決定木モデル

前川眞一

20230906,11,12,13,14,19,1007cot,08,11,17,1112,29, 20241008, 2024. 10. 8

1　モデル

１変数の基準変数と p 変数の説明変数のセット $(y_i, \boldsymbol{x}_i), i = 1, 2, \ldots, n,$ が与えられた時に、説明変数が含まれる p 次元空間をある分類規則 $\boldsymbol{\xi}$ に従い G 個の互いに背反な矩形領域 $R_1, R_2, \ldots, R_G$ に分け、領域 R_j に含まれる $\boldsymbol{x}_i$ に対応する y_i の値の関数 (例えば平均や最頻値等）でその領域に含まれる個体の基準変数の予測値 $\widehat{y_i}(\boldsymbol{x}_i)$ とする方法が決定木モデル (tree models, tree-based models, decision tree models) である。

たとえば、$p = 2$ で x_1 が値域 1,2,3,4 を持つ数値的変数、x_2 が値域 a, b, c, d を持つカテゴリカルな変数の場合、以下の様な分類規則を考えることが出来る。

$$\begin{array}{lll} \text{if} \quad 1 \le x_1 \le 2 \quad \& \quad x_2 = a \text{ or } b & & R_1 \\ \text{if} \quad 1 \le x_1 \le 2 \quad \& \quad x_2 = c & & R_2 \\ \text{if} \quad 3 \le x_1 \le 4 \quad \& \quad x_2 = a \text{ or } c & & R_3 \\ \text{if} \quad 3 \le x_1 \le 4 \quad \& \quad x_2 = b & & R_4 \\ \text{if} \quad x_2 = d & & R_5 \end{array} \tag{1}$$

基準変数 y が連続量の場合は、回帰木 (regression tree) と呼ばれ、説明変数の値 $\boldsymbol{x}_i$ を与えた場合の個体の予測値 $\widehat{y_i}(\boldsymbol{x}_i)$ は、$\boldsymbol{x}_i$ により決まる領域に含まれる個体の基準変数の平均となる。これは指示関数 $I()$ （引数が真の時に 1、偽の時に 0 を取る関数）を使えば

$$\widehat{y_i}(\boldsymbol{x}_i) = \sum_{g=1}^{G} m_g \, I(\boldsymbol{x}_i \in R_g) \tag{2}$$

ただし、n_g を領域 R_g に含まれる個体数として、

$$m_g = \frac{1}{\sum_{i=1}^{n} I(\boldsymbol{x}_i \in R_g)} \sum_{i=1}^{n} y_i \, I(\boldsymbol{x}_i \in R_g) \tag{3}$$

$$= \frac{1}{n_g} \sum_{i \in g} y_i \tag{4}$$

である。なお、$\sum_{g=1}^{G} f_i \, I(\boldsymbol{x}_i \in R_g)$ という表現は R_g に含まれる個体のみを使って計算される量 f_i の値を示しているため、

$$\sum_{g=1}^{G} f_i \, I(\boldsymbol{x}_i \in R_g) = \sum_{i \in g} f_i \tag{5}$$

と書く場合もある。

いずれにしても、個体 i が第 g 群に分類される場合、その予測値は

$$\widehat{y_i}(\boldsymbol{x}_i) = m_g, \quad \text{if} \quad i \in g, \quad \text{i.e.,} \quad \boldsymbol{x}_i \in R_g \tag{6}$$

である。

他方、基準変数が名義変数 (カテゴリカル）な場合は分類木 (classification tree) と呼ばれるが、その水準数（カテゴリ数）を K として、y_i を one-hot-encoding でダミー変数に展開したものを $\boldsymbol{z}_i = \{z_{i1}, z_{i2}, \ldots, z_{iK}\}$ とすれば

$$\widehat{\boldsymbol{z}}_i(\boldsymbol{x}_i) = \sum_{g=1}^{G} \boldsymbol{m}_g \, I(\boldsymbol{x}_i \in R_g) \tag{7}$$

