Vol.2017-CVIM-208 No.28
2017/9/16

Fish Bone Decision Tree：最大距離クラスを分離する決定木構築法とその応用

松尾大典† 和田俊和†

概要：決定木は，データ集合の分割を繰り返し，識別規則を学習する手法である．実ベクトル集合を対象とした場合は，射影や距離計算によって個々のデータに対するスカラ値を求め，その値と閾値の大小比較に基づいて，データ集合の分割が行われる．決定木は，クラスラベルの純度の上昇を表すエントロピー・ゲインまたは Gini ゲインを最大化するようにデータ集合の分割を進めていくため，葉に近い部分では少数のアンバランスなデータ集合に基づいて分割規則が学習される．この結果として末端部分での汎化性能が低下しやすいという欠点がある．これを補うために，ランダムにサンプリングされた複数のデータ集合から複数の木構造を作り識別を行うランダムフォレストがある．しかし，個々の決定木の性能がランダムフォレストの性能を規定するため，性能の良い決定木はやはり必要である．そこで，本報告では，最も分離が容易な２クラスに着目した決定木の構築法を提案する．この手法では着目した２クラス A,B を分離する射影軸と閾値を求めてデータ集合の分割を行い，次に分割された各データ集合内で A と else，および B と else の分割を行う射影軸と閾値の学習が行われる．この結果データは A, else, else, B の４クラスに相当する集合に分割されるが，２つの else は本来の目的であった A と B の分離の副産物であるため，再度一つに統合され過剰なデータ分割を抑制する．この結果，データ集合を３つに分割しようとする木が得られる.これが積み重ねられた決定木をその形状にちなんで Fish Bone Decision Tree (FBDT)と呼ぶ．本報告では，FBDT について，決定木としての性能と Forest の構成要素としての性能それぞれについて実験を通じて評価する．また，{else}部分は入力データが何の処理も受けずに流れるパスであり，３分木の各節での評価を並列に行うことも出来る．この結果，３進数を経由して各ベクトルに対応するスカラ値が求められる．このスカラ値にしたがって，ベクトルデータを数直線上に並べたとき，クラスのまとまりは保存されており，例えばファイル名にこの値を使えば，似たパターンが順に表示できるようになる．

キーワード：決定木，データ分割，コーディング

Fish Bone Decision Tree : method of divide most distinguish class

DAISUKEMATSUO† TOSHIKAZU WADA†

Abstract: Decision tree is a data structure for learning classification rules by recursively dividing given dataset. The data division for real vectors is realized by thresholding the scalar values obtained by inner product or distance computation with certain vectors. Since the division is conducted so as to maximize the purity measure represented by entropy or Gini gain, division rules near leaf nodes are learnt based on small unbalanced dataset. This limits the generalization power of the learnt rules near leaf nodes. For relaxing this problem, L. Breiman proposed random forest that uses decision trees learnt from randomly sampled datasets. However, we still need a good decision tree, because the performance of the forest is dominated by the accuracy of individual decision tree. For realizing better decision tree, this report proposes a decision tree construction method focusing on the most distinguishable class pair. This method first compute the projection vector that separates most distinguishable classes A and B. Each separated dataset is further separated into A and else or B and else. Two separated datasets elses are merged again, because they are the by-products of separating A from B. This merging has some effect for keeping generalization power. As a result, we get a trinary tree separating a dataset into three data sets at each node. We name it Fish Bone Decision Tree (FBDT) based on its shape. This paper reports the performance evaluation of FBDT as a single decision tree and a forest component through experiments. Also, FBDT can be used as a trinary cording by using each node as a primitive encoder. As a result, vectors can be mapped on to the scalar axis keeping the class label integrity. This method can be used in many visualization method. For example, the scalar can be used as file names of image vectors, that keeps the integrity of content similarity.

Keywords: decision tree, data partition, coding

1. 概要

決定木[1]は，データ集合の分割を繰り返し，識別規則を学習する手法である.実ベクトル集合を対象とした場合は，射影や距離計算によって個々のデータに対するスカラ値を求め，その値と閾値の大小比較に基づいて，データ集合の分割が行われる．決定木はデータ集合内のクラスラベルをできるだけ揃えるようにエントロピー・ゲインまたは Gini ゲインを最大化するようにクラスラベルの純度を高める分割を進めていくため，葉に近い部分では少ない，あるいはアンバランスなデータ集合に基づいて分割規則が学習される．この結果として末端部分での汎化性能が低下しやすいという欠点がある．これを補うために，ランダムにサンプリングされた複数のデータ集合から複数の木構造を作り識別を行うランダムフォレスト[2][3]がある．しかし，個々の決定木の性能がランダムフォレストの性能を規定するため，

