決定木によるデータマイニングの比較

Ｔ０１Ｋ０５０Ｆ　鈴木　崇広

1　はじめに

データマイニングとは近年広まってきた言葉であり, 厳密に定義がなされている言葉ではない. 色々な文献の見解をまとめると次のようにいえる. データマイニングとはデータの中から価値のある情報を見つけだすことである. 決定木を使ったデータマイニングアルゴリズムはいくつかあるが, どのアルゴリズムが一番優れているかという議論はされていない. そこで決定木を使ったアルゴリズムについての比較と考察を行う.

2　決定木を用いたデータマイニング

2.1　決定木を用いたデータの扱い

決定木とはデータをもとに枝をたどっていくと適切なターミナルノードにたどり着ける木である. この決定木を生成する際のデータは次のように扱う. 各データセットは複数の属性からなり目的属性と予測属性からなる. 各データセットは１つのクラスに属し, それは目的属性によって表される. 生成された木のターミナルノードには予想されるクラスの値の情報がある.
データのセットにはトレーニングセットとテストセットがある. トレーニングセットはデータマイニングアルゴリズムが木（予測規則）を生成するために利用し, テストセットで木（その規則）を使い目的属性を予測する.

2.2　決定木の生成アルゴリズム

2.2.1　ID3

ID3 について説明する. 決定木では分岐の基準をどうするかが最大の重点となるが,ID3 はエントロピー（情報量）を用いて木を分岐させる条件を決定する. あるデータ集合の事象 X に関する「あいまいさ」は以下の式で定義されるエントロピー Info で測ることができる.

$$Info(X) = -\sum_{j=1}^{k} \{p(j|t) \log_k p(j|t)\}$$

$p(j|t)$: ノード t 内の k 種類あるクラス $a_1, \ldots, a_k$ のうちクラス $a_j (1 \leq j \leq k)$ の出現確率.
ただし $a_i \cap a_j = \emptyset (i \neq j)$

分岐を決定する際上式を用いてまず分岐前のエントロピー $Info_p$ を計算する. 次にある属性がある値であるかどうかで分岐（当然子ノードは２つになる）させたと仮定したときのエントロピー $Info_l$ を計算する. この $Info_l$ を各属性の各カテゴリ変数の全てを計算する. そして最大の $Info_p - Info_l$ となるものを分岐の条件として選ぶ.
この $Info_p - Info_l$ は Gain（相互情報量）と呼ばれている. つまり親ノードのエントロピーと各子ノードの合計のエントロピーとの差である.

2.2.2　C4.5(C5.0)

C4.5 は ID3 をもとに改良されたプログラムである.ID3 と同様にエントロピーを用いて計算するがある属性の値で分岐（子ノードは属性がとる値だけできる）させたと仮定したときのエントロピーを (前述の式の)log の底を２で計算し,Split Info(X) でその Gain(X) を割り規格化を行う. これを Gain に対して Gain 比とよぶ. 式は次のようになる.

$$Gain\ ratio(X) = \frac{Gain(X)}{Split\ Info(X)}$$

$$Split\ Info(X) = \sum_{j}^{C} \frac{N(t_j)}{N(t)} \times \log_2 \frac{1}{\frac{N(t_i)}{N(t)}}$$

$N(t)$: ノード t 内のデータ数
$N(t_j)$: ノード t 内のクラス j をもつデータ数
C : クラスのカテゴリ数
この Gain 比が最大になるものを分岐の条件として選ぶ.

2.2.3　CART

CART は ID3 と同様に２分木を生成するアルゴリズムである.Gini Index という不純度を表す指標を用いて分岐を行う.Gini Index は次の式で表される.

$$Gini\ Index(t) = 1 - \sum_{j=1}^{k} p^2(j|t)$$

$p(j|t)$: ノード t 内のクラス j の割合

CART はできる限りの最高の Gini Index を得ようとする.ID3 と同様に二分岐であるが ID3 のように属性のとりうる値を１つずつ調べていくだけでなく, とりうる値の組み合わせも考慮していく. 例えばある属性のとりうる値が A,B,C,D だとすると
(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 Gini Index を計算することになる. つまり (A|...) という組み合わせが現在のノードでの分岐に選ばれない限り A は次のノードでも分岐の判断に使用することができる.
分岐の手順は ID3 とほぼ同様で Gini Index を計算後親ノードと子ノードの Gini Index の差をが大きいものを分岐の条件として選ぶ.

2.2.4　複数分岐の ID3

今回この３つのプログラムを全て作成したが, それに加えそれぞれのアルゴリズムを比較するために ID3 で複数分岐を行うプログラムを作成した (以後 ID3-p と呼ぶ). 二分岐と複数分岐でどのように異なるのか. また split Info の規格によって違いが生じるのかなどを目的に比較対象として用意したアルゴリズムである. ゲインを用いて複数分岐の木を生成する.