ただし

$$\boldsymbol{m}_g = \frac{1}{\sum_{i=1}^{n} I(\boldsymbol{x}_i \in R_g)} \sum_{i=1}^{n} \boldsymbol{z}_i \, I(\boldsymbol{x}_i \in R_g) \tag{8}$$

$$= \frac{1}{n_g} \sum_{i \in g} \boldsymbol{z}_i \tag{9}$$

すなわち、$\boldsymbol{m}_g = \{m_{g1}, m_{g2}, \ldots, m_{gK}\}$ として

$$m_{gk} = \frac{1}{\sum_{i=1}^{n} I(\boldsymbol{x}_i \in R_g)} \sum_{i=1}^{n} I(y_i = k) \, I(\boldsymbol{x}_i \in R_g) \tag{10}$$

$$= \frac{1}{n_g} \sum_{i \in g} z_{ik} \;=\; p_{gk} \tag{11}$$

である。これは領域 R_g に分類される個体の基準変数の各カテゴリの生起確率 p_{gk} を予測値としていることに他ならない。すなわち

$$\widehat{\boldsymbol{z}}_i(\boldsymbol{x}_i) = \boldsymbol{m}_g, \quad \text{if} \quad i \in g, \quad \text{i.e.,} \quad \boldsymbol{x}_i \in R_g \tag{12}$$

である。

他方、元の y_i と同じ尺度の予測値 $\widehat{y}_i$ が欲しい場合には、各領域に含まれる y の最頻値を予測値とすれば、

$$\widehat{y}_i(\boldsymbol{x}_i) = \sum_{g=1}^{G} m_g^{(*)} \, I(\boldsymbol{x}_i \in R_g) \tag{13}$$

ただし

$$m_g^{(*)} = \mathrm{mode}(y_i, \; \boldsymbol{x}_i \in R_g) \tag{14}$$

であるが、これは $\boldsymbol{m}_g$ の K 個の要素のうち、最大の生起確率を持つ水準を予測値としていることになる。

決定木は次節に述べるアルゴリズムを用いて算出されるが、それが得られた場合には、ある個体の説明変数の値 $\boldsymbol{x}_i$ を与えた場合のモデルの予測値 $\widehat{y}_i$ を計算することが出来る。この関係を

$$\widehat{y}_i(\boldsymbol{x}_i) = T(\boldsymbol{x}_i | \boldsymbol{\xi}) \tag{15}$$

と書く。ただし、T は分類規則 $\boldsymbol{\xi}$ を与えたときに説明変数 $\boldsymbol{x}_i$ を持つ個体を G 個の領域 $R_g, \; g = 1, 2, \ldots, G$ のどれかに分類する関数であり、$\boldsymbol{\xi}$ はそのパラメタ、すなわち、分類規則そのものである。

新しい個体のデータから既存の決定木を用いて予測値を計算する場合には、上記の $\boldsymbol{x}_i$ に新しい説明変数の値を入れれば良いが、これは、すなわち、この決定木の分類規則を新たな個体のデータに適用してその個体を何処かの領域に分類し、その領域の予測値を新たな個体の予測値とすることに等しい。このアルゴリズムの概略をパネル 4 に示した。

2　決定木作成のアルゴリズム

2.1　不純度の指標

決定木モデルでは、説明変数の各領域は各説明変数の値で２領域に分割することを繰り返すことにより最終的に G 個の領域を得るが、その際、各領域に含まれる個体の基準変数 y の値がなるべく均一になるように（純粋になるように、ピュアーになるように）分割をする変数とその分割点を決めていく。これは、各領域に含まれる y の値が似ていることを意味するが、その意味で不純度 (impurity) という用語が用いられる。すなわち、y の値が互いに似ていれば不純度が低いことになる。例えば、連続的な基準変数の場合には、ある領域に含まれる個体の基準変数の値の不純度はそこ領域における基準変数 y の分散で表すことが出来るが、基準変数の予測値 $\widehat{y}$ がその領域における y の平均であるため、これはその領域におけるデータとモデルの予測値との差の２乗和に比例する。カテゴリカルな基準変数の場合は、不純度の指標として、以下のどれかが用いられる。

- Gini 係数

$$G_g = \sum_{k=1}^{K} p_{gk}(1 - p_{gk}) \tag{16}$$

ただし、n_{gk} を領域 g に分類された個体のうちカテゴリ k に属する個体の数として、

$$p_{gk} = \frac{n_{gk}}{n_g} \tag{17}$$

である。Gini 係数は、領域に含まれる全ての個体の基準変数の値が等しい場合に、すなわち $p_{gk}, k = 1, 2, \ldots, k$ のどれか一つが 1 で他が 0 である場合に、その最小値 0 を取る。