† 和歌山大学院システム工学研究科
Wakayama University, Faculty of Systems Engineering

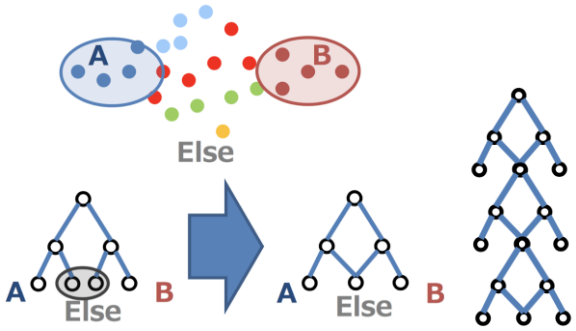

図１　Fish Bone Decision Tree の概要

性
能の良い決定木はやはり必要である．そこで，本報告では，最も分離が容易な２クラスに着目した決定木の構築法を提案する．この手法では着目した２クラス{A},{B}を分離する射影軸を求めてデータ集合の分割を行い，次に分割された各データ集合内で{A}と{else}，および{B}と{else}の分割を行う．この結果データは{A},{else}, {else}, {B}の４つに分割されるが，２つの{else}は着目した２クラス以外のデータの集合であり，本来の目的であった{A},{B}の分離の副産物であるため，再度一つに統合し過剰なデータ分割を抑制する．このようにして，{A},{B},{else}の３つに分割する木が得られる．これが積み重ねられた決定木をその形状にちなんで Fish Bone Decision Tree (FBDT)と呼ぶ．本報告では，FBDT について，決定木単体としての性能とフォレストの構成要素としての性能それぞれについて実験を通じて評価する．また，{else}部分は入力データが結果的には何の処理も受けずに流れるパスであり，３分木の各節での評価を並列に行うことも出来る．この結果，３進数を割り当てることで各ベクトルに対応するスカラ値が求められる．このスカラ値にしたがって，ベクトルデータを数直線上に並べたとき，クラスのまとまりは保存されており，例えばファイル名にこの値を使えば，似たパターンが順に表示できるようになるなどの応用が考えられる．

以下，第２章では関連研究，第３章で提案手法について説明し，第４章では実験，第５章では実験結果を示し，第６章では提案手法の応用を示し，第７章で提案手法の応用に関した実験を行う，第８章でまとめを述べる．

2. 関連研究

ここでは，ベクトルデータの特定の次元に着目して決定木を作成する CART 法[4]や C4.5[5]と，何らかの方法で求めたベクトルに対する射影成分の閾値処理で決定木を構築する手法について述べる．

CART や C4.5 では，ベクトルデータ$\boldsymbol{x}$の各要素のうちのどの次元dを用いてデータセットの分割を行えば，entropy gain や Gini gain など，クラスラベルの純度の指標が最も大きくなるかをテストし，その時の次元dと閾値θを用いてデータ集合の分割を行うというものである．以下に entropy gain を用いた場合の次元dと閾値θの求め方を示す．

ベクトルデータ集合をS_nとする．これを$S_n^{left}(d,\theta)$と$S_n^{right}(d,\theta)$という直和集合に分割する問題を考える．
$S_n = S_n^{left}(d,\theta) \cup S_n^{right}(d,\theta), S_n^{left}(d,\theta) \cap S_n^{right}(d,\theta) = \emptyset$
である．これらの分割された集合は，それぞれ

$$S_n^{left}(d,\theta) = \{\boldsymbol{x} \in S_n | x_d \leq \theta\} \quad (1)$$
$$S_n^{right}(d,\theta) = \{\boldsymbol{x} \in S_n | x_d > \theta\} \quad (2)$$

と表現することが出来る．

また，集合S_nは，個々のデータに付与されたクラスラベルcに基づいてC個の部分集合S_{nc}に分割することが出来る．

$$S_{nc} = \{\boldsymbol{x} \in S_n | class(\boldsymbol{x}) = c\}, \quad c = 1, \cdots, C$$

このとき，エントロピーは下記のように表すことが出来る．

$$Ent(S_n) = -\sum_{c=1}^{C} \frac{|S_{nc}|}{|S_n|} \log \frac{|S_{nc}|}{|S_n|} \quad (3)$$

エントロピー・ゲインとは，S_nに対するエントロピー$Ent(S_n)$と，分割後の各エントロピー$Ent(S_n^{left}(d,\theta))$と$Ent(S_n^{right}(d,\theta))$によって定義される量

$$\begin{aligned} &Egain(d,\theta) \\ &= Ent(S_n) \\ &- \frac{|S_n^{left}(d,\theta)|}{|S_n|} Ent\left(S_n^{left}(d,\theta)\right) \\ &- \frac{|S_n^{right}(d,\theta)|}{|S_n|} Ent\left(S_n^{right}(d,\theta)\right) \end{aligned} \quad (4)$$

