4L-10 決定木の緩和

MIURA, Takao / MORI, Mikihiro / 森, 幹浩 / 三浦, 孝夫 / 塩谷, 勇 / SHIOYA, Isamu

(出版者 / Publisher)
情報処理学会
(雑誌名 / Journal or Publication Title)
全国大会講演論文集
(号 / Number)
2
(開始ページ / Start Page)
169
(終了ページ / End Page)
170
(発行年 / Year)
1999-03-09

4L－10

決定木の緩和

森 幹浩[1] 三浦 孝夫[2] 塩谷 勇[1]

[1] 産能大学経営情報学部 [2] 法政大学工学部

1 まえがき

決定木生成アルゴリズムである C4.5 は、データに関する領域知識を必要としないので適用分野が広い。その反面、問題知識が存在するデータではその知識がうまく利用できない。そこで、本研究では、データに関するクラス間の意味階層を用い、クラスメンバーシップの緩和を行うことにより、今までC4.5では得られなかったような、意味まとまりの得やすい正確な決定木の生成を行う。

具体的には、クラスメンバーシップをより高いクラス階層のものに変更し、正しく分類が行える決定木にする。しかし、クラス階層間の一般化により、正確でも有用でない木が生成されてしまうかもしれない。例えば、全てのクラスをひとつのクラスに一般化してしまえば、全く役に立たない正確な木ができてしまう。つまり、有用な木を生成するためには、(1) エントロピーの意味での精度の保証、(2) 決定木生成の計算量のなるべく減らす、という2つの問題を考えなければならない。

2 クラスメンバーシップの緩和

先ず、オブジェクトを一行のデータとし、その性質を表すものを属性、目的を表すものをクラスとする。このオブジェクト(行)の集合を決定木生成の入力とする。また、属性が同じでクラスが異なるオブジェクトは重複行とする。それぞれのクラスはひとつの親を持つ ISA 階層をなしており、これを単一クラス階層と呼び、この例外として、最上位クラスを**OE**で表す。

決定木の複雑さを決める基準は木の深さとし、**枝刈り率**を α とする。そして、決定木生成中、α を越えた場合、処理を中断する。このとき、ISA 階層により、属性値を上位の概念に変更し、急速な木の収束を期待する。

最初の抽象化として、属性値はその属性値が所属する最下層のクラスに変更される。それでも決定木が閾値を越えている場合には、さらに上位にあるクラスへ変更していく。そして最後には **OE** に到達する。このプロセスにより、属性値はグループ化され、決定木は簡単化されていく。しかし、実際は情報を失っていくことになるので、エントロピーは減少していく。属性 A の属性値 $\{a_1,..,a_w\}$ によって $S_1,..,S_w$ と分解されていた集合 S は、属性値の変更により $S'_1,..,S'_{w'}, w' \leq w$ と結合される。例えば、S_1, S_2 の属性値がそれぞれ $A = "a_1", A = "a_2"$ であったとする。もし、a_1, a_2 が同一のクラスとなったら S_1, S_2 は同じ構成要素になる。このとき、A 以外の属性に関してはエントロピーの変動が起こらない。S_1 のエントロピー V_1 は式 $-(1/m_1)\Sigma_{i=1,..,q} n_i^1 \times \log_2(n_i^1/m_1)$ によって得られる。ここで、m_1 は集合 S_1 に含まれる要素数であり、n_i^1 はS_1 要素のうち、クラス c_i に所属する要素の数である。同様に、V_2 は式 $\log_2 m_2 - (1/m_2)\Sigma_{i=1,..,q} n_i^2 \log_2 n_i^2$ により求められる。S_1 と S_2 が結合したとき、新しいエントロピー V_{12} は、$-(1/(m_1+m_2)\Sigma_{i=1,..,q}(n_i^1+n_i^2)\times\log_2((n_i^1+n_i^2)/(m_1+m_2))$ となり、トータルのエントロピー変動は $V_{12}-(V_1+V_2)$ によって求められる。エントロピーは連続的に減少しないかもしれないが、最後には 0 になる (**OE** の場合)。