- エントロピー

$$E_g = -\sum_{k=1}^{K} p_{gk} \log p_{gk} \tag{18}$$

エントロピーで表される不純度の最小値も、その領域に含まれる個体の基準変数の値が同一の場合、すなわち全く純粋な領域である場合に 0 となる。

- 誤り率

$$er_g = \sum_{i \in g} I(y_{ig} \neq \hat{y}_{ig}) \tag{19}$$

ただし、y_{ig} は領域 g に含まれる個体 i の予測値、$\hat{y}_{ig}$ は最頻値を用いた予測値である。すなわち、この指標は、ある領域で、基準変数の値を間違えて予測された個体の数を表している。

2.1.1　多項分布の尤度との関係

ちなみにダミー展開した基準変数 $\boldsymbol{z}_i$ の予測値 $\widehat{\boldsymbol{z}}_i(\boldsymbol{x}_i)$ は説明変数が $\boldsymbol{x}_i$ の個体の基準変数の各水準の条件付き生起確率であると見なせるため

$$\pi_k(\boldsymbol{x}_i) \equiv \Pr\left(\boldsymbol{Z}_i = \boldsymbol{z}_i | \boldsymbol{x}_i\right) \;=\; \widehat{\boldsymbol{z}}_i(\boldsymbol{x}_i) \tag{20}$$

と置けば多項分布を仮定した場合の尤度関数は

$$L = \prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k(\boldsymbol{x}_i)^{z_{ik}} \tag{21}$$

また、対数尤度は

$$\log L = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \pi_k(\boldsymbol{x}_i) \tag{22}$$

であるが、分類木の場合は $\boldsymbol{x}_i \in R_g$ であれば $\pi_k(\boldsymbol{x}_i) = p_{gk}$ とは定数となるため、$\log L$ は以下のようになる。

$$\log L = \sum_{g=1}^{G} \sum_{i \in g} \sum_{k=1}^{K} z_{ik} \log \pi_k(\boldsymbol{x}_i) \ = \ \sum_{g=1}^{G} \sum_{k=1}^{K} \sum_{i \in g} z_{ik} \ \log p_{gk} \tag{23}$$

$$= \sum_{g=1}^{G} \sum_{k=1}^{K} n_{gk} \ \log p_{gk} \ = \ \sum_{g=1}^{G} \sum_{k=1}^{K} n_g \ p_{gk} \ \log p_{gk} \tag{24}$$

$$= \sum_{g=1}^{G} n_g \sum_{k=1}^{K} p_{gk} \ \log p_{gk} \ = \ -\sum_{g=1}^{G} n_g E_g \tag{25}$$

すなわち、多項分布の尤度を最大化することは各領域に含まれる個体数で重みづけたエントロピーの和を最小にすることに等しい。また、この意味で。分類木の結果はロジスティク回帰や多項ロジスティク回帰や多項プロビット回帰の結果と比較可能である。

2.2 決定木の作成

通常、決定木は、分割に際して一つの説明変数に着目し、その値に基づき最も不純度が減少するように各個体を二つの領域へ分類する、という操作を繰り返して作られる。

グラフ理論の用語を使えば、全個体が含まれるノードから始めてそれを不純度を最も減少させる様に二つの子ノード (left ノードと right ノード）に分割する。次に、left ノードに着目し、同様の手続きでそれを２分割する、という操作を繰り返すことになる。ただし、あるノードの分割に際して、一つの変数しか考慮しない方法は必ずしも最適な分割をもたらす方法ではないが、このような方法は greedy algorithm （貪欲法）と呼ばれ、簡便であるためよく用いられる。

着目した説明変数が連続的な変数の場合には、その変数の値の範囲にある閾値を定め、ある個体のその説明変数の値が閾値未満であれば left ノードへ、閾値以上であれば right ノードへと分割すれば良い。閾値の候補としては、例えばその説明変数のユニークな値のどれかを用いれば良い。この場合、説明変数のユニークな値を Q とすれば、$Q-1$ 回の分割を試すことにより、この説明変数を用いた場合の最適な分割が求められる。なお、Q の値が大きい場合には、適当に間引けば良い。

