2K2-4

離散化による解釈可能な分類モデルの構築

Learning Interpretable Classification Rules Using Discretization

米田 友花[*1] 杉山 麿人[*2] 鷲尾 隆[*1]
Yuka Yoneda Mahito Sugiyama Takashi Washio

[*1]大阪大学 産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University

[*2]独立行政法人科学技術振興機構, さきがけ
JST PRESTO

We propose to learn classification rules from supervised continuous data via a two-step procedure: We first binarize data points and then perform feature selection on the binarized data to obtain minimum necessary binary features that are required to classify given data. Each binary feature represents an interval on the original feature space, hence the user can directly interpret classification rules as intersections of intervals. Although such *interpretability* is indispensable in machine learning applications from biology to economics, we cannot often interpret classification rules obtained by recently emerging highly accurate machine learning methods such as deep learning. We experimentally demonstrate that our method can learn simpler classification rules than decision tree classifiers while keeping reasonable accuracy.

1. はじめに

近年，連続値データからのクラス分類において，深層学習と呼ばれるニューラルネットワークに基づく機械学習手法が着目されている [4]．深層学習では，膨大な計算量とデータを必要とする一方，高い分類精度を達成することが可能となってきているが，学習モデルが複雑すぎるため，人間が分類規則を解釈することができない，という欠点がある．これに対して，CART, ID3, C4.5 [3] などの**決定木**（decision tree）を用いた分類手法に代表される，特徴空間の矩形分割に基づく教師あり学習手法が知られている．これらの手法では，クラスの分類規則を，人間が理解することのできる表現形式として明示的に獲得することができるが，一般的に分類精度が低く，特に多変数の連続値データに対する簡潔な規則の獲得が難しい．そこで，多変数の連続値データに対して高い精度を達成することができ，かつ解釈可能性の高い簡潔な分類規則を獲得する機械学習手法が求められている．

この問題を解決するため，本研究では，連続値データから解釈可能かつ簡潔なクラス分類規則を学習する手法を提案する．提案手法ではまず，データの各特徴を十分に**離散化**（discretization）することで，クラスラベルに対して無矛盾な 2 値特徴の集合を生成する．そして，2 値データに対する教師あり特徴選択手法 Super-CWC アルゴリズム [6] を適用して不要な特徴を削除することで，クラスラベルを判別するために必要最低限の 2 値特徴を獲得する．各 2 値特徴の組合せが多次元空間上の矩形領域に対応しており，これらの分類規則を用いることで未知データを分類することができる．

本稿は，以下の様な構成となっている．まず 2.節で提案手法を説明する．最初に概要を説明したあと，2.1節では離散化，2.2節では提案手法の一部で使用した Super-CWC アルゴリズムについて述べる．3.節で提案手法の動作例を示したあと，4.節で提案手法を実験によって評価し，結果を考察する．最後に 5.節で本研究の結論と今後の課題について述べる．

連絡先: 米田友花，大阪大学産業科学研究所，
〒 567-0047 大阪府茨木市美穂ヶ丘 8-1
yoneda@ar.sanken.osaka-u.ac.jp

2. 提案手法

提案手法は，主に 2 つの段階からなる．第一段階では，連続値データの各特徴をそれぞれ区間を表現する複数の 2 値特徴へと変換することで離散化し，2 値データへ変換する．第二段階では，第一段階で生成した 2 値データを入力とし，Super-CWC アルゴリズムを適用することで，不要な特徴を削除しクラスラベルに無矛盾な最小の 2 値特徴集合を獲得する．

提案手法への入力データは，ラベル付きの連続値ベクトルの集合である．各データ点は，n 次元のベクトル $\boldsymbol{x}_i = (x_i^1, x_i^2, \ldots, x_i^n)$ とクラスのラベル $y_i \in \{0,1\}$ の組 $(\boldsymbol{x}_i, y_i)$ $(i = 1, \ldots, N)$ である．なお提案手法では，各特徴に対し，**min-max 正規化**（min-max normalization）を用いてあらかじめ区間 [0,1] に正規化する．すなわち，特徴ベクトルの各値 x_i^j を $(x_i^j - \min_{1\le l\le N} x_l^j)/(\max_{1\le l\le N} x_l^j - \min_{1\le l\le N} x_l^j)$ へ変換するため，$\boldsymbol{x}_i \in [0,1]^n$ となる．

