Canonical Correlation Forests におけるラベル行列のスパース性を考慮した分類法に関する一考察

情報数理応用研究

5216C028-7　中野修平
指導教員　後藤正幸

Classification Method Considering Sparse Label Matrix Based on Canonical Correlation Forests

NAKANO Shuhei

1　研究背景・目的

カテゴリが既知の学習データから分類規則を学習することで，新たな入力データのカテゴリを予測する自動分類手法は，広い適用範囲を持つことから，盛んに研究が行われている．自動分類問題への有効なアプローチの一つにアンサンブル手法が知られている．アンサンブル手法とは，データから単一の分類器を学習させる一般的なアプローチとは異なり，複数の分類器を学習させ，その組み合わせによって分類精度向上を目指すアプローチである．機械学習の分野において，Random Forests (RF) [1] のような決定木をアンサンブルする手法は自動分類において高い分類性能を持つことから注目を得ている．また，RF をさらに発展させた手法として Canonical Correlation Forests (CCFs) [2] がある．CCFs は座標軸に対して平行な境界面を作る従来の決定木とは異なり，複雑な構造に対して座標軸にとらわれない境界面を作成することを目指した Oblique Decision Tree (ODT) [3] に基づいた手法である．一般的な RF とは異なり，CCFs は各葉ノードにおいて正準相関分析 [4] を行い，得られた正準変数上で閾値を決定し，識別境界を構築することで，元の特徴空間では座標軸に対して平行でない識別境界を得ることができる．加えて，それらの個々の木をアンサンブルすることにより，柔軟な境界面を構築することが可能である．

CCFs における個々の決定木 (以下，CC-Tree) は各ノードごとで説明変数行列と 1-of-K 表現されたカテゴリ行列から計算された正準変数を用いることで，カテゴリ情報を考慮した超平面を構築することができる．しかしながら，各ノードで閾値を決定し，子ノードへデータを分割する木構造のアルゴリズムの特性上，木が深くなるほどカテゴリ行列はスパースになる傾向があり，スパースなカテゴリ行列上で決定した識別境界は学習データに対して過学習しやすいという問題がある．一方で，カテゴリ行列を正準相関分析の枠組みを考慮した形でその都度，適切に構成し直すことができれば，この問題点を解決できる可能性がある．そこで本研究では，CCA の枠組みを考慮したカテゴリ行列の最適化するアルゴリズムの提案を行い，分類精度向上を図る．また，評価実験により提案手法の有効性を示す．

2　準備

2.1　Oblique Decision Tree とそのアンサンブル法

決定木は枝と葉からなる木構造をした分類モデルである．一般的な決定木のアルゴリズムの共通の考えは，1つの変数に着目して分割点を探索し，ジニ係数やエントロピーなどの基準を用いて最も有効な分割を順次決めていくことである．決定木の代表的なアルゴリズムとして CART [5] や C4.5 [6] が知られている．どちらのアルゴリズムも分割点を求めた後，学習データを子ノードへ分岐させる．この処理はノードをさらに分割しても情報利得が得られないか，あるいはノードが同じカテゴリデータのみの集合になるか，終了条件を満たすまで再帰的に繰り返される．

一方，RF [1] のようにランダム性を考慮して生成した複数の木を組み合わせ，アンサンブルを行うことで分類精度が向上することが知られている．アンサンブル学習において分類精度を向上させるためには，各決定木における識別境界の多様性が重要である．ここで，個々の決定木の多様性を高める手段の一つとして Oblique Decision Tree がある．Oblique Decision Tree は，各変数の重み付け和から得られる合成変数を用いることで，座標軸にとらわれない境界面を構築する手法である．個々の決定木に座標軸にとらわれない境界面の構築が可能となることで，木にモデルとしての自由度が生じる．このようにして，木々に多様性を与えることで，Oblique Decision Tree はアンサンブルの効果を高めている．Rotation Forest [7] は Principal Component Analysis (PCA) を説明変数に適用し作成した合成変数を用いることで多様性を得て，高い分類精度を達成する Oblique Decision Tree の考えに基づいた代表的な手法である．Rotation Forest が説明変数のみに着目して合成変数を得ている一方で，CCFs [2] は Canonical Correlation Analysis (CCA) を用いることで説明変数とカテゴリ行列を考慮しており分類に適した合成変数 (正準変数) を用いて分割を生成していく手法である．