着目した説明変数がカテゴリカルな変数の場合、閾値を定めて２分割を行うことが出来ないが、その説明変数の水準数が C の場合、$2^{C-1}-1$ 回の全ての分割を試す方法がある。例えば、カテゴリが a,b,c,d の $C=4$ 個の場合には、

$$(a,(b,c,d)),(b,(a,c,d)),(c,(a,b,d)),(d,(a,b,c)),$$
$$((a,b),(c,d)),((a,c),(b,d)),((a,d),(b,c))$$

の 7 通りの分割が考えられるが、これは、例えば、((a,b),(c,d)) の場合は、説明変数の値が a または b の個体を left ノードに、それ以外を right ノードに分類することを意味している。水準数 C が小さい場合にはこれで良いが、C が大きくなっていくに従って、所謂、組合せ爆発が生じるため、何らかの工夫が必要となる。

2.2.1 カテゴリカルな説明変数の数値化

その工夫としては、何らかの方法を用いてカテゴリカルな説明変数を連続的な説明変数に変換すれば良いが、基準変数 y が連続量もしくは２値のカテゴリカルな変数の場合には、 説明変数の C 個の値 (カテゴリ、水準）ごとに y の平均を計算し、その値を説明変数の各水準の数値としたものが最適な数値化された変数であることが知られている。ただし、２値のカテゴリカルな基準変数は 0 or 1 でコーディングされているものとする。(Fisher, 1958)

カテゴリカルな説明変数の数値化の方法としては以下の様な方法が提案されている。

0. 説明変数の one-hot encoding によるダミー変数化
   説明変数を one-hot encoding により C 列のダミー変数行列に展開したものを新たな数値的説明変数として用いる。
1. 基準変数が水準数３以上のカテゴリカルな変数の場合 1 （判別分析）
   説明変数を C 列のダミー変数行列に展開したものを $\boldsymbol{X}_d$ として y と $\boldsymbol{X}_d$ の間で線形判別分析 LDA（正準判別分析）を行い、その第１判別関数の値を数値化された説明変数の値とする。この方法は Loh & Vanichssetakul (1988) により提案されたものである。
2. 基準変数が水準数３以上のカテゴリカルな変数の場合 2 （判別分析）
   判別関数は一般的に複数存在するので第２判別関数まで取り、それに加えて、第１判別関数と第２判別関数の和と差で定義される 4 つの数値化された説明変数を作る。
3. 基準変数が水準数３以上のカテゴリカルな変数の場合 3 （水準ごとの２値化）
   基準変数が２値の場合は上記の方法 1 により最適な数値化の方法が存在するため、基準変数の水準数が K 個の場合、一つの水準とそれ以外という形で K 個の２分割を行うことが出来、それぞれに対して、方法１で数値的説明変数を作ることが出来る。この方法は、Prokhorenkova, et al. (2018) によれば Micci-Barreca, D. (2001) により提案された方法である。
4. 基準変数が水準数３以上のカテゴリカルな変数の場合 4 （完全な２値化）
   基準変数の水準数 K がそれほど多くない場合は、全ての２値化のための $2^{(K-1)}-1$ 個の２分割を試すことが可能である。たとえば、$K=4$ で各水準が (A,B,C,D) とコード化されている場合には、以下の７通りの分割

$$(A,(B,C,D)),(B,(A,C,D)),(C,(A,B,D)),(D,(A,B,C)),$$
$$((A,B),(C,D)),((A,C),(B,D)),(A,D),(B,C))$$

   が可能であるため、７通りの異なる２値の基準変数を作成することが可能であるが、このそれぞれに関して、方法 1 により説明変数の水準ごとの平均を求めたものが数値化された説明変数となる。

なお、方法２は方法 1 を含むが、この二つの判別分析を行う方法は K の数が大きい場合には大きな非対称行列の固有値問題を解く必要があり最適性が保証されているわけではないため、必ずしも良い結果をもたらすとは限らない。他方、方法４は方法３を含むが、基準変数の水準数が

