2010年度「知能情報学実験３」プログラミング課題：決定木による事例解析–基礎編

諸注意

本資料における演習問題１と２をテーマ開始までに解いてください。1回目でチェックする予定！
また、本テーマの実施に当たって「データ構造とアルゴリズムの教科書」（C言語によるアルゴリズムと事例構造、著者： 柴田望洋、辻亮介←決定木の実装に関して第10章：木構造が参考になる）及びその他の「C言語の参考書」、並びに「関数電卓」を持参してください。実装はLinuxを使用するので、必要がないファイルを削除し、ホームディレクトリーの容量を確保しておいてください。

概要

表１ 事例集合の例

属性 X	属性 Y	クラス
0.125	0.875	0
0.875	0.875	1
0.625	0.875	0
0.375	0.625	1
0.125	0.375	0
0.875	0.625	1
0.625	0.375	0
0.375	0.125	1
0.125	0.125	0
0.875	0.125	1

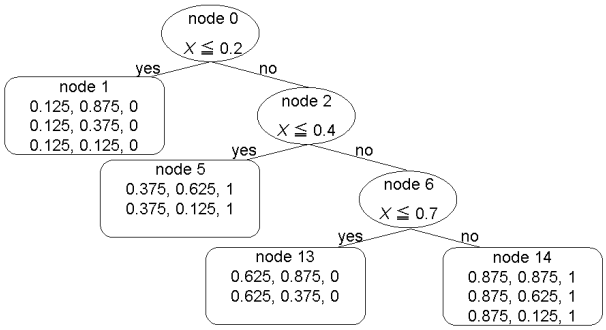

図１ ２分決定木の例

決定木 (Decision Tree) とは、表１にあるような事例項目間の関係を木構造で表現する分析手法である。事例項目には属性項目及びクラス項目がある。決定木は図１にあるようにノードとリンクから構成され、各ノードには分類する属性、ノードとその下位ノードを結ぶリンクには属性値がそれぞれ対応付けされている。下位ノードは上位ノードからのリンク属性値により分類された事例集合を表現する。

決定木を作成する方針によって、分岐が３つある木や分岐数が色々ある混合木というものもあるが、本テーマでは、最も基本となる、ノードを２分岐してゆく**２分決定木（Binary Decision Tree）**を対象とする。

1　２分決定木の作成

1.1　アルゴリズム

C を表１のように属性 $A_1, A_2, \ldots, A_m$ 及びクラスからなる複数の事例をもつ学習用の事例集合または分岐対象ノードが表現する事例集合とする。以後、クラスは「0」と「1」の二つ、属性は連続値をもつと仮定する。C に対して、属性 A の属性値が a 以下となる C の部分集合を $C_{A \leq a}$ とし、$C_{A>a}$

を $C - C_{A \leq a}$ とする。

2 分決定木作成アルゴリズム $BDT(C)$ は以下のように出発点（**ルートノード**）からノードを分岐していく。

$BDT(C)$

- 属性 A とその属性値 a を選択
- C を 2 つの部分集合 $C_{A \leq a}$ と $C_{A>a}$ に分ける
- If $C_{A \leq a}$ におけるクラスの数 $\quad > \quad 1$, then $BDT(C_{A \leq a})$
- If $C_{A>a}$ におけるクラスの数 $\quad > \quad 1$, then $BDT(C_{A>a})$

ここで、分岐が止まったノード (図 1 中の四角い円) を**クラスノード**、それ以外のノード (図 1 中の丸い円) を**判別ノード**と呼ぶ。

各判別ノードでどの属性とその属性値を選択するかが重要である。以下は主な選択基準について述べる。

- 利得　（information gain）
- 利得比　（gain ratio）
- GINI　（GINI index）

利得 [1] と利得比 [2] は Quinlan によって導入された、情報量に基づく基準で、GINI[3] は Breiman によって導入された、非純粋性に基づくものである。