2.1 離散化

離散化では，連続値の特徴を要素にもつ各特徴ベクトルを離散化して分割することで，2 値特徴ベクトルへ変換する．変換された特徴はもとの空間内の区間に対応しており，それぞれの 2 値特徴ベクトルの要素は，その区間への所属を表す．この変換を各区間の幅を狭めていきながら繰り返し，クラスラベルが判別できたところで終了する．ただし，分割回数の限度をパラメータとして与えることで，過度の分割を防ぐ．

提案手法における**分割**（partition）とは，区間 [0,1] をいくつかの区間に等分することである．すなわち，区間 [0,1] を k 区間に分割するとき，各区間は $[0, 1/k), [1/k, 2/k), \ldots, [(k-1)/k, k/k = 1)$ となる．一般に，関数 g を用いて，各区間を $g(w_k) = [w_k/k, (w_k+1)/k)$ と定める．ただし，$w_k \in \{0, 1, \ldots, k-1\}$ である．

これらの定義を用いて，特徴 $j \in \{1, 2, \ldots, n\}$ を k 区間に分割した中で，$g(w_k)$ で表される区間に属するか否かによ

Algorithm 1 離散化アルゴリズム

```
S ← ∅
for all k = 2 to α_max do
    generated by Eq.(1)
    S ← S ∪ Z^k
    if Bn(Z^k) = 0 or k = α_max then
        break
    end if
end for
S を出力する
```

って 2 値化した特徴ベクトルを $\boldsymbol{z}^{(j,w_k)}$ と表し，その特徴を (j, w_k) と書く．具体的には，特徴ベクトル $\boldsymbol{z}^{(j,w_k)}$ の各要素 $z_i^{(j,w_k)}(i = 1, \ldots, N)$ は以下で与えられる．

$$z_i^{(j,w_k)} = \begin{cases} 1 & \text{if } x_i^j \in g(w_k), \\ 0 & \text{otherwize.} \end{cases}$$

そして，すべての特徴を k 分割したときに得られる 2 値特徴集合を Z^k とおく．すなわち，

$$Z^k = \left\{ (j, w_k) \mid j \in \{1, 2, \ldots, n\}, w_k \in \{0, 1, \ldots, k-1\} \right\} \tag{1}$$

となる．このとき，各データ $\boldsymbol{x}_i = (x_i^1, x_i^2, \ldots, x_i^n)$ は $\boldsymbol{z}_i = (z_i^{(1,0)}, \ldots, z_i^{(1,k-1)}, z_i^{(2,0)}, \ldots, z_i^{(2,k-1)}, \ldots, z_i^{(n,0)}, \ldots, z_i^{(n,k-1)})$ という 2 値ベクトルに変換される．$k = 2, \ldots, \alpha$ まで離散化したときの全特徴からなる集合を S とすると，$S = Z^2 \cup Z^3 \cup \cdots \cup Z^\alpha$ となる．ここで，S を出力するのは，第二段階の特徴選択後に獲得する区間が，必要以上に細かくなることを防ぐためである．

提案手法は，離散化で用いる分割数 k を 2 からひとつずつ増やしていく．分割数を増やすたびに，その時点で 2 値化されたデータが，与えられたクラスラベルに対して無矛盾かどうかを判定する．ここで，与えられた特徴集合がクラスラベルに対し無矛盾であるかどうかを，関数 Bn を用いて次のように判定する．ある特徴集合 I に対し $\mathrm{Bn}(I) = 1$ であるとは，I がクラスラベルに矛盾している状態に対応しており，あるデータ $\boldsymbol{x}_i$ と $\boldsymbol{x}_{i'}$ が存在し，$\boldsymbol{x}_i = \boldsymbol{x}_{i'}$ かつ $y_i \neq y_{i'}$ を満たすことをいう．それ以外の場合は $\mathrm{Bn}(I) = 0$ とし，このとき I はクラスラベルに対して**無矛盾**（consistent）である．