ある程度少ない限り、説明変数の水準数が多い場合でも利用可能な方法である。

Table 1 にこれらの方法の数値例を示すが、データはパネル data に示した x と y であり、両者ともにカテゴリカルな変数で、基準変数 y の水準数は $K = 4$ で説明変数 x の水準数は $C = 5$ である。また、表中 $\boldsymbol{Y}$ に、基準変数の 4 水準を用いて作られる全ての２分割のパタンを示すダミー行列 $\boldsymbol{Y}$ を示した。Y 行列の qx1 列から qx4 列はそれぞれ、A とそれ以外、B とそれ以外、C とそれ以外、そして D とそれ以外の２分割に対応する列であり、qx5 列から qx8 列は A または B とそれ以外、A または B とそれ以外、そして B または C とそれ以外の２分割に対応する列である。Table 2 のパネル q2 の qx1 列と qx2 列に y と x の正準判別関数のうち x のユ

Table1　説明変数の数値化の例：データ

データ		Y 行列						
x	y	qx1(A)	qx2(B)	qx3(C)	qx4(D)	qx5(AB)	qx6(AC)	qx7(BC)
1	A	1	0	0	0	1	1	0
1	B	0	1	0	0	1	0	1
1	C	0	0	1	0	0	1	1
1	D	0	0	0	1	0	0	0
2	A	1	0	0	0	1	1	0
2	B	0	1	0	0	1	0	1
2	B	0	1	0	0	1	0	1
2	C	0	0	1	0	0	1	1
3	B	0	1	0	0	1	0	1
3	B	0	1	0	0	1	0	1
3	B	0	1	0	0	1	0	1
3	C	0	0	1	0	0	1	1
4	A	1	0	0	0	1	1	0
4	A	1	0	0	0	1	1	0
4	B	0	1	0	0	1	0	1
4	D	0	0	0	1	0	0	0
5	B	0	1	0	0	1	0	1
5	B	0	1	0	0	1	0	1
5	D	0	0	0	1	0	0	0

ニークな値の部分のみの値を２次元まで示したが、説明変数ダミー変数行列としては 16 x 4 の行列のうち水準 1 に対応する第１列を落としたものを用いた。このデータでは判別関数は二つ求まり、qx3 列と qx4 列にそれぞれの和と差を示した。実際には、パネル data に示した x をこれらの値に置き換えたものが数値化された説明変数となる。

これを利用すると、例えば、qx1 を使えば、閾値が 0 の場合には個体を x=(1,2,3,4) と x=(5) の群に、閾値が -1.362 の場合は x=(2,3) と x=(1,4,5) の群に分割することが出来る。また、qx2 を使えば、閾値が 0 の場合は x=(1,4) と x=(2,3,5) の群に、閾値が 1.709 の場合には個体が x=(1,2,3,4) と x=(5) の群に分割することが出来る。

なお、閾値を設定して個体を２群に分けるという観点からは、パネル p2 rank に示したように

x のランクがわかれば良いが、qx2 列と qx3 列の二つの列は同順位となり冗長であるため qx3 列は省いてある。また、数値化された変数またはランクと閾値の組み合わせでは、同じ個体の分割が得られる可能性があるが、これは事前には避けることが出来ない。

次に Table 2 のパネル q4 に、基準変数の全ての水準の組み合わせを用いて基準変数を２値化する方法の結果を示した。なお、x のユニークな値のみを表示しているが、実際には、パネル data に示した x をこれらの値に置き換えたものが数値化された説明変数となる。

２値化のパタンは先に示した様に

> A とそれ以外、B とそれ以外、 C とそれ以外、D とそれ以外
> A or B とそれ以外、 A or C とそれ以外、 A or D とそれ以外
> （なお、A or D とそれ以外は B or C とそれ以外と同じであるため、以下ではこの後者の表現を用いる。）

の 7 通りでありそれらは $\boldsymbol{Y}$ 行列の各列に対応しているが、これの x の水準ごとの平均が求める方法４の数値化された変数の値となる。例えば、パネル q4 の 1 行 qx1 列の 0.25 は、y を A とそれ以外で２値化した際に、x=1 の部分の y=A の割合 $1/4$ である。また、2 行 qx1 列の 0.00 は同様に y を２値化した場合の x=2 の部分の y=A の割合 $0/4$ である。また、2 行 qx5 列の 0.50 は y を A または B とそれ以外で２値化した際に、x=2 の部分の y=A または B の割合 $2/4$ であり、また、5 行 qx7 列の 0.25 は y を B または C とそれ以外に２値化した場合の x=5 の部分の B または C の割合 $1/4$ である。したがって、パネル q4 の第 qx1 列を使えば、閾値