2.2　ノーテーション

自動分類問題は，D 次元特徴空間上の特徴ベクトル $\boldsymbol{x} \in \mathbb{R}^D$ から K 個の離散カテゴリ集合 $\mathcal{C} = \{c_1, \dots c_K\}$ への写像を得ることである．そのために，カテゴリが既知である N 個の学習データ $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ を用いて，この写像 (分類器) を学習することを考える．ここで，$\boldsymbol{x}_n \in \mathbb{R}^D$ は n 番目の学習データの D 次元の特徴ベクトル，$y_n \in \mathcal{C}$ は n 番目の学習データの $\boldsymbol{x}_n$ の所属カテゴリである．この学習によって得られた分類器を用いて，カテゴリが未知の新規入力データ $\tilde{\boldsymbol{x}} \in \mathbb{R}^D$ の所属カテゴリを推定することができる．ここで，$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2 \dots \boldsymbol{x}_N)^{\mathrm{T}}$ をカテゴリが既知である特徴量行列，$\boldsymbol{y} = (y_1, y_2 \dots y_N)^{\mathrm{T}}$ をカテゴリベクトル，$\mathcal{T} = \{t_l\}_{l=1}^L$ を L 個の木からなる木集合と定義しておく．また，$t_l = (\boldsymbol{\Psi}_l, \boldsymbol{\Theta}_l)$ は中間ノード集合 $\boldsymbol{\Psi}_l = \left\{\psi_{lj}\right\}_{j \in \mathcal{J} \setminus \partial\mathcal{J}}$，葉ノード集合 $\boldsymbol{\Theta}_l = \left\{\theta_{lj}\right\}_{j \in \partial\mathcal{J}}$ からなるなる単独木を表すものとする．ただし，$\mathcal{J}$ はノード番号集合，$\partial\mathcal{J} \subseteq \mathcal{J}$ は葉ノード集合，$\setminus$ は差集合を表す演算子であるとする．また，各中間ノードは $\psi_{lj} = \left\{\chi_{lj_1}, \chi_{lj_2}, \boldsymbol{\phi}_{lj}, s_{lj}\right\}$ によって定義される．ただし $\left\{\chi_{lj_1}, \chi_{lj_2}\right\} \subseteq \mathcal{J} \setminus j$ は l 番目の木のノード j からの二つの子ノードであり，$\boldsymbol{\phi}_{lj}$ は l 番目の木のノード j における特徴ベクトルへの重みベクトル，s_{lj} は l 番目の木のノード j における写像空間 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\phi}_{lj}$ における分割点である．

また，$B(j, t_l)$ は l 番目の木におけるノード j が表す特徴空間上の集合を示す．例えば $B(j, t_l)$ は l 番目の木におけるノード j が表わす特徴空間上の部分集合を意味する．この定義により，$B(j = 0, t_l)$ は木 t_l に関わらず全ての木において全特徴量空間を表す．また，$B(j, t_l) = B(\chi_{j_1}, t_l) \cup B(\chi_{lj_2}, t_l)$ の関係が成り立つ．つまり，$B(j, t_l)$ の二つの子ノード $B(\chi_{j_1}, t_l)$，$B(\chi_{j_2}, t_l)$ は以下の式 (1), (2) より表現できる．ここで，$\boldsymbol{z} \in \mathbb{R}^D$ はノードに所属するデータにおける任意の特徴ベクトルであるとする．

$$B(\chi_{j_1}, t_l) = B(j, t_l) \cap \left\{ \boldsymbol{z}^{\mathrm{T}} \boldsymbol{\phi}_{lj} \leq s_{lj} \right\} \tag{1}$$

$$B(\chi_{j_2}, t_l) = B(j, t_l) \cap \left\{ \boldsymbol{z}^{\mathrm{T}} \boldsymbol{\phi}_{lj} > s_{lj} \right\} \tag{2}$$