ここで、いくつかの定義をしておく。C 中の事例のクラスが「0」となる確率を p、$C_{A \leq a}$ 中の事例のクラスが「0」となる確率を q、C 中の事例で属性 A の属性値が a 以下となる確率を r とする。また、C^0 で、クラスが「0」となる C の部分集合とする。C の事例数を $|C|$ で表すとすると、これらの確率は次のように求められる。

$$\frac{|C^0|}{|C|} = p, \quad \frac{|C^0_{A \leq a}|}{|C_{A \leq a}|} = q, \quad \frac{|C_{A \leq a}|}{|C|} = r$$

なお、以後 log の底はすべて 2 とする。

1.2　利得と利得比

まず、C の平均情報量 $H(C)$ を以下の式で定義する。

$$H(C) = -p \log(p) - (1-p) \log(1-p)$$

属性 A とその属性値 a の C における平均情報量 $H(A, a)$ を以下の式で定義する。

$$H(A, a) = rH(C_{A \leq a}) + (1-r)H(C_{A>a})$$

属性 A とその属性値 a の**利得**は以下のようになる。

$$gain(A, a) = H(C) - H(A, a)$$

利得が最も大きい属性とその属性値を選択する基準のことを**利得基準**という。これに対して**利得比基準**は以下に定義する**利得比**が最も大きい属性とその属性値を選択するものである。

属性 A とその属性値 a の分割平均情報量 $split(A, a)$ を以下のように定義する。

$$split(A, a) = -r\log(r) - (1-r)\log(1-r)$$

属性 A とその属性値 a の利得比 $gain\_ratio(A, a)$ は以下のようになる。

$$gain\_ratio(A, a) = \frac{gain(A, a)}{split(A, a)}$$

1.3 GINI

以下に定義する **GINI** が最も大きい属性とその属性値を選択する基準のことを **GINI 指標**という。まず、C の非純粋性 (impurity)、$IM(C)$、を以下のように定義する。

$$IM(C) = 1 - p^2 - (1-p)^2$$

属性 A とその属性値 a に対する非純粋性 $IM(A, a)$ を以下のように定義する。

$$IM(A, a) = rIM(C_{A \leq a}) + (1-r)IM(C_{A>a})$$

属性 A とその属性値 a の GINI、$gini(A, a)$、を以下のように定義する。

$$gini(A, a) = IM(C) - IM(A, a)$$

1.4 例題

表 1 の事例を学習用の事例集合 C とし、属性 X とその属性値 0.2 に対する利得 $gain(X, 0.2)$ を求める。

まず、C 中の事例のクラスが「0」となるのは 10 個中 5 個なのでその確率 p は 0.5 である。
よって $H(C)$ は

$$H(C) = -0.5\log(0.5) - (1-0.5)\log(1-0.5) = 1$$

次に C 中の事例で属性 X の属性値が 0.2 以下となるのは 10 個中 3 個なのでその確率 r は 0.3 である。よって $H(X, 0.2)$ は

$$H(X, 0.2) = 0.3H(C_{X \leq 0.2}) + (1-0.3)H(C_{X>0.2})$$

後は $H(C_{X \leq 0.2})$ と $H(C_{X>0.2})$ を求めればよい。
$C_{X \leq 0.2}$ 中の事例のクラスが「0」となるのは 3 個中 3 個なのでその確率 p は 1 である。よって $H(C_{X \leq 0.2})$ の値は 0 になる。
また、$C_{X>0.2}$ 中の事例のクラスが「0」となるのは 7 個中 2 個なのでその確率 p は 0.286 である。よって $H(C_{X>0.2})$ は

$$H(C_{X>0.2}) = -0.286\log(0.286) - (1-0.286)\log(1-0.286) = 0.863$$

これらの値を代入すると

$$H(X, 0.2) = 0.3 * 0 + 0.7 * 0.863 = 0.604$$

よって $gain(X, 0.2)$ は

$$gain(X, 0.2) = H(C) - H(X, 0.2) = 1 - 0.604 = 0.396$$

1.5 演習問題１

表 1 の事例を学習用の事例集合 C とする。属性 X とその属性値 0.4 に対する各選択基準を求めよ。

$gain(X, 0.4) =$ [　　], $gain\_ratio(X, 0.4) =$ [　　], $gini(X, 0.4) =$ [　　]

