリのサブクラスが同一のグループに属している．また，$\{c_{1,1}\}$ と $\{c_{1,2}, c_{1,3}\}$ は同一カテゴリでありながら，異なるグループに属している．このようなグループを生成することにより，$\{c_{1,1}\}$ と $\{c_{1,2}, c_{1,3}\}$

の間で二値判別が行われている．このように，サブクラスグループを用いることで，サブクラス間の類似度を考慮した同一カテゴリ内での二値判別を許容する符号表の生成が可能となる．ここで述べた二値判別器構成の決定手順は，4.2.4 節で述べるアルゴリズムの Step3 にあたる．

表 4: 同一カテゴリ内での二値判別を許容する符号表

c'	$c_{k,j}$	判別器 1	2	3	$\cdots$
c'_1	$c_{1,1}$	1	1	$*$	$\cdots$
	$c_{2,1}$	1	$*$	1	$\cdots$
	$c_{2,2}$	$*$	1	1	$\cdots$
c'_2	$c_{1,2}$	0	0	$*$	$\cdots$
	$c_{1,3}$	0	$*$	0	$\cdots$
	$c_{2,3}$	$*$	0	0	$\cdots$
			$\vdots$		

4.2.3 符号表生成の繰り返しと結合

4.2.1 節と 4.2.2 節の手順により，カテゴリ情報とサブクラスの特徴を考慮した 2 種類の符号表生成が可能となる．しかし，生成される符号表の判別器数が少ない場合には，分類精度の低下が懸念される．そのため，2 種類の符号表の作成を，判別器数が所望の大きさとなるまで繰り返す．これにより様々なサブクラスに対する二値判別器を十分な数だけ持つ 1 つの符号表の生成が可能となる．

しかし，各繰り返しで生成されるサブクラスは異なるため，同じ行番号であっても同一のサブクラスとは限らず，繰り返しによって得られた複数の符号表を単純に連接させることができない．そこで提案手法では，行数が総学習データ数となる符号表に変換を行うことでこれを解決する．具体的には，各学習データの符号語はその学習データの属するサブクラスの符号語と同一のものにする．これにより，学習データごとに符号語の追加を行うことが可能となる．またサブクラスグループを用いて符号表を生成する際には，あらかじめ K 行の符号表を準備しそれに基づき生成する．この際，準備する符号表は従来手法と同一のものでよく，様々な特徴を持つ符号表を本提案手法に容易に適用することが可能である．ここで述べた繰り返しと結合による符号表生成は，4.2.4 節で述べるアルゴリズムの Step4 にあたる．

4.2.4 提案符号表生成アルゴリズム

以下では全学習データ数を N, あらかじめ準備する K 行の符号表を $\boldsymbol{H}$ とする．

Step0 基となる符号表 $\boldsymbol{H}$ の構成

Step1 サブクラスのランダム生成

Step1-1 サブクラスの代表ベクトルの選択　各カテゴリ c_k において，ランダムに C 個の学習データを選択し，それを各サブクラスの代表ベクトルとする．

Step1-2 サブクラスの生成　各カテゴリ c_k において，サブクラス $c_{k,j}$ の代表ベクトルと学習データとの距離を計算し，代表ベクトルとの距離が小さい学習データをサブクラスへと分割する．

Step2 同一カテゴリ内で二値判別しない割り当て
各カテゴリ c_k において，ランダムに一つのサブクラスを選択し，そのサブクラスは判別に用いないことにする．つまり符号語要素を $*$ とする．その他のサブクラスはあらかじめ準備した $\boldsymbol{H}$ と同様の割り付けとする．上記の操作を，すべてのサブクラスが 1 回ずつ選択されるまで行うことにより，符号表 $\boldsymbol{H}_1$ を得る．

Step3 同一カテゴリ内で二値判別を許容する割り当て

Step3-1 サブクラスグループの代表ベクトルの選択　各カテゴリ c_k において，Step 2 で生成されたサブクラスの中から 1 つ選択し，選択されたサブクラスの代表ベクトルを各サブクラスグループの代表ベクトルとする．

Step3-2 サブクラスのグルーピング　サブクラスグループの代表ベクトルと各サブクラスの代表ベクトルとの距離を計算し，距離の近いグループへとサブクラスを分割する．

Step3-3 サブクラスグループに基づく符号表の生成　生成された K 個のサブクラスグループを予め準備した $\boldsymbol{H}$ の K 個のカテゴリとみなし，Step2 と同様に割り付けを行うことにより，符号表 $\boldsymbol{H}_2$ を得る．