であり，これを最大化するd,θがS_nに対して最も良い分割パラメータと見なされることが多い．すなわち，

$$(d_n^*, \theta_n^*) = \underset{(d,\theta)}{\operatorname{argmax}} Egain(d,\theta) \quad (5)$$

である．

これに対して，次式のように，ある射影ベクトル$\boldsymbol{a}$に対して射影し，同様にデータを分割する方法[2]もある．

$$S_n^{left}(\boldsymbol{a},\theta) = \{\boldsymbol{x} \in S_n | \boldsymbol{a} \cdot \boldsymbol{x} \leq \theta\}$$
$$S_n^{right}(\boldsymbol{a},\theta) = \{\boldsymbol{x} \in S_n | \boldsymbol{a} \cdot \boldsymbol{x} > \theta\}.$$

この場合，$\boldsymbol{a}$の解空間は非常に広いため，判別分析を用いる方法やランダムに$\boldsymbol{a}$を生成する方法などがある．もちろん，このような方法だけでなく，内積の代わりにカーネルを用いる方法や，距離計算を用いる方法など，ベクトルを対象とした決定木構成法については数多くの可能性があるが，これらについては過去の研究であまり詳しく調べられていない．

3. FBDT

本研究では，内積を用いてデータ集合を分割する方法について検討を行う．このため，射影軸$\boldsymbol{a}$をどのように求めれば良いかについて検討することが重要になる．

データセット名	訓練数	テスト数	次元数	クラス数
Shuttle	43,500	14,500	9	7
Mnist	60,000	10,000	784	10

表 1　実験に使用したデータセット一覧

データセット名	提案手法	CART 法	判別分析軸
Shuttle	99.83	99.98	99.67
Mnist	93.44	88.51	93.11

表 2　木単体での各手法の正答率（%）一覧

データセット名	提案手法（100）	提案手法（200）	ランダム軸*（100）	ランダム軸*（200）	ランダム軸**（100）	ランダム軸**（200）	CART 法（100）	CART 法（200）
Shuttle	99.90	99.90	99.73	99.73	99.77	99.78	99.98	99.98
Mnist	96.34	96.38	90.99	91.54	87.56	88.63	95.67	95.63

表 3　フォレストの要素とした場合の各手法の正答率（%）一覧
カッコ内はフォレストの木の本数，*はデータの次元数をDとすると$\sqrt{D}$分だけ射影軸を求めた場合，**は$1+\log_2 D$分の射影軸を求めた場合を表している．

射影軸$\boldsymbol{a}$の決定の際には「データ集合を２つに分割する」という明示的な意図がある．これに対して，分割の善し悪しを判定するエントロピー・ゲインは，全てのクラスのデータがどちらの集合に分かれたのかによって決まる．このため，データを二分しながら全クラスラベルを参照して閾値を決定しなければならなくなるという矛盾が生じる．このため，FBDT では，以下のようにデータの分割を行う．

1. まず，特定の２クラスに属するデータS_{nA*}，S_{nB*}のみに着目して射影軸$\boldsymbol{a}$を求める．さらに，そのデータのみに着目してエントロピーを計算し，各データの射影成分$\{\boldsymbol{a}\cdot\boldsymbol{x}\}$に対する閾値の決定を行い，データ集合$S_n$を$S_n^{left}$と$S_n^{right}$に分割する．
2. 次に，これまで無視してきたクラスラベルを持つデータとS_{nA*}, S_{nB*}のデータの分離を行うため，S_n^{left}とS_n^{right}内で，S_{nA*}と$S_n^{left}\setminus S_{nA*}$，および$S_{nB*}$と$S_n^{right}\setminus S_{nB*}$の分割を行うための射影軸と閾値の学習を行い，４つの集合$S_n^{left*}, S_n^{left}\setminus S_n^{left*}$，$S_n^{right}\setminus S_n^{right*}$，$S_n^{right*}$を得る．

この際に，どのようにして「特定の２クラス」に属するデータを決定するかが問題になる．様々な試行の結果，FBDT では以下のようにクラスを決定することとした．

各クラスのデータ集合S_{nc}について，その重心を以下のように求める．

$$\overline{\boldsymbol{x}_{nc}} = \frac{1}{|S_{nc}|}\sum_{\boldsymbol{x}\in S_{nc}} \boldsymbol{x}\,, \qquad c = 1,\cdots,C \tag{6}$$