3 Canonical Correlation Forests

3.1 概要

RF や多くの決定木のアンサンブル法と同じく，CCFs の木は独立して学習される．学習アルゴリズムはすべてのデータが所属する根ノードから始まる．そして，各ノード $B(j, t_l)$ ごとで正準相関分析 (式 (3) – (5)) を行い，式 (6), (7) で定義されるように，正準変数上で分割点 s_{lj} を決定し，その閾値 s_{lj} に対する大小で子ノードへのデータを所属させる．この処理をすべてのノードが終了条件を満たすか，ノードに所属するデータが一つのカテゴリになるまで再帰的に繰り返す．ただし，$\boldsymbol{X}_j \in \mathbb{R}^{N_j \times D}$ はノード j の特徴行列，$\boldsymbol{Y}_j \in \mathbb{R}^{N_j \times K_j}$ はノード j のカテゴリベクトルに対して所属するカテゴリに対して 1 の値をとり，それ以外は 0 の値をとる 1-of-K 表現を適用したカテゴリ行列，N_j はノード j に属するデータ数，K_j はノード j に属するデータのカテゴリ数，$\boldsymbol{a}_j \in \mathbb{R}^D$ はノード j で正準相関分析によって得られる説明変数に対する重みベクトル，$\boldsymbol{b}_j \in \mathbb{R}^{K_j}$ はノード j で正準相関分析によって得られるカテゴリベクトルに対する重みベクトルとする．

RF のような一般的な決定木アルゴリズムとの違いは，CCFs は各ノード j において特徴ベクトルとカテゴリを考慮した写像空間 $\boldsymbol{X}_j \boldsymbol{a}_j$ で分割点 s_{lj} を探索し識別境界を決定することである．これにより，元の特徴量空間では座標軸にとらわれない識別境界を得ることができる．

$$(\boldsymbol{a}_j, \boldsymbol{b}_j) = \underset{\boldsymbol{a}, \boldsymbol{b}}{\mathrm{argmax}} \ corr(\boldsymbol{X}_j \boldsymbol{a}, \boldsymbol{Y}_j \boldsymbol{b}) \tag{3}$$

$$\text{subject to} \quad \left|\left|\boldsymbol{a}_j\right|\right|_2 = 1 \tag{4}$$

$$\left|\left|\boldsymbol{b}_j\right|\right|_2 = 1 \tag{5}$$

$$B(\chi_{j_1}, t_l) = B(j, t_l) \cap \left\{ \boldsymbol{z}^{\mathrm{T}} \boldsymbol{a}_j \leq s_{lj} \right\} \tag{6}$$

$$B(\chi_{j_2}, t_l) = B(j, t_l) \cap \left\{ \boldsymbol{z}^{\mathrm{T}} \boldsymbol{a}_j > s_{lj} \right\} \tag{7}$$

3.2 CCFs のアルゴリズム

CCFs の学習アルゴリズムでは一般的な RF と同じく，データセット $(\boldsymbol{X}, \boldsymbol{Y})$ からブートストラップサンプリングされたデータセットを対象に個々の木が並列に学習を行う．CCFs のアルゴリズムでは各ノードごとに CCA を行い，重みベクトルを導出する．次に，合成変数上で，最適な分割点を探索する．その決定した分割点に基づく識別境界は正準変数上では座標軸に対して平行ではあるが，元の特徴量空間では座標軸に対して平行ではなくなる．これらの正準変数上で識別境界を決定し，さらにそれらの識別境界をアンサンブルすることで CCFs は全体の識別境界の決定を行う．そのため，CCFs は柔軟な識別境界を構築することが可能である．CCFs の学習アルゴリズムを以下に示す．

Step1) $l = 0$ とする．

Step2) $j = 0$ とする．

Step3) $(\boldsymbol{X}, \boldsymbol{Y})$ からブートストラップサンプリングを行いデータセット $B(\chi_j, t_l)$ を作成する．