離散化アルゴリズムの疑似コードを，Algorithm 1 に示す．以下では，この動作について説明する．入力データは，[0,1] に正規化された連続値を取る特徴ベクトルとクラスラベルの組からなる集合 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$ であり，出力は離散化され 2 値になった特徴ベクトルの集合である．ただし，提案した離散化では，入力パラメータとして，離散化の分割回数の限度である $\alpha_{\max}$ を用いる．2 分割から順番に分割数 k を増やしながら，各分割で得られた 2 値ベクトルが無矛盾かどうか，すなわち $\mathrm{Bn}(Z^k) = 0$ かどうかを調べていく．$k = 2$ から始まり，$k = \alpha$ のときの特徴集合 Z^α に対し，$\mathrm{Bn}(Z^\alpha) = 0$ であれば分割をやめ，離散化後の全特徴からなる集合 $S = Z^2 \cup Z^3 \cup \ldots Z^\alpha$ を出力する．$\mathrm{Bn}(Z^k) = 1$ であれば k に $k + 1$ を代入し，初めにもどる．しかし，$k = \alpha_{\max}$ となっても，$\mathrm{Bn}(Z^k) = 1$ となる場合，そこで分割をやめ，$S = Z^2 \cup Z^3 \cup \cdots \cup Z^{\alpha_{\max}}$ を出力する．

2.2 Super-CWC による特徴選択

提案手法では，離散化で得た 2 値ベクトルに対して，Super-CWC アルゴリズムと呼ばれる特徴選択手法を適用することで，クラス分類に不必要な特徴を削除する．この節では Super-CWC アルゴリズムの概要について説明する．

Super-CWC アルゴリズム [6] は，2015 年に提案された 2 値データに対する特徴選択手法である．Super-CWC アルゴリズムでは，$\mathrm{Bn}(I) = 0$ を満たす特徴の組合せ I を 2 分探索を用いて効率的に探すことによって，クラスラベルの判別において不要な特徴を削除し，必要最低限の特徴集合を選択することに成功している．

Super-CWC への入力データは，n 次元の 2 値ベクトル $\boldsymbol{z}_i = (z_i^1, z_i^2, \ldots, z_i^n) \in \{0, 1\}^n$ とクラスのラベル $y_i \in \{0, 1\}$ の組 $(\boldsymbol{z}_i, y_i)$ $(i = 1, \ldots, N)$ である．また，特徴ごとにデータを表す際は，N 次元の 2 値ベクトル $\boldsymbol{z}^j = (z_1^j, z_2^j, \ldots, z_N^j)^{\mathrm{T}}$，クラスラベルは，$N$ 次元のベクトル $\boldsymbol{y} = (y_1, y_2 \ldots, y_N)^{\mathrm{T}}$ のように表現する．また，入力データのすべての特徴からなるインデックス集合を $S = \{j \mid j = 1, 2, \ldots, n\}$ とする．出力された特徴集合を $I \subseteq S$ とし，I を用いた特徴ベクトルの集合は $\{\boldsymbol{z}^j \mid j \in I\}$ と表される．このとき，Super-CWC は以下の問題を解く．

$$\underset{I \subseteq S, \mathrm{Bn}(I)=0}{\operatorname{argmim}} |I|.$$

つまり，出力は $\mathrm{Bn}(I) = 0$ を満たす無矛盾な $I \subseteq S$ の中で，最小サイズのものである．

また，Super-CWC では，Symmetric Uncertainty [2]（以下では SU と書く）と呼ばれる，正規化された相互情報量を用いて特徴を選択する．SU は，以下の式で定義される．

$$\mathrm{SU}(\boldsymbol{z}^j, \boldsymbol{y}) = 2\frac{I(\boldsymbol{z}^j; \boldsymbol{y})}{H(\boldsymbol{z}^j) + H(\boldsymbol{y})}.$$

ここで，$H(\boldsymbol{z}^j)$，$H(\boldsymbol{y})$ はそれぞれ特徴ベクトル $\boldsymbol{z}^j$，クラスラベル $\boldsymbol{y}$ の情報エントロピー，$I(\boldsymbol{z}^j; \boldsymbol{y})$ は相互情報量を表す．定義から，SU は [0,1] の値をとる．SU の小さい特徴の方が，大きい特徴よりもクラスラベルとの矛盾が大きいため，Super-CWC アルゴリズムでは SU の小さい特徴から，S より除いていく．