Step4 繰り返しと符号表の結合　Step1 から Step3 を M 回繰り返す．各繰り返しで生成される 2 つの符号表 $\boldsymbol{H}_1$，$\boldsymbol{H}_2$ は，各行を学習データとした N 行の符号表へと変換される．各学習データの符号語は属するサブクラスの符号語とする．変換後の 2 つの符号表を並べて 1 つにしたものを $\boldsymbol{H}'$ とし，M 個の $\boldsymbol{H}'$ を並べてひとつの符号表とする．□

4.3 l 多数決による分類基準

上記の通り，提案手法で生成される符号表は，各カテゴリが複数の符号語を持つ．すなわち，新規データに対する判別器の出力と比較される各カテゴリのテンプレートが複数存在する．そのため，類似度が最も高い符号語のみを考慮して分類を行うよりも，複数の符号語の類似度を総合して分類を行うことにより，複雑な構造を持つデータに対して頑健な分類が可能になると考えられる．一方，従来の類似度最大に基づく分類基準は，各カテゴリに対応する符号語が 1 つしか存在しないことを前提とした分類基準であり，複数の符号語を持つ符号表に対する分類基準として適していない．

そこで，提案する分類基準では，新規データに対する判別器出力と全符号語との類似度を従来と同様に計算し，その中から類似度の高い上位 l 個の符号語の属するカテゴリの多数決によってカテゴリを推定する．

5 実験

提案した符号表生成アルゴリズムと分類基準の有効性を検証するために新聞記事を用いた分類実験を行った．実験データは 2010 年の毎日新聞記事から 9 カテゴリを使用した．学習データは各カテゴリ 200 件とし，テストデータは各カテゴリ 100 件とした．評価指標はテストデータに対する正解率，判別器の学習における計算時間とし，それぞれ 5 回の実験の平均値を用いた．二値判別器には RVM[3] を用いた．また，学習時間に関しては，各判別器の学習を並列処理した場合についても示す．また比較手法は，Exhaustive 符号を用いた方法と一対他に基づく方法，提案手法 1 として Step2 で生成された同一カテゴリ内で二値判別を許容しない符号表のみを用いた手法，提案手法 2 として Step3 で生成された同一カテゴリ内で二値判別を許容する符号表のみを用いた手法，提案手法 1 と提案手法 2 の両方を用いた手法を提案手法 3 とする．また，各提案手法で予め準備する符号表 $\boldsymbol{H}$ には，一対他法を用いた．各手法における二値判別器数を表 5 に示す．

表 5: 判別器数

符号表	判別器数 二元	判別器数 三元
Exhaustive 符号	255	-
一対他法	9	-
提案手法 1	-	$27 \times M$
提案手法 2	-	$27 \times M$
提案手法 3	-	$54 \times M$

提案手法 1，2 の繰り返し数 $M = 10$，提案手法 3 の $M = 5$，提案手法 1,2,3 のサブクラス数 $C = 3$，多数決数 $l = 3$ とした時の正解率と判別器の学習における計算時間の結果を表 6 に示す．

表 6: 正解率と計算時間

符号表	正解率	計算時間 (秒)	
		直列	並列
Exhaustive 符号	0.763	58,822	330
一対他法	0.723	1,259	230
提案手法 1	0.767	11,844	111
提案手法 2	0.776	15,711	113
提案手法 3	0.775	13,888	111

表 6 より，正解率は提案手法 2 と提案手法 3 が高くなり，学習の計算時間は通常の直列処理の場合は一対他が最も短く，並列処理の場合は提案手法が短くなった．

また，Exhaustive 符号に基づく方法と 3 つの提案手法を比較すると，判別器数はほぼ同程度である一方で，計算時間，正解率ともに提案手法の方が優れた値となった．計算時間に関しては，提案手法は三元符号表であるため，学習に使用しないデータの分だけ，計算時間が低減されたと考えられる．また，正解率に関しては，提案手法 2，提案手法 3 は一対他に基づく手法でありながら，高い正解率となった．

次に，多数決分類基準と繰り返しによる効果を確認するために，繰り返し数 M と多数決数 l を変化させた際の提案手法 1，2，3 の正解率の変化について示す．$l = 1, 3, 5$ と変化させ，M は，提案手法 1，2 と提案手法 3 の判別器数をそろえるために，提案手法 1，2 では，$M = 1 \sim 10$，提案手法 3 では $M = 1 \sim 5$ とした．図 1 が提案手法 1，図 2 が提案手法 2，図 3 が提案手法 3 の結果である．