Step4) $B(\chi_j, t_l)$ の D 個の変数から d $(< D)$ 個の変数をランダムに選択する．

Step5) CCA を行い $\boldsymbol{a}_j$ を求める．

Step6) ジニ係数を基準に最適な分割点 s_{lj} を探索する．

Step7) 式 (6), (7) を用いて二つの子ノードを計算する．

Step8) 式 (8), (9) に従い，$B(\chi_{j+1}, t_i), B(\chi_{j+2}, t_i)$ を更新する．

$$B(\chi_{j+1}, t_l) \leftarrow B(\chi_{j_1}, t_l) \tag{8}$$

$$B(\chi_{j+2}, t_l) \leftarrow B(\chi_{j_2}, t_l) \tag{9}$$

Step9) $j = j+1$ とし，終了条件を満たさなければ Step4) に戻る．

Step10) $l = l+1$ とし，$l \leq L$ であれば Step2) へ戻る．

4 提案手法

4.1 概要

CCFs における木単独モデルである CC-Tree は CCA を用いることにより，正準変数上で分割点を探索するモデルである．各ノードで正準変数を求め，それに対する識別境界を決定し，アンサンブルを行うことで，CCFs は柔軟な境界面を構築することができ，高い分類精度が期待できる．しかしながら，カテゴリ行列がスパースな場合，CCFs はノードごとに分類に適切な識別境界を得ることができない．これは，決定木のような木構造を用いた学習アルゴリズムは木が深くなるほどノードに所属するデータ数が減少し，カテゴリ行列がスパースになる傾向があるためである．カテゴリ行列がスパースな状態で正準相関分析を行った場合，推定するパラメータ数に対してデータ数が不足となり推定精度が低くなる．そのような状態で得られた正準変数上で識別境界を決定しても学習データに対して過学習しやすいことが挙げられる．また，カテゴリ数が多くなると写像された空間が複雑となり，ある正準変数における閾値のみで分類を可能とするような問題ではなくなる可能性がある．

CCFs が多カテゴリ問題に対して脆弱性を持つ一つの理由として，CCA は写像する際に $\boldsymbol{Y}_j \boldsymbol{b}_j$ の計算が用いられていることが挙げられる．$\boldsymbol{Y}_j$ はカテゴリ行列であるため，各行に対してほとんどが 0 の値を持つスパースな行列になっている．そのような行列に対して，$\boldsymbol{b}_j$ の次元数が大きい場合に式 (3) – (5) の計算を行うと $\boldsymbol{b}_j$ の推定が過学習してしまう可能性がある．そこで，本研究では，適切な分割点を得るため CCA を行なった後，カテゴリ行列 $\boldsymbol{Y}_j$ に対しても最適化を行うことを考える (式 (10) – (12))．$\boldsymbol{Y}_j$ は各行において，所属カテゴリの 1 箇所のみ 1，それ以外で 0 のスパースな行列であったが，これを再構成することで，複数の箇所で 1 を取る密な行列が得られる．これは，正準変数 $\boldsymbol{X}_j \boldsymbol{a}_j$ 上で，類似度が高いデータに対して同じカテゴリとして扱うことと等しい．つまり，得られた正準変数では分類困難なデータを一時的に同カテゴリとみなすことで明確な識別境界が得られる可能性が高まると考えられる．また，一時的に同カテゴリと判断されたデータ群は子ノードで新たな正準変数を得ることで分類することが可能である．

しかし，直接カテゴリ行列 $\boldsymbol{Y}_j$ の最適化を行うと $\boldsymbol{Y}_j$ の全ての要素に対して $\{0, 1\}$ を与えるような組み合わせ問題となり $O(2^{N_j - 1} - 1)$ の計算量が必要となる．計算量がデータ数 N_j に依存する形となり，現実的に実行困難となってしまう．そこで本研究では，一つ一つのデータに着目するのではなく，カテゴリに着目して $\boldsymbol{Y}_j$ の最適化を試みる．以上の議論より，カテゴリ数の多い分類問題を対象に，新しい CCFs のアルゴリズムを提案する．このため，Exhaustive 符号 [8] の考えを CCFs へ導入する．Exhaustive 符号とは 2 値分類器を組み合わせて多値分類問題へと拡張さた ECOC 法の 1 つであり，カテゴリの組み合わせ示す符号表で与えられ，2 値分類器に対してすべてのカテゴリの組み合わせを示す Exhaustive 符号は，カテゴリごとの組み合わせの探索領域と考えられる．ここで，Exhaustive 符号に基づくカテゴリの組み合わせに従い，式 (10) の値が最も高くなるような $\boldsymbol{Y}_j'$ を導出する．そうすることで，CCA は分割に適した写像を行うことができる．

$$\boldsymbol{Y}_j' = \underset{\boldsymbol{Y}_j}{\mathrm{argmax}} \quad corr(\boldsymbol{X}_j \boldsymbol{a}_j, \boldsymbol{Y}_j \boldsymbol{b}_j) \tag{10}$$

$$\text{subject to} \quad y_{im} \in \{0, 1\} \tag{11}$$

$$||\boldsymbol{y}_i||_2 = 1 \tag{12}$$

4.2 Exhaustive 符号

Exhaustive 符号は 2 値分類器を多値分類問題へと拡張させるために考案された ECOC 法で用いられる符号表である．表 1 のように各要素は {0,1} で構成されており，各要素分類器は与えられた 1 と 0 のカテゴリを判別する 2 値分類問題として学習を行う．また，Exhaustive 符号はカテゴリ数 K に対して $2^{K-1}-1$ 個の考えられる全ての 2 値分類に対する判別器構成となっている．新たなデータを分類する場合，各 2 値分類器の出力結果と符号表とのハミング距離が最も近いカテゴリに分類する手法である．