3. 提案手法の動作例

この節では，提案手法の動作例を紹介する．表 1に示した $n = 2$，$N = 8$ のデータを入力とし，十分に離散化すると，各特徴を 4 分割することになる．動作例では簡単のため，出力 S から Z^2 と Z^3 を省略すると，変換した 2 値データは表 2のようになる．その後，2 値データを Super-CWC アルゴリズムに入力すると，5 つの特徴へと削減され，表 3のようになる．獲得した特徴集合から得られる 2 次元平面上での分類規則は，図 1における黄色と青色に塗られた部分で表される．なお，図 1には，表 1の連続値データもプロットした．

4. 評価実験

この節では，提案手法を人工データに適用して性能評価を行い，その結果について考察する．まず 4.1節では，実験に用いるデータ，実験設定，比較手法について述べる．その後，4.2節で実験結果を述べ，結果について考察する．

4.1 実験条件の設定

4.1.1 実験環境

実験に用いた計算機は，Intel Core i7-4790 CPU 3.60 GHz のプロセッサ，実験メモリ 8.00 GB，Windows 8 Enterprise 64

表1: 入力の連続値データ

	x^1	x^2	y
x_1	0.2	0.8	1
x_2	0.3	0.6	1
x_3	0.6	0.4	1
x_4	0.7	0.7	1
x_5	0.1	0.3	0
x_6	0.4	0.2	0
x_7	0.8	0.1	0
x_8	0.9	0.9	0

表2: 表 1を離散化した後の 2 値データ

	$z^{(1,0)}$	$z^{(1,1)}$	$z^{(1,2)}$	$z^{(1,3)}$	$z^{(2,0)}$	$z^{(2,1)}$	$z^{(2,2)}$	$z^{(2,3)}$	y
z_1	1	0	0	0	0	0	0	1	1
z_2	0	1	0	0	0	0	1	0	1
z_3	0	0	1	0	0	1	0	0	1
z_4	0	0	1	0	0	0	1	0	1
z_5	1	0	0	0	0	1	0	0	0
z_6	0	1	0	0	1	0	0	0	0
z_7	0	0	0	1	1	0	0	0	0
z_8	0	0	0	1	0	0	0	1	0

bit の OS が搭載された Project White 社製の exCompuer である．提案手法および比較対象は，ともに Python 言語（version 3.5.2）で記述されている．また，Super-CWC 本体は，著者によって公開されている Scala 言語での実装[*1]を用いた．

4.1.2 データセット

本稿では，人工的に生成したデータ B，H を用いて評価実験を行った．人工データ B は，2 つの正規分布からなるデータであり，それぞれの平均 μ と標準偏差 σ を $(\mu, \sigma) = (0, 1)$ と $(\mu, \sigma) = (2, 1)$ に設定した．また，人工データ H は，$(\mu, \sigma) = (0.5, 0.125)$ の n 次元正規分布に従うデータを 1 つ目のクラス，領域 $[0, 1]^n$ の一様分布に従うデータを 2 つ目のクラスとし，それぞれのサイズが 4 : 1 の割合となるようにサンプリングした．

実験は，人工データ B, H に対し，分割回数の限度 $\alpha_{\max} = 10$ のとき，特徴数 n，データ数 N を変化させ，10 分割の交差検証法で分類精度を評価した．実験における評価指標は，提案手法と比較手法における閾値数，分類精度である．ここで，閾値数とは，提案手法においては，得られた 2 値ベクトルの特徴数，すなわちベクトルの次元数とし，比較手法である決定木分析 CART においては，生成した木構造における葉以外の全ノード数とした．閾値数は，多いほど分類規則が複雑になっていると言えるため，規則の簡潔さの指標である．

4.1.3 比較手法

本実験では，提案手法を評価するための比較手法として，決定木分析 CART [1] を採用した．その理由は，CART が回帰問題を解くことができ，かつ分類規則を明示的に取得することが可能なためである．Python の機械学習ライブラリ scikit-learn [5] では，決定木分析の標準的手法として採用されており，本実験でもこの実装を用いた．

4.2 実験結果

人工データ B に対する実験では，特徴数を $n = 2$ に固定し，データ数を $N = 1,000$，$10,000$，$50,000$，$100,000$ に変化させた場合と，データ数を $N = 10,000$ に固定し，特徴数を

*1 https://github.com/tkub/scwc/

表3: 表 2から Super-CWC により獲得した 2 値データ

	$z^{(1,2)}$	$z^{(1,3)}$	$z^{(2,0)}$	$z^{(2,1)}$	$z^{(2,2)}$	y
z_1	0	0	0	0	0	1
z_2	0	0	0	0	1	1
z_3	1	0	0	1	0	1
z_4	1	0	0	0	1	1
z_5	0	0	0	1	0	0
z_6	0	0	1	0	0	0
z_7	0	1	1	0	0	0
z_8	0	1	0	0	0	0