これらの間で最も距離の開いているクラスのペアA^*, B^*にのみ注目して射影軸と閾値の決定を行う．

$$(A^*, B^*) = \underset{A,B}{\operatorname{argmax}} \|\overline{\boldsymbol{x}_{nA}} - \overline{\boldsymbol{x}_{nB}}\| \tag{7}$$

また，射影軸も単純に以下のように求める．

$$\boldsymbol{a} = (\overline{\boldsymbol{x}_{nA}} - \overline{\boldsymbol{x}_{nB}})/\|\overline{\boldsymbol{x}_{nA}} - \overline{\boldsymbol{x}_{nB}}\| \tag{8}$$

得られた４つの集合のうち，S_n^{left*}は，S_{nA*}を切り出そうとして得られた集合，S_n^{right*}は，S_{nB*}を切り出そうとして得られた集合であり，$S_n^{left}\setminus S_n^{left*}$ と $S_n^{right}\setminus S_n^{right*}$は副産物であるので，再びマージする．これによって，集合S_nを$S_n^{left*}, S_n^{else}, S_n^{right*}$の３つに分割するノードが得られる．

以上が，FBDT のおおまかな構築手続きであるが，FBDT をコーディングに用いることも出来る．これは，図２に示すように FBDT の各節をばらし，入力データに対して３進数を計算するというものである．このような計算を行う正当性は，else のノードを伝ってくるデータに対しては何らコミットされていないという FBDT の性質に依拠している．この結果，ベクトルデータが，クラスのまとまりを保ったまま数直線上にマッピングできるようになる．

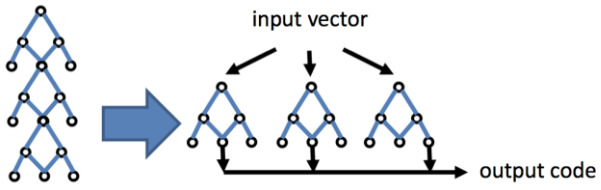

図 2 Fish Bone Decision Tree を用いたクラスの近接性を保つスカラ・コーディング

4. 実験

第３章で述べた提案手法によって構築された決定木の識別性能が既存の決定木と比べ改善されているのかを確認する．識別性能は決定木単体を構築した場合と，決定木の応用としてフォレストの構成要素とした場合とを確認し比較を行った．

提案手法と比較する対象としては，決定木単体としては，一般的な決定木構築法である CART 法と，提案手法と同様にクラス情報を用いて射影軸を求める方法である判別分析により求まる軸に基づいた木構築を行う手法の２つを考えた．実験で用いたプログラムだが，CART 法に関しては，公正を期すために Python のライブラリである scikit-learn に実装されているものを使用した．提案手法をフォレストの構成要素とした場合では，CART 法をフォレストの構成