次に、どの属性を選択すれば良いのかを考える。集合 S におけるエントロピーを $E(S)$ とし、S が、属性 A の属性値 $\{a_1,..,a_w\}$ で分割された後のエントロピーを $E_A(S)$ とする。ここで、我々は $E(S)-E_A(S)$ で計算される利得が最小になる属性を選択することにする。これは、最小利得の $E_A(S)$ は最大のエントロピーであり、その属性値を変更することで効率的に分割を縮小できると考えられるからである。また、決定木生成過程において選ばれなかった属性についても一度はエントロピーを計算するので、そのリストを持つことにする。そうすることで、属性選択のオーバーヘッドを少なくすることが可能である。

属性値の書き換えにより決定木の縮小が期待できるが、クラスに矛盾が起きることがある。$S \subseteq T$ である S の中の属性値を変更したあと、T の中にそれぞれクラスが異なる重複行 $t:c_1, t:c_2$ があったとする。このとき、c_1 の祖先が c_2 であれば無矛盾であるとし、$t:c_1$ を $t:c_2$ に書き換える。また、c_1, c_2 が互いに素の場合には、c_1, c_2 の両方に所属する要素がないので、$t:c_1, t:c_2$ は矛盾すると考える。これを解消するために、ここでは、どちらかの行のクラスメンバーシップを動的に書き換える。階層構造は、単一クラス階層を用いているので、c_1, c_2 は最も近い共通祖先 (least common ancestor) c_0 を必ず持つ。このとき、c_0 により近い方 (ここでは c_1) を選び、c_0 に書き換える。こうすることにより、$t:c_1, t:c_2$ は $t:c_0, t:c_2$ に変わり、行 $t:c_1$ はクラス c_1 に所属するのではなく、クラス c_0 に所属することとなる。これは階層構造により c_1 のクラスメンバーシップが c_0 に緩和されたことを意味する。

最下層のクラス c_1 の比率 n_1/k は $(n_1-1)/k$ に減り、c_1 の上位クラス c_0 の比率は $(n_0+1)/k$ に増える。ここで、n_0, n_1 は T 中における c_0, c_1 の要素数$\Gamma(c_0), \Gamma(c_1)$ であり、k は T 全体の要素数を意味している。c_1 の情報量 $(n_1/k)\log_2(n_1/k)$ は$((n_1-1)/k)\log_2((n_1-1)/k) = (n_1/k)\log_2(n_1/k)-(1/n)\log(n_1/k)$ に変動し、同様に、c_0 の情報量は$((n_0+1)/k)\log_2((n_0+1)/k) = (n_0/k)\log_2(n_0/k)+(1/k)\log(n_0/k)$ に変動する。従って $(1/k)(\log_2 n_0 - \log_2 n_1)$ だけエントロピーの変動が起こったことになる。もし、c_i から c_j への書き換えが、v_{ij} 個存在したら、情報量は $(1/k)\Sigma_{i,j}v_{ij}\times(\log_2 n_j - \log_2 n_i)$ だけ減少する。

Relaxation of Decision Trees
Mikihiro MORI[1], Takao MIURA[2] and Isamu SHIOYA[1]
[1] SANNO College, Dept.of Management and Informatics, Kamikasuya 1573, Isehara, Kanagawa, JAPAN
[2] Hosei University, Dept.of Elec. and Elec. Eng., Kajino-cho 3-7-2, Koganei, Tokyo, JAPAN

3 エントロピーの保存

これまでは、より簡単な決定木を得るために T を緩和する方法について議論してきた。しかし、急激なエントロピーの減少は避ける必要がある。

属性 A について緩和したあと、決定木全体のエントロピー $E(T)$ からみると、$(1/k)\Sigma_{i,j}v_{ij} \times (\log_2 n_j - \log_2 n_i)$ の情報を失ったことになる。ここで、**全体閾率** β $(0.0 \le \beta \le 1.0)$ を導入し、$E(T)$ からのエントロピー変動許容範囲を $|(1/k)\Sigma_{i,j}v_{ij} \times (\log_2 n_j - \log_2 n_i)|/|E(T)| \le \beta$ と定義する。もし、エントロピーの変動がこの閾値を越えた場合、決定木の生成は失敗したこととし、そこで中止する。