ここで式 (10) – (12) を直接用いた場合，計算量は $O(2^{N_j-1}-1)$ となる．そこで，本研究では式 (10)–(12) を Exhaustive 符号を用いることでその計算量を $O(2^{K_j-1}-1)$ まで削減をする．各ノード間で最適な Exhaustive 符号のカテゴリの組み合わせを探索的に行うことにより，相関が最も高くなるようなカテゴリ行列 $\boldsymbol{Y}_j'$ を求めることで $\boldsymbol{Y}_j$ の最適化を行う．

表 1. Exhaustive 符号 (K=4)

c_1	1	1	1	1	1	1	1
c_2	0	0	0	0	1	1	1
c_3	0	0	1	1	0	0	1
c_4	0	1	0	1	0	1	0

4.3 提案アルゴリズム

従来の CCFs のアルゴリズムは各ノードで CCA で得た正準変数上で閾値を決定する一方，提案アルゴリズムはノードごとに CCA を行い，カテゴリ行列への重みベクトル $\boldsymbol{b}_j$ を導出後，$\boldsymbol{a}_j, \boldsymbol{b}_j$ が与えられたものとして最適なカテゴリ行列 $\boldsymbol{Y}_j$ の最適化を行う．その後，正準変数上で閾値を決定する．各ノードで CCA を適用した後，Exhaustive 符号に基づくカテゴリの組み合わせに従い，カテゴリ行列 $\boldsymbol{Y}_j$ を変換し，式 (10) で定義される相関が最も高くなったカテゴリの組み合わせを式 (10) – (12) の解とする．その後，CCA によって得られた正準変数と最適化されたカテゴリ行列 $\boldsymbol{Y}_j'$ を用いて，ジニ係数を基準に最適な分割点を探索する．分割点の決定後，式 (6), (7) に従いデータを子ノードへの所属させる．ただし，一度同じカテゴリとして統合されたカテゴリは子ノードでは再び別カテゴリとされ，再度 Exhaustive 符号に基づき最適なカテゴリごとの組み合わせを求める．これらの処理を終了条件を満たすか，ノードが同じカテゴリデータの集合になるか，終了条件を満たすまで各子ノードにおいて再帰的に繰り返される．

Step1) $l=0$ とする．

Step2) $j=0$ とする．

Step3) $(\boldsymbol{X},\boldsymbol{Y})$ からブートストラップサンプリングを行いデータセット $B(\chi_j,t_l)$ を作成する．

Step4) $B(\chi_j,t_l)$ の D 個の変数から d $(<D)$ 個の変数をランダムに選択する．

Step5) CCA を行い $\boldsymbol{a}_j$ を求める．

Step6) Exhaustive 符号に基づき最適な $\boldsymbol{Y}_j'$ を求める．

Step7) ジニ係数を基準に最適な分割点 s_{lj} を探索する．

Step8) 式 (6), (7) を用いて二つの子ノードを計算する．

Step9) 式 (8), (9) に従い，$B(\chi_{j+1},t_i), B(\chi_{j+2},t_i)$ を更新する．

Step10) $j=j+1$ とし，終了条件を満たさなければ Step4) に戻る．

Step11) $l=l+1$ とし，$l\leq L$ であれば Step2) へ戻る．

5 評価実験

5.1 実験条件

提案手法の有効性を示すため，表 2 で表される UCI データセット (balance scale, nursery, optDigitsHandwritten, libras) [9] を用いて分類実験を行なった．比較手法として RF，Rotation Forest [7]，CCFs を用いる．評価指標にはテストデータに対する分類誤り率を用いた．

また，木ごとの最大深さは 100，ノードの最小データ数は 20 とし，木の数は 50 から 200 まで 50 ずつ増加させるものとする．加えて，学習データとテストデータの比を 4:1 とする．また，今回はカテゴリ行列がスパースな場合の汎化性能の検証を行うため，学習に用いるデータ数を以下のように制限する．学習データの f%のみをパラメータ推定に使用するものとし，$f=20,40$ の 2 通りを実験した．以上の条件で，5-fold-cross-validation を 10 回繰り返し，その平均の結果が最良であった設定を実験結果として示す.