Table2　方法２：判別分析

方法２

	qx1	qx2	qx3	qx4
x1	0.00	0.00	0.00	0.00
x2	0.90	-0.77	0.13	1.67
x3	1.97	0.16	2.13	1.81
x4	-1.13	0.21	-0.92	-1.34
x5	0.72	2.33	3.05	-1.61

方法２　ランク

	qx1	qx2	qx3	q4
x1	2	2	2	3
x2	4	1	3	4
x3	5	3	4	5
x4	1	4	1	2
x5	3	5	5	1

を 0 として個体を x=(2,3) の群と x=(1,4,5) の２群に分割することが出来るし、閾値を 0.25 とすれば (1,2,3,4) と (5) への分割となる。また第 qx5 列を使えば、閾値を 0.5 とすれば個体を x=(1,2,3,4) と x=(5) の群に分割することが出来るし、第 qx7 列を使えば、閾値を 0.75 とすれば個体を x=(2,3) と x=(1,4,5) の群に、閾値を 0.50 とすれば x=(1,4,5) と x=(2,3) の群に分割することが出来る。

なお、2 分割の観点からは、全ての数値化された変数を 1 から K までのランクに変換したパネル q4 rank を用いても同様であるが、ランクに変換することにより、qx4 の順位が qx3 のランクが等しくなるため、このパネルから qx4 は省いてある。

また、先ほどの例が示すように、数値化された変数またはランクと閾値の組み合わせでは、例えば qx3 と qx5 のように、同じ個体の分割が得られる可能性があるが、これは事前に簡単には避けることが出来ない。

Table3　方法４：２値化

方法４

	qx1(A)	qx2(B)	qx3(C)	qx4(D)	qx5(AB)	qx6(AC)	qx7(BC)
x1	0.25	0.25	0.25	0.25	0.50	0.50	0.50
x2	0.25	0.50	0.25	0.00	0.75	0.50	0.75
x3	0.00	0.75	0.25	0.00	0.75	0.25	1.00
x4	0.50	0.25	0.00	0.25	0.75	0.50	0.25
x5	0.00	0.67	0.00	0.33	0.67	0.00	0.67

方法４　ランク

	qx1(A)	qx2(B)	qx3(C)	qx4(D)	qx5(AB)	qx6(AC)	qx7(BC)
x1	3	1	3	3	1	3	2
x2	3	3	3	1	3	3	4
x3	1	5	3	1	3	2	5
x4	5	1	1	3	3	3	1
x5	1	4	1	5	2	1	3

2.3　決定木作成のアルゴリズム

これらをまとめると、決定木の作成のアルゴリズムは以下の様になる。まず、基本となるのは、一つのノード（親ノード）を与えられた時に、それを二つのノードに分割する操作である。すなわち、p 個の説明変数の一つ一つに着目し、その説明変数の元でも最適な分割を決定し、作られた二つのノードにおける不純度を各ノードの個体数で重みづけた和として得られるその変数による分割の不純度を計算する。これを、全ての説明変数ごとに計算し、最も不純度が低くなる説明変数での分割を採用するという操作である。

なお、分割に際して、同時に複数の変数を考慮する方法や、３分割以上の分割を行う方法等も提案されているが、ここでは上記の、一つの説明変数を用いた２分割の方法のみを取り上げる。

したがって、一つの分割で親ノードから新たに二つのノードが作成され、そのそれぞれがまた親ノードとなり二つの子ノードを作成する、という形で分割が進んでいくことになるが、このような分割は、分割を行う関数を再帰的に利用するか、もしくはスタック (Last In First Out, First In Last Out) を利用することにより実現することが出来る。

このスタックを利用した深度優先探索 (depth-first-search, DFS) を用いた決定木作成のアルゴリズムをパネル 1 から 3 に示したが、パネル 1 のスタックを利用する全体の流れは鈴木 (2020) を参考にした。また、パネル 2 や 3 に示した分割を行う関数は、実際には、全ての個体に関して比較をするのではなく、そのユニークな値のみを用いて比較を行うことにより効率化が図れる。