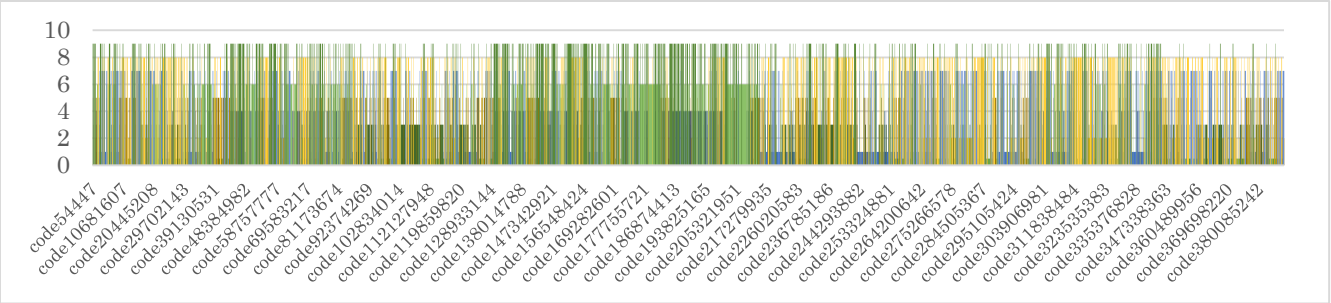

図 3　ランダムな軸を基に Mnist をコーディングした結果
縦軸はクラスラベル毎に振られた数字，横軸はコーディング結果を 10 進数に変換したもの

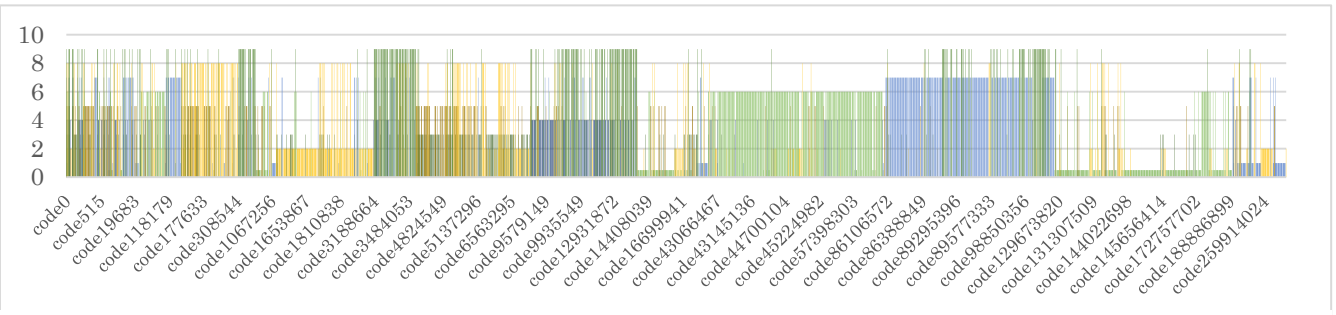

図 4　提案手法の内，束構造全てを用いた方法によって Mnist をコーディングした結果(1)
縦軸はクラスラベル毎に振られた数字，横軸はコーディング結果を 10 進数に変換したもの

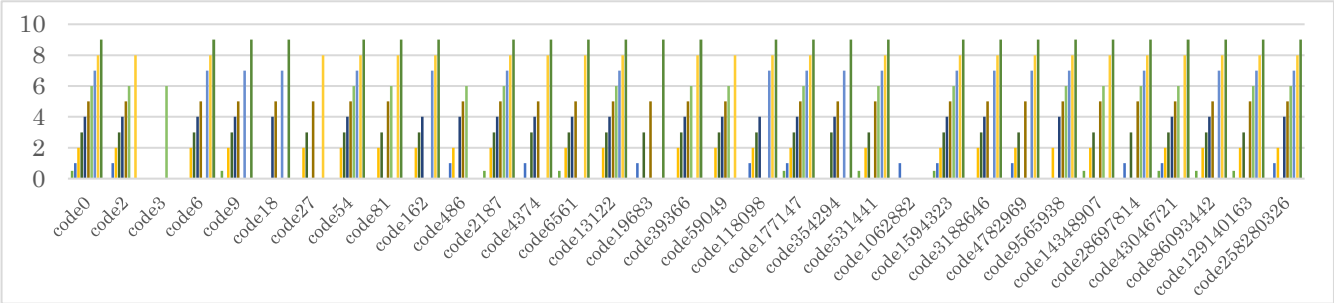

図 5　提案手法の内，最初に反応のある桁のみを用いる方法によって Mnist をコーディングした結果
縦軸はクラスラベル毎に振られた数字，横軸はコーディング結果を 10 進数に変換したもの

要素としたものと，木の各ノードでランダムな多次元の軸を射影軸として構築した決定木をフォレストの構成要素としたものの２つを比較対象として考える．

決定木をフォレストの構成要素とする場合にはフォレストの決定木同士の相関度合いを調整するために個々の決定木が元の学習データからデータをランダムに抽出し，木の各ノードでは注目する次元数をランダムに選択し木構築をする．そのため元データに対して各木が用いるデータと注目する次元数の割合を設定する必要があるが，本報告での実験では先行研究[3]を踏まえ，元データの 90%をランダムに抽出したデータを学習に用いり，データの次元数をDとすると$d = 1 + \log_2 D$または$\sqrt{D}$より求まるdを各ノードでランダムに求める射影軸の本数として設定し，木構築を行った．

データセットとしては識別問題において一般に用いられるものを UCI Machine Learning Repository[6]より引用する．データセットの詳細については表 1 にまとめる．また，ランダムフォレストでは，各木のランダムさにより構築される個々の木は一定ではなく，識別結果が試行ごとに異なることが考えられる．そこで本実験では，ランダムフォレストによる各識別性能の確認を繰り返し行い，識別率の平均値を計算したものを各フォレストの識別率として計算する．今回の試行回数は 40 回とした．