アルゴリズム	分岐数	分岐基準	調べる分岐数
ID3	2	Gain	$\left(\sum_{i=1}^{\text{属性数}} n\right)$ × (属性数)
C4.5	属性がとる値の数	Gain 比	属性数
CART	2	Gini Index	$(\sum_{i=1}^{\text{属性数}} {}_nC_1 + {}_nC_2 + \ldots + {}_{n-1}C_{\frac{n}{2}-1}[+\frac{1}{2}{}_nC_{\frac{n}{2}}])$ ×(属性数) [ ] は n が偶数のとき
ID3-plural	属性がとる値の数	Gain	属性数

n:属性 i がとる値の数

表 1: 各アルゴリズムの違い

2.3 枝刈り

決定木を構築しただけでは有効な規則を得ることはできない. なぜなら予測属性が全く同じでも目的属性が違う場合があるかもしれない.
そのような場合は決定木を生成する際に予測をどのクラスにするのかわからなくなる. しかし実際のデータはこのようなケースが多いのは当然だろう. どちらも正しいデータかもしれないしどちらかのデータが間違っている可能性もある. 間違ったデータを使用して決定木を生成すると当然予測の正解率は悪くなる. この間違ったデータは他と比べて間違いであると判断するしかない. そして間違っている部分木を枝刈りする.
枝刈りとはトレーニングデータによる過学習を補正するためにあまり使われていない（間違いの可能性がある）部分木を取り去る（刈る）ことである.

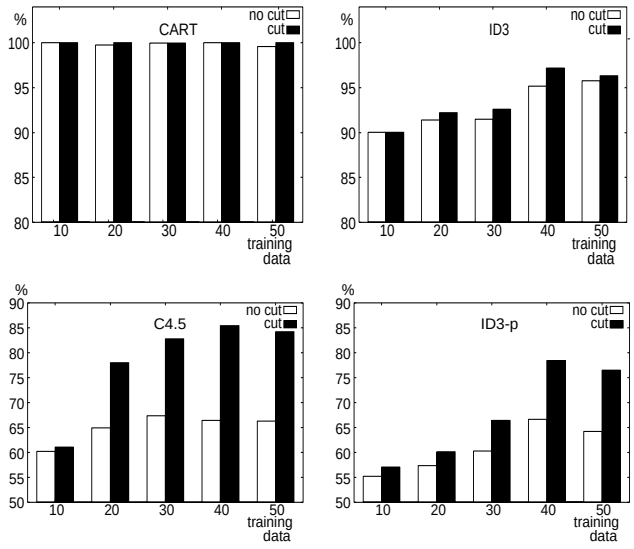

図 1: それぞれのアルゴリズムの正解率

3 各アルゴリズムの実験

作成したプログラムで扱うデータはクレジットカード適用に関するデータ群で目的属性 1 つを含む 16 属性. 目的属性は 2 値で yes,no どちらかの値をとる. トレーニングデータをランダムに 10 個〜70 個選んで使い, テストデータを６７２個で行った. 結果は次の図 1 のようになった. 白いグラフが枝刈りをしない木で実験した場合の正解率であり, 黒い方は枝刈りをした場合の正解率である.

4 考察

1. ID3 と ID3-p と C4.5 はトレーニングデータが増えるにつれて正解率が良くなるが, データ数 40 の時点で正解率の上昇が止まる.
2. ID3 と ID3-p と C4.5 についてはいずれも枝刈りをするとさらに正解率が向上している.
3. CART の正解率が突出してよい.

1. については, トレーニングデータを利用してより正確な木を生成していると考えられる. データ数 40 で正解率ののびが止まるのはある程度の学習が行われ正確な木ができたからである.

2. については, 枝刈りをしたことにより正解率が向上していることよりオーバーフィッティングを避けていると考えられる.C4.5 については枝刈りをすることによりかなりの正解率の上昇が見られる.ID3-p では行っていない Split Info の規格化の効果と考えられる.

3. については,CART はデータ数 10 の時点で 100 %の正解率を得ている. これは CART の全ての組み合わせを考えるアルゴリズムによる. 計算が終了する保証があれば CART は最高水準の正解率を得られるだろう.
しかし CART の場合は計算時間が他の 3 つのアルゴリズムに比べてかなりかかってしまう. またデータ数が増えた場合ノード数が飛躍的に増えメモリ不足で計算できなかったり, 相当の時間を要することが考えられる.

以上のことより CART は計算できる範囲のデータ数で, 時間を気にしないならば最も良いアルゴリズムである.C4.5 は短い計算時間である程度よいの正解率を導けるアルゴリズムであることがわかる.ID3 は CART ほどの正解率は望めないが C4.5 よりはよい正解率を得られることがわかる.

参考文献

[1] Alex A.Freitas：「Data Mining and Knowledge Discovery with Evolutionary Algorithms」p13-p43, Springer, 2002.

[2] マイケル J.A. ベリー/ゴードン・リノフ:「データマイニング手法」p157-p210, KAIBUNDO, 1999.

[3] 福田剛志/森本康彦/徳山豪:「データマイニング」p94-p130, 共立出版, 2001.

[4] 山口和範/高橋淳一/竹内光悦:「よくわかる多変量解析の基本と仕組み」p144-p168, 秀和システム, 2004.

[5] SGI データセット: http://www.sgi.com/tech/mlc/db/

[6] データマイニングの宝箱: http://www5.ocn.ne.jp/ shinya91/index.html