2 決定木の検証

クラスが未知の事例（以後、未知事例と呼ぶ）は、決定木をルートノードから下位の方向に事例の属性値によって辿って見つかったクラスノードのクラスをもつと推定される。クラスノードのクラスはそのノードに分類された学習事例のクラスの中から多数決で決定される。**認識率**とは、対象とする事例集合のうち正しくクラスが推定できた事例の割合を示す数字である。なお、**誤認識率**は 1–認識率となる。

決定木の作成に用いた学習用の事例集合に対する誤認識率が高いと、この現象は**未学習（under-fitting)** と呼ばれる。決定木における原因は、木の深さで分岐を強制終了する条件を設ける際にその深さの値が少なすぎることや、連続値をもつ属性を量子化する際にその等分割の幅が広すぎることにある。

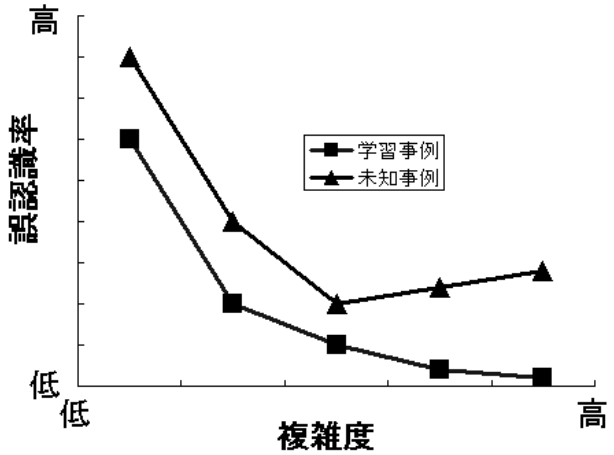

図 2 誤認識率と学習モデルの複雑度との関係

もう一つの問題として、学習は学習用の事例特有の非常に稀な場合まで適合してしまい、新しい事例ないし一般的な事例に適応しなくなり、すなわち、未知事例に対して誤認識の可能性が高いことで

ある。この現象は**過学習（overfitting）**と呼ばれる。決定木における原因は、未学習と反対に、決定木を成長させすぎることや連続値の量子化が細かすぎることにある。

学習事例及び未知事例に対する誤認識率と学習モデルの複雑度との関係は、図 2 のように示すことができる。この図からわかるように、未学習及び過学習の問題を同時に解決して、両誤認識率を低く抑えるには、適切な複雑度をもつ決定木の作成が重要である。この目的を達成するために次章で述べる**枝刈り** (pruning) があるが、その前に、未知事例に対する誤認識率を推定するための検証法について述べる。

2.1 事例が十分ある場合

十分な事例がある場合、よく用いる検証法は事例集合を学習用と検証用のグループに分けることである。事例集合を分ける際には各項目（特にクラス項目）に関して同じような分散を持たせる必要がある。学習用グループを用いて決定木を作成した後、検証用グループに対する誤認識率を未知事例に対する推定値として求める。

2.2 事例が不十分な場合

検証用のグループが十分にとれない場合は、**N-交差検証法（N-hold Cross Validation）**がよく使われる。例えば、N を 10 に設定すると、この方法では事例集合を各項目に関して同じような分散をもつグループに 10 等分した後、最初の 9 つのグループを用いて決定木を作成し、検証用として残った 1 グループに対する誤認識率を求める。次に検証用グループを順々に変えながら、残りの 9 グループを用いて同様に決定木を作成して各回の誤認識率を求める。最後にこれらの誤認識率の平均をとり、この平均値を未知事例に対する推定値として採用する。この N-交差検証法のアルゴリズムを以下にまとめる。

N-hold Cross Validation

- 事例集合 C を、均等に N グループに分割する（すなわち、$C \rightarrow C_1, C_2, \ldots, C_N$）
- $i = N$ とする
- C_i を C から取り除き、$C - C_i$ を学習用グループ、C_i を検証用グループに設定
- $C - C_i$ を用いて作成した決定木に対して C_i を用いて評価を行った後、C_i を C に戻して i を一つ減らす
- この作業を N 回繰り返した後、N 回分の誤認識率の平均をとる。この平均値を未知事例に対する推定値として採用