（なお、このアルゴリズムで、基準変数の不純度ではなく、説明変数の不純度を用いれば、所謂、分割型の階層的クラスター分析の解を求めることが可能となるが、クラスターとして得られる説明変数の領域は決定木同様の矩形領域となる。）

1 : 全体の流れ

```
node はその要素が各ノードに対応するリストで、各要素には
      親ノードの番号、含まれる個体番号、分割に使う変数番号とその閾値
      もしくは left に分類されるべきカテゴリの組み合わせ等を記録。
stack はその要素が各ノードに対応するリストで、各要素には
      親ノードの番号、含まれる個体番号、不純度の値、ノードの深度等を記録。

stack の初期化：全ての個体を含む最上位のルートノードを入れる。
                （ノード番号 = 0、深度 = 0）
node  の初期化：空リスト。

# ノード番号
r = 0

while ( stack が空でない ){

   POP して stack から一つのノードを取り出す。

   r = r+1

   ノードを分割する関数を利用してそのノードを left と right に分割し、
   分割に付随する情報を受け取る。

   分割に際する不純度の減少分、現在のノードに含まれる個体数等を考慮して、
   分割の効果が無ければ、{
     そのノードを終端ノードとして ŷ を計算し node[r] に記録して次へ。
     }
   分割の効果があれば、{
     深度を一つ増やす。
     そのノードを中間ノードとして node[r] に記録する。
     stack に right を PUSH する。
     stack に left  を PUSH する。
    }
}

後処理として、node に含まれる中間ノードにその子の名前を付加する。

決定木のパラメタとして node を出力する。
```

以下のパネル 2 と 3 に、パネル 1 で参照されている一つのノードを二つに分割する関数の概要を示す。パネル 2 はカテゴリカルな説明変数を数値化して分割を行う場合であり、パネル 3 はカテゴリカルな説明変数のみが与えらとして、その水準の全ての組み合わせを試す方法の概要であるが、実際には、この二つの方法をオプションにより切り替えて利用することになる。

2 : ノードの分割のための関数（カテゴリカルな説明変数の数値化）

```
X はーつのノードに含まれる説明変数行列
y はーつのノードに含まれる基準変数
impfunc はあるノードに含まれる個体の基準変数 y を与えたときに
        不純度の指標を返す関数

BestScore = 大きな値

for( j in 1:ncol(X) ){

  xsj = X[,j]  # j 番目の説明変数の値

  もしも j 番目の説明変数がカテゴリカルな場合には
  それを何らかの方法で数値化したものを新たに xsj とする。
  数値化の候補が複数ある場合を考慮して xsj を行列とする。

  for( k in 1:ncol(xsj){

    uxsj = xsj[,k] の第 k 列のユニークな要素からなるベクトル
                   もしくはそれを間引きしたもの

    for( i in 1:length(uxsj) ){

      locLL = xsj[,k] <= uxsj[i] となる個体の番号からなるベクトル
      locRR = xsj[,k] >  uxsj[i] となる個体の番号からなるベクトル

      Score = n_left*impfunc(y[locLL]) + n_right*impfunc(y[locRR])

      if( Score < BestScore ){

        BestScore = Score
        Bestx = j
        BestThreshold = uxsj[i]
        出力するリストに以下のものを保存する。
          BestScore, Bestx, BestThreshold, locLL, locRR

      }

    } # end of i loop

  } # end of k loop

} # end of j loop

保存したリストを返す。
```

3 : ノードの分割のための関数（カテゴリカルな説明変数の全ての水準の組み合わせ）

```
X はーつのノードに含まれる説明変数行列
y はーつのノードに含まれる基準変数
impfunc はあるノードに含まれる個体の基準変数 y を与えたときに
        不純度の指標を返す関数

BestScore = 大きな値

for( j in 1:ncol(X) ){

  第 j 番目のカテゴリカルな説明変数のベクトルを xsj=X[,j] とし、
  その水準数を C とする。


  for( k in 2^{C-1} - 1 ){


      locL = 第 k 番目の left に分類されるべき水準の値の組み合わせ
      locR = 第 k 番目の right に分類されるべき水準の値の組み合わせ

      locLL = xsj の値が locL に含まれる個体の番号からなるベクトル
      locRR = xsj の値が locR に含まれる個体の番号からなるベクトル

      Score = n_left*impfunc(y[locL]) + n_right*impfunc(y[locR])

      if( Score < BestScore ){

        BestScore = tempScore
        Bestx = j
        BestThreshold = uxsj[i]
        出力するリストに以下のものを保存する。
          BestScore, Bestx, locLL, locRR

      }


  } # end of k loop

} # end of j loop

保存したリストを返す。
```

