ワンポイント講座

ID 3[†]

馬野 元秀*

エキスパートシステムを構築する際に、知識を専門家からどのようにして獲得するかは非常に難しいが重要な問題である。これを克服するために、知識工学の分野では、データから知識を獲得する方法がいろいろと提案されている。このとき、どのような形の知識を考えるかが重要であるが、分かり易く、ルール抽出が容易であることから、決定木がよく用いられている。

決定木とは、例えば、図1のようなもので、分類型の知識を表現するのに適している。決定木が知識として与えられていて、いまあるデータの分類クラスを知りたいとする。まず、一番上の節点(ルート節点)でその属性の値を調べる。そして、その値の枝をたどることにより次に調べる属性を知ることができる。これを次々と繰り返し、葉節点(下に枝がない節点)に到達すると、その分類のクラスを知ることができる。例えば、図1の決定木に対しては、まず「髪の色」を調べ、それが「黒」の場合には他の属性を調べなくても分類クラスが「-」であることが分かり、「赤」の場合にも同様に「+」であることが分かる。また、「ブロンド」の場合にはさらに「目の色」を調べると、それが「青」の場合には「+」で、「茶」の場合には「-」であることが分かる。さらに、決定木から容易に葉節点の数だけのルールを得ることができる。

それでは、図2のようなデータ集合 D が与えられたときに、図1のような決定木を作成するにはどのようにすればよいだろうか。データ集合 D

は、次の3つの属性と分類クラス

身長={高い, 低い}

髪の色={黒, 赤, ブロンド}

目の色={青, 茶}

分類クラス={+, -}

によって記述されている。このとき、どのような順序で属性を調べるかによりいろいろな決定木が

図1. 決定木の例

身長	髪の色	目の色	クラス
低い	ブロンド	青	+
高い	黒	茶	-
高い	黒	青	-
高い	赤	青	+
低い	黒	青	-
高い	ブロンド	茶	-
低い	ブロンド	茶	-
高い	ブロンド	青	+

図2. データ集合 D

[†] ID 3

* Motohide UMANO

大阪大学 基礎工学部システム工学科

Department of Systems Engineering, Osaka University

作られる。J.R. Quinlan によって 1979 年に提案された ID 3 (Interactive Dichotomizer 3) [1], [2] は「データを分類するときのテスト回数の期待値を最小にする決定木」を生成することを目的としている。そのために、獲得情報量 (相互情報量) の期待値を最大にする属性を決定木のテスト節点として選択している。

以下、図 2 のデータを用いて、ID 3 による決定木の作成方法を説明しよう。まず、 D の情報量 $I(D)$ を計算する。これは、各クラスの出現する確率を p_1 と p_2 とすると、

$$I(D) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \quad (1)$$

と定義できる。いまは、「+」クラスに属するものが 3 つで、「-」クラスに属するものが 5 つなので、それぞれのクラスの出現する確率は $3/8$ と $5/8$ となり、

$$I(D) = -\frac{3}{8} \log_2 \frac{3}{8} - \frac{5}{8} \log_2 \frac{5}{8} = 0.954 \text{ (bit)}$$

となる。 $I(D)$ の値は、式 (1) で p_1 と p_2 の値が等しいとき、すなわち、各クラスに属するデータの数が等しくて、クラスが「バラバラ」のときに、最大になることに注意する必要がある。

次に、各属性について、その属性でテストした場合の情報量を計算する。例えば、属性「身長」でテストすると考えると、データ D は図 3 のように分類されることになるので、「高い」と「低い」に対するデータ集合 $D_{\text{高い}}$ と $D_{\text{低い}}$ の情報量を計算する。これは

$$I(D_{\text{高い}}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.971 \text{ (bit)}$$

$$I(D_{\text{低い}}) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} = 0.918 \text{ (bit)}$$

となる。そして、属性「身長」でテストした後の情報量の期待値を求めると、 D での「高い」と「低い」の確率がそれぞれ $5/8$ と $3/8$ なので、

$$\begin{aligned} E(\text{身長}, D) &= \frac{5}{8} \times 0.971 + \frac{3}{8} \times 0.918 \\ &= 0.951 \text{ (bit)} \end{aligned}$$

となる。さらに、同じようにして、属性「髪の色」

低い・ブロンド・青：+
高い・黒・茶：-
高い・黒・青：-
高い・赤・青：+
低い・黒・青：-
高い・ブロンド・茶：-
低い・ブロンド・茶：-
高い・ブロンド・青：+

図3. 身長によりテストしたデータ

と「目の色」でテストした場合の情報量の期待値は、それぞれ、

$$\begin{aligned} E(\text{髪の色}, D) &= \frac{3}{8} \times 0 + \frac{1}{8} \times 0 + \frac{4}{8} \times 1 \\ &= 0.5 \text{ (bit)} \end{aligned}$$

$$\begin{aligned} E(\text{目の色}, D) &= \frac{5}{8} \times 0.971 + \frac{3}{8} \times 0 \\ &= 0.607 \text{ (bit)} \end{aligned}$$