5. 実験結果

最初に，提案手法の決定木単体としての性能を他の手法と比較した結果を示す．前述のデータセットを対象とした

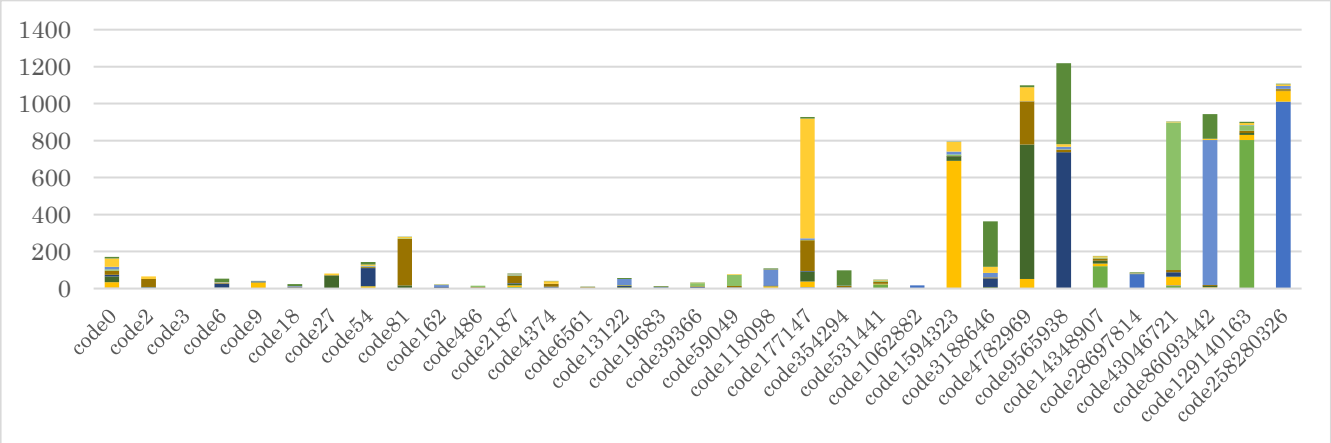

図 6　　提案手法の内，最初に反応のある桁のみを用いる方法によって Mnist をコーディングした結果(2)
縦軸はコードにマッピングされるベクトルの累積数，横軸はコーディング結果を 10 進数に変換したもの，クラスごとに色分している．

場合の各手法の識別率を表 2 にまとめる．表から，低次元なデータセットを対象としたときに各手法の識別率は総じて高く若干の差しか見て取れない．対して高次元なデータセットを対象としたとき，提案手法と判別分析軸を用いる木構築の識別率は高く，CART 法の識別率が低いことが見て取れる．特に高次元データを対象としたときに提案手法が最も高い識別率となっている．

このことから，提案手法は，高次元データを対象とした識別において特に有効であり，かつクラス情報を用いた単純な木構築よりも性能が高いと言える．

次に，提案手法をフォレストの構成要素とした場合の識別性能と他の手法とを比較した結果を示す．各データセットを対象とした場合の各手法の識別率を表 3 にまとめる．表から，低次元なデータセットを対象とするときに各手法の識別率にほぼ差は見て取れないが，高次元なデータセットを対象としたときの識別率においてランダムな軸に基づいた木構築をする手法が最も低く，提案手法が最も高かった．

このことから，提案手法はフォレストの構成要素とした場合の性能も十分であり，高次元データを対象とする場合にも高い性能を確認することができた．

フォレストの性質として一般に言われるように，各木の相関が低いほどフォレストとしての性能が高いことが挙げられる[3]が，提案手法をフォレストとした場合には，データのブーストラップによるランダムさはあるが，射影軸や閾値の決定にランダムさは無い．今後，フォレストの構成要素として提案手法を改良することを考えるならば，フォレストの各木の相関を互いに低くする必要がある．例えば提案手法では距離最大の２クラスに注目した射影軸の決定をするが，これにデータ数に応じた重み付けをすることでブーストラップされるデータ毎にクラスの偏りが変われば，異なった２クラスに注目することになり，各木の相関を低くできることが考えられる．

6. 提案手法の応用

本章では，提案手法の応用として多次元多クラス多数のデータに対してコーディングを行う運用を提案し，説明をする．

提案手法で構築された決定木にクエリが与えられ，木のノードを辿る場合を考える．図 1 より，木の中央には根から下に向かって束構造を持つ三分岐が連なった構造が構築されるが，クエリはそのいずれかでクラスの識別がされると三分岐の{A}または{B}を辿り，判断されない場合は木の{else}を辿り続ける．ここでその三分岐の識別結果に数値を割り当てることを考える．すると木にクエリを与えることで木の中心に連なる束構造の数と同じ桁数を持つ３値のコーディング結果を得る事ができる．