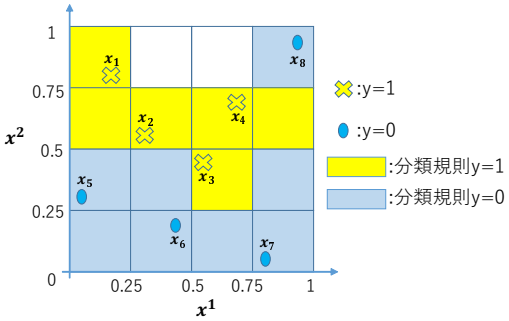


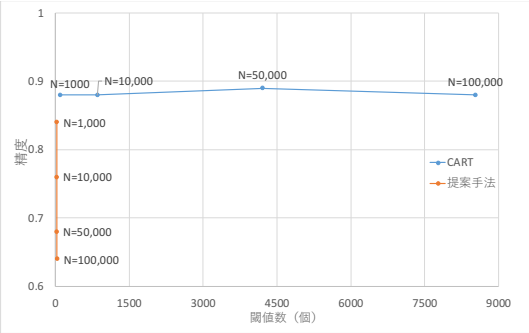
図1: 表 3に対応する 2 次元平面上の領域

$n = 2, 3, 4, 5, 10, 50, 100$ に変化させた場合の閾値数，分類精度を測定した．結果を図 2に示す．横軸は閾値数，縦軸は分類精度である．図 2aにおいて，データ数を 1,000 個から 100,000 個に変化させたとき，CART では，分類精度を 0.88 付近で保ちつつ，閾値数が 94.4 から 8530.3 へ約 90 倍に増加したが，提案手法では，分類精度は 0.84 から 0.64 に低下したものの，閾値数は 27.5 から 35.7 へと 1.3 倍の増加で抑えられた．図 2bにおいて，特徴数を 2 から 100 に変化させたとき，CART では，分類精度が 0.88 から 0.98 へ上昇し，閾値数が 852.2 から 60.8 へ 1/14 に減少したことに対し，提案手法の分類精度は，$n = 2$ から 3 で 0.76 から 0.85 へ上昇し，そ $n = 4$ では 0.72 へ低下したが，その後 0.97 へ上昇し，閾値数は，$n = 2$ から 3 では 28.6 から 41.7 へ 1.5 倍に増加したが，その後 $n = 100$ では 12 へと 1/3.5 に減少した．

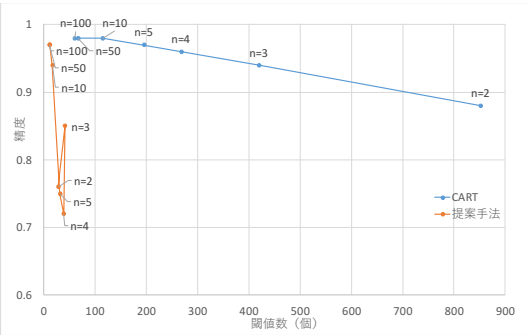
人工データ H に対する実験では，特徴数を $n = 2$ に固定して，データ数を $N = 1,000$，$10,000$，$50,000$，$100,000$ に変化させた場合と，データ数を $N = 10,000$ に固定して，特徴数を $n = 2, 3, 4, 5, 10, 50, 100$ に変化させた場合の閾値数，分類精度を測定した．結果を図 3に示す．横軸は閾値数，縦軸は分類精度である．図 3aにおいては，データ数を 1,000 から 100,000 へ増やしたとき，CART においては，分類精度を 0.85 付近で保ちつつ，閾値数が 110.2 から 11233 へ 102 倍に増加したことに対し，提案手法では，$N = 100$ から $10,000$ では分類精度が 0.85 から 0.9 に上昇したあと 0.87 に低下したものの，閾値数は 32.8 から 54 へ 1.6 倍の増加で抑えられた．図 3bにおいては，特徴数を 2 から 100 へ増やしたとき，CART においては，精度が 0.86 から 0.95 へ上昇し，閾値数は 1082.2 から 154.9 へ 1/7 に減少したことに対し，提案手法では，分類精度は 0.9 から 0.8 へ低下し，閾値数は 51 から 21.5 へ 1/2.4 に減少した．