表 2. UCI データセットの概要

データセット名	次元数	カテゴリ数	データ数
balance scale	4	3	625
nursery	8	5	12960
optDigitsHandwritten	64	10	5620
breast tissue	9	6	106
libras	90	15	360

5.2 実験結果

UCI のデータセットに対する各手法の分類性能を表 3 に示す．その際，提案手法の有効性を示すため，CCFs と提案手法で統計的有意差があるか否かについて，t 検定により確認を行った．表 3 における $*$ は 5%有意，$**$ は 1%有意を示している．

表 3 より，balance scale (40%), nursery (20%), (40%), optDigitsHandwritten (20%), (40%) libras(20%),(40%) で提案手法が高い分類性能を得ていることがわかる．これは，提案手法は各ノードでカテゴリを統合し学習するため，各ノードごとで得られる分割境界が大きく異なり，結果としてアンサンブルの効果が高まったためと考えられる．

図 1 – 6 は各学習データに対してアンサンブル数 L を増加させた時の分類性能の変化を示している．Rotation Forest, CCFs, 提案手法はアンサンブル数を増加させることで分類性能が高くなっていくことがわかる．これは，これらの手法が各ノードごとで合成変数を得て分類することで，アンサンブルの効果が高まり，Random Forests に比べて高い分類性能を得られたためと考えられる．また，同じデータセットに対して同じアンサンブル数の場合は，学習に使用できるデータ数が多いほど分類精度が高い結果が得られた．加えて，同じデータセットに対して，学習に使用できるデータ数が異なる場合，データ数が多いほどアンサンブルの効果が高いこともわかる．

図 7 にアンサンブル数を L=100 にし，使用する学習データ数を増加させた時の分類性能の変化を示す．使用できる学習データ数が増加するにつれて，すべての手法の分類性能は向上した．その中でも，提案手法はすべての条件に対して最も高い分類精度を持つことが確認できる．これは，学習に使用できるデータ数が異なる場合でも，提案手法は高い分類性能を維持することを示している．

表 3. UCI データセットの実験結果 (アンサンブル)

データセット名	Random Forests	Rotation Forest	CCFs	提案 (Forest)
balance scale (20%)	0.179	0.144	**0.113**	0.117
balance scale (40%)	0.157	0.124	0.098	**0.088**$*$
nursery (20%)	0.068	0.099	0.060	**0.040**$*$
nursery (40%)	0.044	0.0664	0.020	**0.018**$*$
optDigits Handwritten (20%)	0.048	0.048	0.036	**0.030**$*$
optDigits Handwritten (40%)	0.033	0.030	0.022	**0.020**
breast tissue(20%)	**0.536**	0.550	0.554	**0.536**$*$
breast tissue(40%)	0.520	0.595	0.541	**0.518** $*$
libras(20%)	0.490	0.572	0.450	**0.316**$**$
libras(40%)	0.501	0.586	0.410	**0.312**$**$

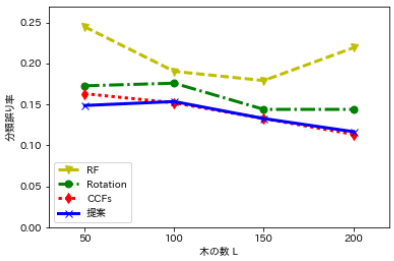

図 1: balance scale の分類性能と木の数 (20%)

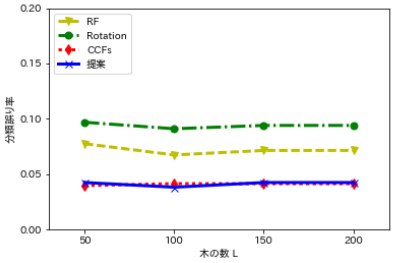

図 2: nursery の分類性能と木の数 (20%)

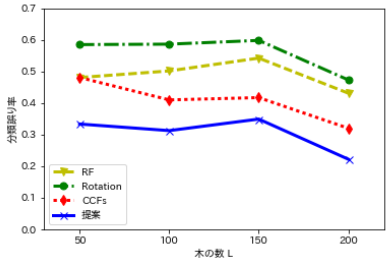

図 3: libras の分類性能と木の数 (20%)