次に、決定木のノードごとのエントロピー変動について見てみる。属性 A を選択した場合の情報量変化については $E_A(S)$ として定義したが、ここでは $E_B(S)$ を考える。属性 B は、自分より上のノードで $E_A(S)$ として選ばれていない属性で緩和されていない。もしくは、$E_B(S)$ は属性選択の時に計算されているものとする。ここで、クラスメンバーシップを変える度に B の属性値 $(B = "b_w")$ は再定義され、エントロピーの変動が起こる可能性があるので、クラスが c_i から c_j に変更された行の各 b_w について $v_{i,j}^w$ を数える必要が出てくる。このときのエントロピーの減少は、$\Sigma_{i,j}(v_{i,j}^w/m_w) \times (\log_2 n_j^w - \log_2 n_i^w)$ で表される。ここで、n_j^w は属性値が $B = "b_w"$ である集合の中でクラスが c_j の行の数とし、m_w はその要素数とする。

結果として、最適化されている C4.5 の決定木と比べ、どの程度変化したものになるのかという許容範囲を考えることができる。このために、**選択閾率** γ $(0.0 \le \delta \le 1.0)$ を与え、$E_B(S)$ の変化率が γ の範囲内になるようにする。もし、γ の範囲を超えてしまったら、そのノードまで戻り、属性選択の段階から再起的に決定木生成をやり直す。

属性値の書き換えによるクラスメンバーシップの緩和に関わっていない属性に関しては、エントロピーの変動が起こらず、再計算処理を必要としないので、実行時間を軽減することができる。

4 新しい決定木アルゴリズム

我々の新しい決定木 (*new decision tree*) は以下のアルゴリズム $\mathtt{NDT}(S, AT, AL)$ で表される。ただし、S は行の集合、AT は属性の集合、AL は状態の集合とする。これまで三つの閾値、α (枝刈り率)、β (全体閾率)、γ (選択閾率)、を定義してきた。また、入力として、属性 $A_1, .., A_n$ の集合であるテーブル T が与えられるので、新しいアルゴリズムの初期値は、$\mathtt{NDT}(T, \{A_1, .., A_n\}, \phi)$ であるとする。最終的に決定木生成が成功した場合、木の状態、つまり、クラスの分類に対応したパス $\alpha_1(d_1), ..., \alpha_k(d_k)$ を得る。

(1) もし、S の全ての行が同じクラス d に所属していれば、生成の成功として $AL(d)$ を返す。

(2) さもなければ、まだ選ばれていない全ての属性に関して $E_A(S)$ を計算する。

(3) もし、木の一部が 枝刈り率 α を越えたら、属性値を書き換え、クラスメンバーシップを緩和する。もし、エントロピー変動が全体閾率 β を越えたら、決定木生成失敗として処理を中断する。もし、エントロピー変動が選択閾率 γ を越えたら、バックトラックで属性選択の段階まで戻る。

(4) 各 S_j に関して再帰的に $\mathtt{NDT}(S_j, AT - \{A\}, AL \cup \{A = "a_j"\})$ を行う。

(5) 全ての結果を $AL_1(d_1) \vee .. \vee AL_w(d_w)$ として集める。

ここに載せたアルゴリズムは簡略化したものであるので、詳細は参考文献を参照のこと。

EXAMPLE

ここで、NDT によって決定木を生成してみる。扱うデータは、C4.5 の説明でも使われていた‘レース開催中止’のデータに多少手を加えたものである。まず、次のように **ISA** 階層を定義する。

属性 ***Weather*** は、属性値 ***Fine, Cloudy, Rainy*** を持つ。そして、***Fine, Cloudy*** は ***NotWet*** に、***Rainy*** は ***Wet*** に所属し、***NotWet, Wet*** は ***DontCareWeather*** クラスに所属する。これらは**ISA**を用いると、***Fine*** **ISA** ***NotWet,..., Wet*** **ISA** ***DontCareWeather*** と表現できる。