まず，繰り返し数 M の変化が正解率に与える影響に着目する．提案手法 1 と提案手法 2 を比較すると，繰り返し数が大きい時，正解率の最大値は提案手法 2 の方が高いものの，繰り返し数が少ない時には，提案手法 1 の正解率が大きく上回っている．これは，提案手法 2 では，サブクラスグループを用いて，異なるカテゴリに属するサブクラスも同じカテゴリとみなした符号表生成を行うため，繰り返し数が少ない時には，異なるカテゴリ間で類似した符号語が多く存在することが理由だと考えられる．一方で，繰り返し数の増加に伴い，提案手法 1 では，比較的類似した符号表が追加されるのに対して，提案手法 2 では，様々なサブクラスグループに対して符号表が生成されるため，符号表の多様性が向上し分類精度が大きく向上したと考えられる．

また，提案手法 3 の結果に着目すると，小さい繰り返し数ですでに Exhaustive 符号と同程度の正解率となり，繰り返し数が 5 の時は，提案手法 2 と同様の高い正解率となった．このことから，提案手法 3 は，繰り返し数が小さいときにも比較的高い正解率となり，繰り返し数を大きくすることによってさらに高い正解率となる提案手法 1，2 のそれぞれの長所を有した手法といえる．

次に，多数決数 l の変化が与える影響について考察する．各図より，繰り返し数が小さいときに多数決数による正解率の差がみられる．すなわち，判別器数が少ないときに多数決による分類基準が有効に機能すると考えられる．また，提案手法 1 では，多数決数によって正解率に大きな差がみられない．提案手法 1 では，同一カテゴリ内では二値判別を行わないため，同じカテゴリに属している学習データに，類似した符号語が付与されたものと考えられる．よって，上位 l 個に含まれる符号語の属するカテゴリの多くが同一のカテゴリに限定され，多数決による効果が小さくなったと考えられる．提案手法 1 と 2 を組み合わせている提案手法 3 においても，同様のことが言える．

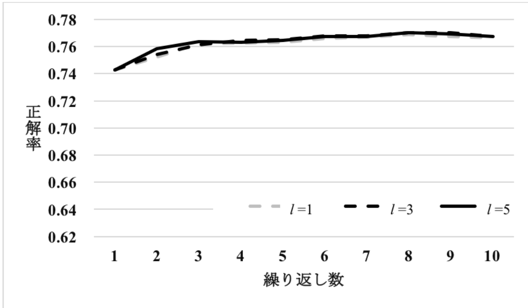

図 1: 提案手法 1 の正解率

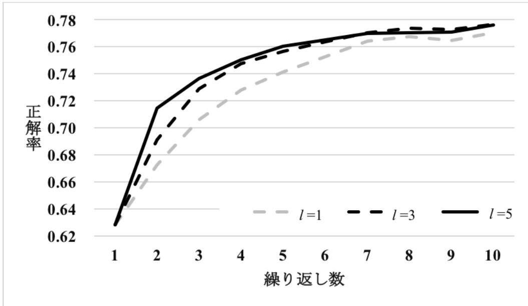

図 2: 提案手法 2 の正解率

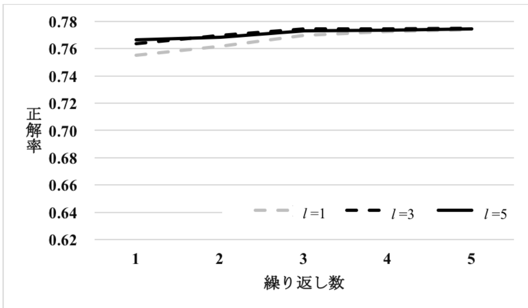

図 3: 提案手法 3 の正解率

6 まとめ

本研究では，多値判別問題を対象とし，同一カテゴリ内での二値判別を許容した符号表生成とそれに伴う多数決分類基準の提案を行った．実験結果より，計算量の増加を抑えつつ高い正解率が得られることを示した．今後の課題としては，サブクラス数の決定方法，ランダム性のない符号表生成などがあげられる．

参考文献

[1] T. G. Dietterich and G.Bakiri. “Solving Multiclass L.earning Problems via Error-Correcting Output Codes,” *Artif. Intell.*, vol.2, pp. 263–286, 1995.

[2] S. Escalera, D. M. Tax, O. Pujol, P. Radeva and R. P. Duin. ”Subclass Problem-Dependent Design for Error-Correcting Output Codes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol.30, pp. 1041-1054, 2008.

[3] M. E. Tipping. “Sparse Bayesian Learning and the Relevance Vector Machine,” *Mach. Learn. Res.*, pp. 211-244, 2001.