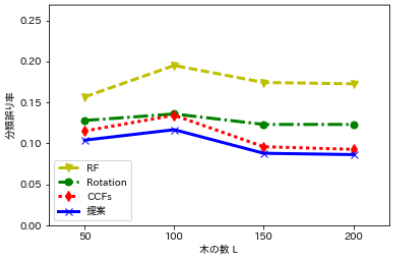

図 4: balance scale の分類性能と木の数 (40%)

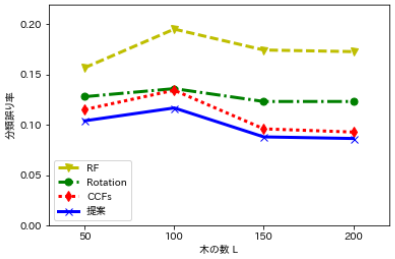

図 5: nursery の分類性能と木の数 (40%)

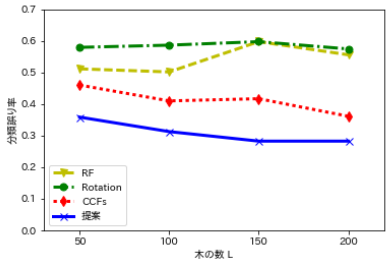

図 6: libras の分類性能と木の数 (40%)

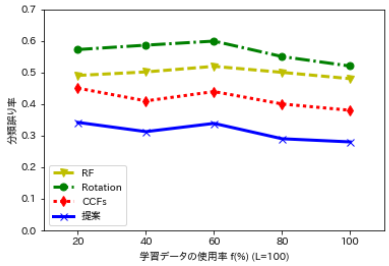

図 7: libras の分類性能と学習データの使用率 (L = 100)

6 考察

提案手法では様々なデータにおいて高い分類性能が得られている．つまり，木構造のアルゴリズムにおいてノードごとで一時的に同カテゴリとしてみなすことで特徴空間を分割する方法は分類問題に対して有効な手段であると考えられる．これは，一度に分類することが難しいデータ (異なるカテゴリではあるが類似性が高いデータ) を一時的に同カテゴリをみなすことで，段階的に識別し，分類しにくいデータも分類することが可能になったためである．それに加えて，提案手法はノードごとに異なる分類問題を行なっていることと等しいので，個々の決定木に多様性が高まり，アンサンブル効果が向上したことも分類精度が向上した理由と考えられる．また，提案手法はカテゴリの組み合わせに対して Exhaustive 符号を用いて全探索を行なっている．その結果、従来の CCFs のアルゴリズムに比べ学習時間が必要である．このことから，実応用では 学習時間と分類精度のトレードオフの関係に対しても考慮しなくてはならない．

7 まとめと今後の課題

本研究では，カテゴリ行列がスパースである多値分類問題を対象とし，各ノードごとで ECOC の考えに基づいたカテゴリ行列の最適化学習アルゴリズムの提案を行った．実験結果より，学習に扱えるデータ数が少ない場合，提案手法のアンサンブル法を用いることにより高い分類性能が維持できることを示した．今後の課題としては，CCA を行う試行回数の削減とカテゴリ行列を直接最適化する手法の提案などがあげられる．

参考文献

[1] L. Breiman, “Random Forests,” *Machine learning*, Vol.45, No.1, pp.5–32, 2001.

[2] T. Rainforth, F. Wood, “Canonical Correlation Forests,” unpublished paper. [Online]. Available: `https://arxiv.org/pdf/1507.05444.pdf`, 2015.

[3] J. Gama, “Functional Trees,” *Machine Learning*, Vol.55, No.3, pp.219–250, 2004.

[4] H. Hotelling, “Relations between Two Sets of Variates,” *Biometrika*, Vol.28, pp.321–37, 1936.

[5] L. Breiman, J. Friedman, C.J. Stone and R.A. Olshen, “Classification and Regression Trees,” *CRC press*, 1984.

[6] J.R. Quinlan, “C4.5: Programs for Machine Learning,” *Elsevier*, 2014.

[7] J.J. Rodriguez, L.I. Kuncheva, and J.A. Alonso, “Rotation Forest: A New Classifier Ensemble Method,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*,” Vol.28, No.10, pp.1691–1630, 2006.

[8] T.G. Dietterich, G. Bakiri, ” Solving Multiclass Learning Problems via Error-Correcting Output Codes,” *Journal of Artificial Intelligence Research*, Vol.2, 263-286, 1995.

[9] Dua,D.,Taniskidou.E., UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml`. Irvine, CA: University of California, School of Information and Computer Science., 2017.