属性 ***Temperature*** は、属性値 ***VeryHigh, High, Medium, Low, VeryLow*** を持つ。そして階層構造を **ISA** で表すと、***VeryHigh*** **ISA** ***Hot, High*** **ISA** ***Hot, Medium*** **ISA** ***Warm, Low*** **ISA** ***Cold, VeryLow*** **ISA** ***Cold*** となり、次に ***Warm*** **ISA** ***Confortable, Cold*** **ISA** ***Confortable, Hot*** **ISA** ***NotGood*** となり、最後に ***Confortable*** **ISA** ***DontCareTemp, NotGood*** **ISA** ***DonCareTemp*** となる。

属性 ***WindForce*** は、属性値 ***VeryWindy, Windy, Breeze, Windless*** を持つ。そして階層構造を **ISA** で表すと、***VeryWindy*** **ISA** ***Wind, Windy*** **ISA** ***Wind, Breeze*** **ISA** ***NoWind, Windless*** **ISA** ***NoWind*** となり、最後に ***Wind*** **ISA** ***DontCareWin, NoWind*** **ISA** ***DontCareWin*** となる。

最後に 目的であるクラスは ***Held, Half, No*** を持ち、***Half*** **ISA** ***Held*** であるとする。これは、***Half***(半日開催) が ***Held***(開催) の特別な場合であることを表す。

また、閾値は $\alpha = 1.0, \beta = 0.15, \gamma = 0.15$ とする。このとき、α により高さの閾値は 3 となる。最初のエントロピーは 1.265982 である。1つ目の分割として、最小エントロピー 0.637596 の属性 ***Temperature*** が選ばれる。そして、***Temperature = High*** において、NDT はエントロピー 0.0 の ***WindForce*** を選択し、2つの枝、***WindForce = VeryWindy***(***Half*** クラス)、***WindForce = Breeze***(***Held*** クラス) を生成する。また、***Temperature = Low*** において、NDT はエントロピー 0.452846 の ***WindForce*** を選択し、3 つの枝、***WindForce = Breeze***(ここでは次の分割として属性 ***Weather*** が選ばれるが、重複行が発生し、***Half*** クラスが ***Held*** クラスに緩和される)、***WindForce = Windless***(***Held*** クラス)、***WindForce = Windy***、を生成する。この ***WindForce = Windy*** の次の分割では、属性 ***Weather*** が選択される。***Weather*** の属性値 ***Cloudy*** と ***Rainy*** は、***DontCare*** に統一され、クラスメンバーシップは ***DontCareRace*** に緩和される。そして、処理は閾値により許容範囲を保たれたまま終了し、最終的な決定木は次のようになる:

(Temperature)		
VeryHigh		*No*
Medium		*Held*
VeryLow		*No*
High	*(WindForce)*	
	VeryWindy	*Half*
	Breeze	*Held*
Low	*(WindForce)*	
	Windy	*DontCareRace*
	Breeze	*Held*
	Windless	*Held*

ここでは触れられなかったが、C4.5 の初期エントロピーは1.49261 で、NDT の決定木よりも複雑なものが生成された。これに対し、NDT では定義の変更により、初期エントロピーは1.265982 となり、C4.5 より正確で簡単な決定木が生成された。□

5 むすび

この研究では、正確で簡単な決定木を得るために、新しい分類法を提案してきた。我々の新しい決定木生成アルゴリズムは、データベースの背景知識を用い、属性値とクラスメンバーシップを緩和する。このとき、エントロピーを保存することにより決定木の質を一定に保つ。アルゴリズムの処理時間は、最悪の場合、多項式時間になるが、バックトラックは可能な限り回避され、一度計算した結果は再利用され、重複計算は注意深く避けられる。

参考文献

[1] Miura T., Shioya I. and Mori M.: Making Decision Trees More Accurate By Loosing Information (to appear)