4.3 実験の考察

図 2a，3aからデータ数を増加させたとき，CART では閾値数が線形に増加するが，提案手法ではデータ数の増加にはあまり影響されず，CART より大幅に少ない閾値数で分類を達成し

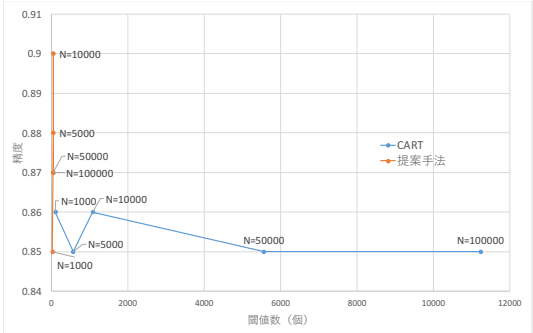


(a) データ数 N を変化させた場合

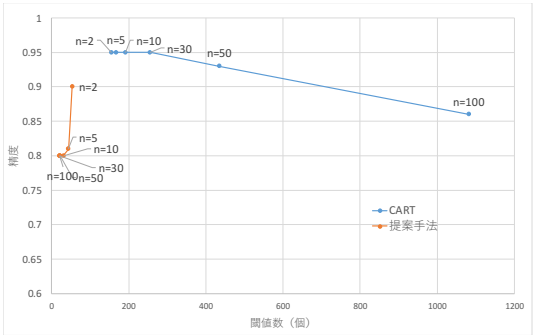


(b) 特徴数 n を変化させた場合

図2: データ B に対する実験結果



(a) データ数 N を変化させた場合



(b) 特徴数 n を変化させた場合

図3: データ H に対する実験結果

ている．これは，提案手法における離散化が，データの疎密によらずに分割し，かつ Super-CWC で不要な特徴を削除しているためであると考えられる．ただし，データ B において，データ数を増やしたとき，提案手法は分類精度が低下する．これは，提案手法がデータ数を増やしても閾値数があまり変わらないため，データ数が多くなるほど正確に分類することが難しくなっているためであると考えられる．それに対しデータ H では，$N = 10,000$ までは分類精度が上がり，また $N = 1,000$ 以外では CART より精度が高いため，データ H のようなデータに対する分類は，提案手法の方が CART よりも適していると考えられる．

特徴数を増加させたとき，まず図 2bのデータ B に対する実験では，提案手法は CART より閾値数が少なく，分類精度も上昇している．これは，離散化後の区間数が分割回数に特徴数を乗じた値となるため，次元が大きくなると比較的少ない分割で無矛盾な特徴を獲得することができ，離散化後の区間の幅が狭くなりすぎないためであると考えられる．これに対して，図 3bのデータ H における実験では，提案手法は特徴数が増えるほど閾値数が減少し，同時に分類精度も低下している．これは，データ H においては，特徴数が増え区間数が増えると，訓練データにフィットしすぎている規則を学習しているためと考えられる．したがって，同じ特徴数やデータ数でもデータによって過学習となる分割回数が違うため，分割回数の限度を適切に与える必要がある．

5. まとめ

本稿では，連続値データに対し解釈可能性の高いクラス分類規則を構築するため，連続値データを離散化して 2 値データに変換したあと，既存の特徴選択手法である Super-CWC アルゴリズムを適用する手法を提案し，その性能の評価を行った．その結果，提案手法を用いることで，特に閾値数において，CART より少ない閾値数で同程度の分類精度を達成することができた．また，データ H の特徴数が比較的小さいときには，データ数が増加すると，低い閾値数の簡潔な分類規則で，テストデータに対し高い精度で分類することができた．

一方，今回の評価実験においては，人工データに対してのみの評価であった．そのため，今後は実データに対して，有益な分類規則を得ることができるかを確かめていく必要がある．

参考文献

[1] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[2] Witten I. H., E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3 edition, 2011.

[3] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3 edition, 2011.

[4] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[6] K. Shin, T. Kuboyama, T. Hashimoto, and D. Shepard. Super-CWC and super-LCC: Super fast feature selection algorithms. In *Proceedings of 2015 IEEE International Conference on Big Data*, pages 1–7, 2015.