提案手法の応用のコーディング方法としては二種類が考えられる．一つは前述したとおりに木の中央に連なる束構造分のいずれかで{A}または{B}に識別されるまで順にクエリを与え反応を見る方法，もう一つは単体の木から中央の束構造を抜き出し横に並べ，クエリを並列に与えられるようにし，全ての木で得た反応を統合する方法である．こうすることでクエリを与えるときに並列計算が可能になるというメリットがある．また両方の方法において得られた 3 進数のコードを 10 進数に変換する場合は木の根に近い識別結果ほど重み付けを重くする．これは木の根に近い識別は全体における影響が大きいと考えるためである．

対象データとして Mnist[6]を用いて，提案手法によるコーディングを行った結果を次章で示す．

7. 応用手法の実験

提案手法の応用である二種類のコーディング手法の妥当性を確かめるために他の手法との比較を行う．比較の対象としてはランダムな射影軸にデータを射影し三等分割す

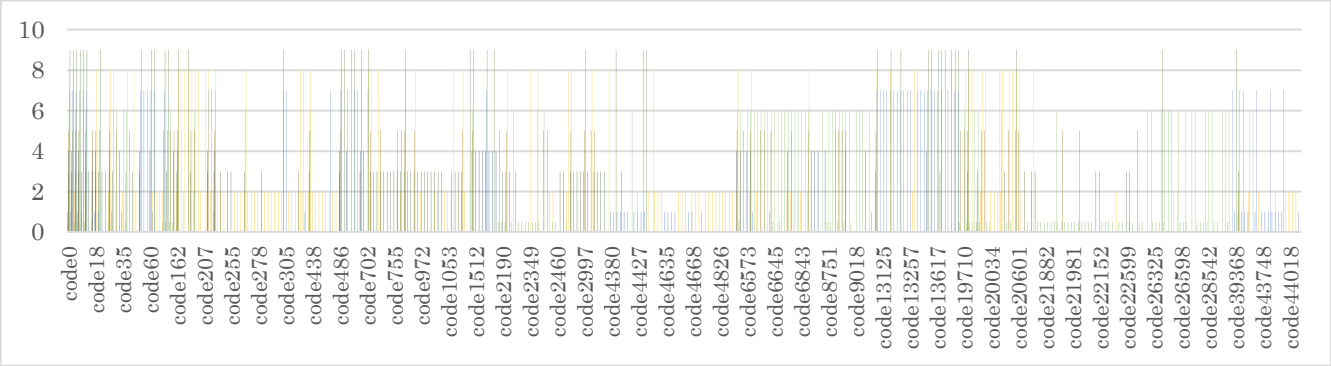

図 7 Deskew による前処理を行った Mnist を対象に提案手法の内，束構造全てを用いた方法によって Mnist をコーディングした結果
縦軸はクラスラベル毎に振られた数字，横軸はコーディング結果を 10 進数に変換したもの

る結果に三進数を割り当て，コーディング結果とする手法を考える．評価指標としてはデータのクラスラベル数に対する求められたコードの種類から計算する正規化相互情報量（NMI）を用いる．またコーディング結果を視覚的に俯瞰して確認するために，得られたコードと頻度をそれぞれ横軸と縦軸としグラフにプロットしたものを確認する．対象とするデータセットとしては数字の手書き文字の画像セットである Mnist を用いた．

提案手法の二種類のコーディング方法を提案手法(1), (2)とし，ランダム軸を用いたコーディングと比較をする．各手法で求まったコードの種類とデータのクラスラベル数から計算した NMI を表 4 にまとめる．

	提案手法(1)	提案手法(2)	ランダム軸
NMI	0.45	0.60	0.40

表 4 コーディング結果から得た，各手法の NMI
提案手法(1)は束構造全てを使った場合，提案手法(2)は最初に反応のある桁のみ使った場合

表 4 より，NMI の値はランダムな軸に基づいたコーディングを用いる場合に最も低く，二種類の提案手法によるコーディングの内でも特に根ノードから順に{A}または{B}に識別されるまでの桁のみでコーディングを行う手法が最も高いことが分かった．

次に，前述した各手法において計算されたコードを 10 進数化したものを横軸に，コーディングされ各データのクラスラベルを縦軸にプロットしたものをグラフにまとめる．これにより連続したコードに特定のクラスがマッピングされる場合に，その視覚的な判断が可能になると考えられる．