3　枝刈り

未学習及び過学習の問題を同時に解決するために、いったん決定木を成長させた後に **Occam's razor** の考え方に基づいて枝刈りを行う方法がよく用いられる。Occam's razor とは、事例に適合する仮説のうち、最も単純なものを優先する考え方で、新しい理論の考案からものづくりまで理工学の分野に幅広く応用されている。決定木においては、枝刈りに伴う誤認識率の増加が少なく、多くの下位の判別ノードをもつ判別ノードに対して優先的に枝刈りを行うことである。ここで、ある判別ノードに対して枝刈りを行うとは、そのノードを削除せずそのノードから分岐したすべての下位ノードを１つにまとめることを指す。

枝刈り候補の判別ノードを n、$\Delta error\_rate(tree\ without\ n)$ を n に対して仮に枝刈りを行うことに伴う学習用の事例集合（または学習用グループ）に対する誤認識率の増加（すなわち、枝刈り後の誤認識率 − 枝刈り前の誤認識率)、$size(subtree\ n)$ を n とその下位の判別ノードを合わせたノード数（すなわち、1+下位判別ノード数）とする。代表的な n の**枝刈り基準** $g(n)$[4] は以下のような式で定義する。

$$g(n) = \frac{\Delta error\_rate(tree\ without\ n)}{size(subtree\ n)}$$

3.1　演習問題２

図１の決定木における判別ノードに関する各枝刈りのパラメタを求めよ。

n	$\Delta error\_rate(tree\ without\ n)$	$size(subtree\ n)$	$g(n)$
node 6			
node 2			
node 0			

ヒント：$node6$ に対して枝刈りを行った場合の決定木を図 3 に示す。

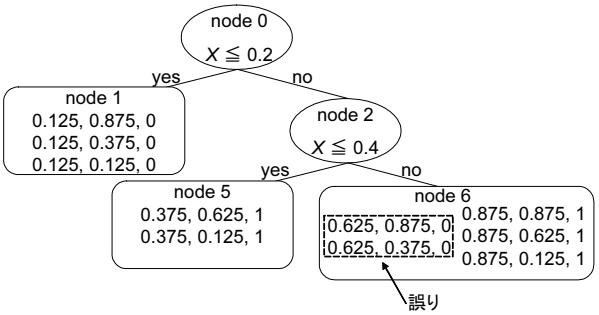

図３ node 6 に対して枝刈りを行った様子

3.2 アルゴリズム

枝刈りは、作成した決定木において枝刈り基準が最も小さい判別ノードから順番に k 回行われるが、枝刈り回数 k をどう設定するかが重要になってきる。以下は k を設定するための N-交差検証法に基づいたアルゴリズムである。

- 事例集合 C を、均等に N グループに分割する（すなわち、$C \rightarrow C_1, C_2, \ldots, C_N$）
- $i = N$ とする
- C_i を C から取り除き、$C - C_i$ を学習用グループ、C_i を検証用グループに設定
- $C - C_i$ を用いて作成した決定木において枝刈り基準が最も小さい判別ノードから順番にルートノードまで枝刈りを行っていく。同時に C_i に対して枝刈り回数ごと（$k = 1, 2, \ldots$）の誤認識率を記録
- C_i を C に戻して i を一つ減らす
- この作業を、N 回繰り返した後、検証用グループに対する N 回分の誤認識率の平均が最も小さい枝刈り回数 $k_{optimal}$ を採用

最後に、すべての事例集合 C を用いて決定木を作成した後、枝刈り基準が最も小さい判別ノードから順番に枝刈りを $k_{optimal}$ 回行っていく。

参考文献

[1] J. R. Quinlan: Induction of decision trees, Machine Learning 1, 81.106, 1986.

[2] J. R. Quinlan: C4.5: Programs for machine learning, Morgan Kaufmann Publishers, 1993.

[3] L. Breiman, J. Friedman, R. Olshen, C. Stone: Classification and regression trees, Wadsworth, 1984.

[4] S. M. Weiss and C. A. Kulikowski: Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems, Morgan Kaufmann Publishers, 1991.