となる。

そして、これらと D の情報量 $I(D)$ との差である獲得情報量 $G(\text{身長}, D)$ を計算すると、

$$\begin{aligned} G(\text{身長}, D) &= I(D) - E(\text{身長}, D) \\ &= 0.954 - 0.951 \\ &= 0.003 \text{ ビット} \end{aligned}$$

となる。そして、属性「髪の毛」と「目の色」での獲得情報量 $G(\text{髪の色}, D)$ 、 $G(\text{目の色}, D)$ を計算すると、

$$\begin{aligned} G(\text{髪の色}, D) &= 0.454 \text{ ビット} \\ G(\text{目の色}, D) &= 0.347 \text{ ビット} \end{aligned}$$

となる。そして、これらが最大となる属性を D のテストとして選択するので、いまは、「髪の色」が選択されることになる。

これを各節点が同じクラスのデータだけになるまで繰り返して適用すると、図 1 の決定木が得られる。さらに、この決定木から次の 4 個のルールを得ることも可能である。

r 1: if 髪の色='黒' then 分類='.'
 r 2: if 髪の色='赤' then 分類='+'
 r 3: if 髪の色='ブロンド' & 目の色='青'
       then 分類='+'
 r 4: if 髪の色='ブロンド' & 目の色='茶'
       then 分類='.'

以上のアルゴリズムをまとめると次のようになる。いま ℓ 個の属性 $A_1, A_2, \dots, A_\ell$ があり、それぞれ $A_i = \{a_{i1}, a_{i2}, \dots, a_{im}\}$ (m は A_i によって異なる) の属性値を持ち、分類クラスは $C = \{C_1, C_2, \dots, C_n\}$ とする。このような属性と分類クラスを持つ学習データの集合があるときに、ID 3 により決定木を作成するアルゴリズムは次のようになる。

1. すべての学習データを対応付けた節点(ルート節点)を生成する
2. 節点に対応付けられた学習データ集合 D がすべて同じ分類クラス C_k に属するならば、その節点を葉節点とし、分類クラス C_k をラベル付けする
3. そうでなければ、その節点は葉節点ではないので、次の処理を行なう
  - (1) 各属性 A_i に対して、その属性値により D を部分集合に分割し、獲得情報量 $G(A_i, D)$ を下の式で計算する
  - (2) 計算した獲得情報量が最大となる属性をこの節点のテスト属性 A_{max} とする
  - (3) テスト属性 A_{max} の各部分集合に対して、もとの節点と各部分集合の節点とを結ぶ枝に、対応する属性値 $a_{max,j}$ をラベル付けする
  - (4) 作成したすべての節点に対して、2 から実行する

ここで、獲得情報量は次の式で計算する。このとき、 D_{C_k} を分類クラスが C_k である D の部分集合とし、 $D_{a_{ij}}$ を属性 A_i の値が a_{ij} である D の部分集合とする。

$$G(A_i, D) = I(D) - E(A_i, D) \quad (2)$$

ただし、

$$I(D) = - \sum_{k=1}^n (p_k \cdot \log_2 p_k) \quad (3)$$

$$p_k = \frac{|D_{C_k}|}{|D|} \quad (4)$$

$$E(A_i, D) = \sum_{j=1}^m (p_{ij} \cdot I(D_{ij})) \quad (5)$$

$$p_{ij} = \frac{|D_{a_{ij}}|}{|D|} \quad (6)$$

ここで、 $|D|$ は D の要素数である。

アルゴリズムの 3 において計算される獲得情報量 $G(A_i, D)$ は負になることはなく、属性 A_i で分けても変化がないときに 0 となり、それ以外は正の値をとるので、確率分布どうしの離れ具合をあらわす「距離」に相当する量と考えられ、Kullback-Leibler の情報数とも言われる[3]。また、情報量 $I(D)$ には最大値 $\log n$ が存在し、それは確率がすべて等しいときである。

さらに、決定木からより少数のルールを抽出する方法やファジィ決定木の研究も行われている。

参 考 文 献

- [1] J.R. Quinlan (1979): "Discovering Rules by Induction from large Collections of Examples," in D. Michie (ed.): *Expert Systems in the Micro Electronics Age*, Edinburgh University Press
- [2] J.R. Quinlan (1986): "Induction of Decision Trees," *Machine Learning*, Vol.1, pp.81-106
- [3] 松原望(1977): 意思決定の基礎、朝倉書店

[問い合わせ先]

〒560 豊中市待兼山町 1-3

大阪大学 基礎工学部 システム工学科
馬野 元秀

☎: 06-844-1151(内)4629

☎: 06-857-7664

Internet: umano@sys.es.osaka-u.ac.jp