グラフより，まずランダムな軸に基づいたコーディング結果からプロットしたグラフ（図 3)では，横軸の数直線上に複数のクラスが散らばっており，連続したコードにマッピングされる特定のクラスの有無は判然としない．次に一つ目の提案手法である得られる束構造全てでコーディングを行う場合の結果からプロットしたグラフ（図 4）では，横軸に連続してマッピングされるクラスの固まりがいくつか見て取れる．最後に二つ目の提案手法である最初に識別される桁のみからコードを得る場合の結果からプロットしたグラフ（図 5）では，コードのほとんどに複数のクラスがマッピングされており，連続して出現するクラスの固まりは見て取れない．しかし，第 5 章での実験にあるように提案手法では Mnist に対して 9 割を超える識別率が確認されている．識別されるまで木を辿る処理は決定木本来の処理と等価と考えられ，この結果には矛盾が感じられる．そこで，最初に識別される桁のみからコードを得る場合の結果から，縦軸をコードにマッピングされるベクトルの累積数とし，クラスごとに色分けをしたものをプロットしたグラフ（図 6）を確認する．（図 6）より，ほとんどのコードに複数のクラスがアッピングされていることが確認できるが，個々のコードの累積数を特定のクラスの割合が多くを占めていることが見て取れる．計算された NMI の値と合わせて，この結果は妥当であると考えられる．つまり二つ目の提案手法では連続したコードへのマッピングという点では視覚的な確認が困難だが，個々のコードに対する特定のクラスがマッピングされることで，比較的良いコーディング結果が得られていると考えられる．

これらの結果から，提案手法の応用はクラス情報に敏感なコーディングを可能としており，コードをスカラ値に変換することで結果を視覚的に確認できることが分かった．提案手法の二種類のコーディング方法では，NMI に基づく評価では最初に識別される桁のみからコードを得る場合の結果が優位であるが，得られたコードのスカラ値に基づき結果をプロットする場合は，得られる束構造全てでコードを得る方が結果の視覚的な判断が容易だと考えらえる．単純にベクトルに対応するコードを得たい場合と，データ全体の傾向を視覚的に確認したい場合など，その時々で得たい結果ごとに用いる手法を選択することが考えられる．

特に得られる束構造全てでコードを得る場合では，多次

元多クラスのデータであっても 1 次元にプロットされた状態で全体を俯瞰した観察が可能であり，画像認識分野だけでなく，データマイニングなどの分野における研究であっても利用可能と考えられる．

また，画像処理において Deskew などの対象データに対する前処理を施す場合にその処理の効果の如何を視覚的に確認することができる．Deskew による前処理を施した Mnist を対象に提案手法によるコーディングを行った結果をプロットしたグラフ（図 7）を示す．

グラフより，前処理を施さなかった結果（図 4）と比べ，連続したコードにマッピングされるクラスの固まりがより凝集したように見て取れる．得られるコードの種類の数に関しても，前処理を施すことで元の 2,138 種類から 364 種類へ減少している結果が反映されていると考えられ，前処理を行うことによるデータへの影響を俯瞰して観察することができていると言える．

結果としては，ランダムな軸に基づいたコーディングよりも正規化相互情報量において提案手法によるコーディングでより良い数値が確認できた．また，コーディングを行った結果を図的に確認する方法を示し，主観的にだがコーディングによって，似た特徴が数直線上に固まって存在していることを確認した．

8. まとめ

データ内の最も分離しやすいクラスの組への注目と分割を繰り返す決定木の構築法を提案した．結果的に決定木の構造は束構造を含んだ三分木とほぼ同等になることを示し，得られる木の構造に注目することで多次元多クラスのデータをコーディングすることで 1 次元の数直線上へのマッピングを行う応用も示した．

結果としては，提案手法の決定木単体とフォレストの要素とした場合の性能としては，特に多次元多ベクトルのデータを対象とした時に優れていることを示した．手法の応用として２種類のコーディング方法を示したが，正規化相互情報量による評価では決定木本来の処理を踏襲する場合が有利であり，木の束構造全てでコーディングを行う場合に図的な解釈が容易であることを示した．今後は，決定木としては，よりフォレストの要素とした時の性能が向上するような木構築の工夫を実装することや，応用としてコーディングに用いる場合は，TF-IDF のような計算やコーディングにより得られた数字をファイル名として画像に対する自動名前付けを行うなどの発展も考えられる．

参考文献

[1] J. R. Quinlan, “Induction of decision trees,” Machine learning 1.1, pp. 81-106, 1986
[2] A. Criminisi, J. Shotton, E. Konukoglu, “Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning,” Foundations and Trends® in Computer Graphics and Vision: Vol. 7: No 2-3, pp 81-227, 2012
[3] L. Breiman, “Random forests,” Machine learning, Vol. 45, no. 1, pp. 5-32, 2001
[4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, “*Classification and regression trees*,” Monterey, CA: Wadsworth & Brooks, 1984
[5] J. R. Quinlan, “C4.5: Programs for Machine Learning,” Morgan Kaufmann Publishers, 1993.
[6] “UCI Machine Learning Repository“, (http://archive.ics.uci.edu/ml/)