2.4 予測値の計算

最後に、上記のアルゴリズムを用いて作成された決定木を用いて新しい個体を分類するためのアルゴリズムを以下に示す。作成された決定木の情報は node に含まれているものとする。これは Eq.(15) に示した T 関数そのものである。

決定木モデルにおいては、各個体は必ずどれか一つの領域へ分類されるため、ある個体がある領域に属する確率という概念はない。しかし、基準変数が水準数 K のカテゴリカルな変数の場合には、領域 g における基準変数の分布が確率 $p_{gk},\ k = 1, 2, \ldots, K,\ \sum_{k=1}^{K} p_{gk} = 1$ として与えられているため、その値が個体が K 個のカテゴリに属する確率となる。

4：既存の決定木を用いた予測値の計算

```
r = 1
while( node[r] が終端ノードではない ){

 そのノードの分割に関する情報（用いた変数、閾値、等）を取り出し、
 その分類規則に従って、 新しい個体をどちらかの子ノードに分類する。
 分類された子ノードの番号を r とする。

}
最終的に得られた第 r 番目のノード（終端ノード）の予測値を返す。
```

3 決定木の拡張

前節までに示した決定木は、結果の解釈が容易となる場合が多いので良く用いられるが、データの微小な変化により得られる決定木が大きく異なる場合があるという欠点を持つ。その欠点を克服する試みとして以下のものが提案されている。

- バギング
  データからブートストラップサンプルを B 個取り出し、それぞれを用いて B 個の決定木、$T(\boldsymbol{x}|\boldsymbol{\xi}_b), b = 1, 2, \ldots, B,$ を作成し、基準変数が連続量の場合には

$$\hat{y}_i = \frac{1}{B} \sum_{b=1}^{B} T(\boldsymbol{x}_i|\boldsymbol{\xi}_b) \tag{26}$$

  として最終的なモデルの予測値を計算する。基準変数がカテゴリカルな場合には B 個の予測値の最頻値を求める。（多数決）また、各カテゴリの確率の予測値を求める場合には、第 b 番目のサンプルで得られる確率の予測値を $\widehat{\boldsymbol{z}_i}^{(b)}(\boldsymbol{x}_i)$ とすれば

$$\widehat{\boldsymbol{z}_i}(\boldsymbol{x}_i) = \frac{1}{B} \sum_{b=1}^{B} \widehat{\boldsymbol{z}_i}^{(b)}(\boldsymbol{x}_i) \tag{27}$$

  となる。
  このような方法をアンサンブル学習と呼ぶ。
- ランダムフォーレスト (RF)
  バギングを行う際に、決定木の作成に用いる説明変数の数を少数に限定し、説明変数も毎回ランダムに選ぶ方法。
  予測値はバギングと同様に計算できる。
- ブースティング
  ？？？
- 潜在クラスモデルへの拡張 MoET (Mixture of Expert Trees)

References

Hastie, Tibshirani, & Friedman (2009) *The Elements of Statistical Learning: Data Mining,Inference,and Prediction.* 2nd edition.
url: https://hastie.su.domains/ElemStatLearn/

Loh, W. and Vanichsetakul, N. (1988). Tree structured classification via generalized discriminant analysis, Journal of the American Statistical Association 83: 715-728.

Micci-Barreca, D. (2001) A preprocessing scheme for high-cardinality categorical attributes in classifi-cation and prediction problems. *ACM SIGKDD Explorations Newsletter*, 3(1):27-32
url: https://zero.sci-hub.se/4157/2b8690a9a7cfcec82623173c75f43b9f/micci-barreca2001.pdf

Prokhorenkova, et al. (2018) *CatBoost: unbiased boosting with categorical features.*
https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf

鈴木護 (2020) 統計的機械学習の数理 100 問 with R 共立出版