DEWS2004 7-I-01

KL情報量を用いた決定木の刈り込み

高光　知哉† 　三浦　孝夫† 　塩谷　勇††

† 法政大学 工学研究科 電気工学専攻　〒184–8584 東京都小金井市梶野町 3-7-2
†† 産能大学 経営情報学部　〒259–1197 神奈川県伊勢原市上粕屋 1573
E-mail: †{i02r3230,miurat}@k.hosei.ac.jp, ††shioya@mi.sanno.ac.jp

あらまし　本論文では決定木の事前枝刈り法 KLC4 を提案する．すなわち，分類に不必要な属性の枝刈りを行い，決定木の大きさを低減化することができることを示す．ここではノードが持つデータ集合のクラス分布の変化に注目し，属性値間の相関規則と KL 情報量を用いて，簡潔で信頼性の高い決定木を獲得する．本論文で提案する手法を実験し，従来の手法と比較しながらその有効性を評価する．

キーワード　知識発見　データマイニング

Decision Trees using KL Divergence

Tomoya TAKAMITSU†, Takao MIURA†, and Isamu SIOYA††

† Dept.of Elect.& Elect. Engr., HOSEI University　3-7-2, KajinoCho, Koganei, Tokyo, 184–8584 Japan
†† Department of Management and Information Science, SANNO University　1573, Kamikasuya, Isehara city, Kanagawa 259–1197 Japan
E-mail: †{i02r3230,miurat}@k.hosei.ac.jp, ††shioya@mi.sanno.ac.jp

Abstract　In this paper, we propose *pre-pruning* method for decision tree. By this method, we prune attributes unnecessary for classification thus we have much simple, small and reliable trees. Here we put on our attension on class distribution of given datasets, and we extract local association rules for t epurpose of pruning. We comprae our results to conventional techniques, and we show usefulness of our technique.

Key words　Knowledge Discovery , Data Mining

1. 前　書　き

近年，大規模なデータを高速かつ正確に分類するデータマイニング技術に注目が集まっている．特に決定木 (Decision Tree) はこの分野に代表される技術の 1 つであり，木構造形式の分類モデルである [9]．決定木は，その表現形式のわかり易さから分類規則の記述に利用されることが多い．

決定木を生成する手法には，ID3 や C4.5 など，訓練データの頻度分布に従うアルゴリズムが代表的である [2], [9]．これらのアルゴリズムには多くの利点がある．問題領域ごとの知識を必要とせず訓練データのクラス分布のみに従っているため幅広い応用が可能である．また高速に木を生成することができる．

決定木は，大きさと精度の観点から評価される．決定木に含まれるノード数が少ないほど理解しやすく良いと考えられる．分類精度とは，新たに与えられたデータの所属するクラスを正確に予測し，分類できる機能を意味する．一般的に，与えられたデータを訓練データとして決定木を生成しても評価の高い決定木を得ることは困難である場合が多い．これは，属性記述が冗長であったり不足しているなど，データ自体に起因するものと，訓練データを重視し大きな決定木を生成したことに対する過学習 (overfitting) によるものがある．

このような問題点を解決する方法として，生成した決定木を統計的に評価し効果の期待できないノードを削減する事後枝刈り (post-pruning) や，事前に属性値の分布や属性値間の相関性を利用し不要なノードの生成を抑制する事前枝刈り (pre-pruning) が提案されている [2], [10]．事後枝刈りに関する研究はこれまで多く提案され，訓練データを過学習する決定木を刈り込む手法が論じられている．統計的に誤分類率の変化がない場合や複雑すぎる木に対して，最も有害と期待される部分を刈り込む方法がとられている．従って，一般的に効率が悪い．これに対して事前枝刈りでは，属性生成や属性間の相関の処理のために発見的な知識や確率的仮定をとる．しかし，閾値の自由度が高すぎたり，局所的な判断に影響されがちであり，大規模データに対するスケーラビリティやデータ分布の局所的特性の抽出に問題が発生することが多い．

著者らは検定を用いて決定木の内部構造を調査して，パスの適応性や決定木の信頼性を評価する手法を提案した [11]．また，クラス階層とクラスに複数のクラスを持たせる選言クラスを用

いてクラスの意味を曖昧にする手法とパスエントロピを用いて興味深さを評価する手法を提案した [12]．これらは簡潔で信頼できる興味深い決定木を構築する方法を論じるものである．本論文では，属性値間の相関に関する知識を基に属性を局所的に削除する事前枝刈り手法を提案する．本手法では，

(1) 属性値間の相関規則を訓練データをもとにして自動的に抽出する．KL 情報量を用いてクラス分類の分布近似を判定するため，極めて高速に相関規則の検出処理ができる．特に経験的・確率的な知識を必要としない．
(2) 至る所で規則の検出が行えるため，局所的な属性値間の相関性を判定できる．

など，大規模データを扱うための処理手法として優れた特長を備えている．

本論文の構成は次のとおりである．2 章では決定木とその生成について要約し，特徴と問題点を述べる．3 章ではデータ発見分野で盛んに研究されている相関規則とその抽出について述べ，決定木の生成の基礎となるエントロピの性質と相関規則の関連を示す．この性質を用いて 4 章では，決定木生成と事前の属性削減の関連，および生成手法を述べる．5 章では実験によりこの手法の有用性を示し，6 章で関連研究を要約したあと，7 章で結びとする．

2. 決 定 木

この章では決定木とその生成手法を要約する．

2.1 データの分類と決定木

決定木を考察するとき，次の形式のデータ多重集合 T を用いる．

$$T = \left(\begin{array}{cccc} A_1 & ... & A_k : \mathcal{C} \\ \hline a_1^1 & ... & a_n^1 : c_1 \\ .. & ... & .. : .. \\ a_1^n & ... & a_k^n : c_n \end{array} \right) = \left(\begin{array}{c} \mathcal{A} : \mathcal{C} \\ \hline t_1 : c_1 \\ .. \\ t_n : c_n \end{array} \right)$$

各行は一つのオブジェクトを示す．オブジェクトは複数の属性 $\mathcal{A} = \{A_1, A_2, ..., A_k\}$ と一つのクラスカテゴリ $\mathcal{C}$ 上で定義される．各行の値 $< a_1, a_2, ..., a_k >$ は各属性 $A_i \in \mathcal{A}$ 上でその領域値 a_i をとり，当該オブジェクトの特徴を示す．クラス c はオブジェクトが属するクラスを示す．

分類とは，新たに与えられた入力オブジェクトの属性値を調べることによって，それが所属するクラスを予測することをいう．この判定の規則集合を分類器と呼び，上述の様に表現された T を用いて構築される．データ (多重) 集合から分類器を構築することを (分類) 学習という．学習には教師つき学習と教師なし学習がある．教師つき学習は，訓練データと呼ばれるデータ多重集合を用いて分類器を構築する手法であり，決定木生成はこの代表例である．これに対して，教師なし学習は訓練データを用いずに分類器を構築する手法であり，代表的には自己組織化マップ (SOM) などが知られている [3], [6].

決定木は中間ノードと葉ノードからなる木構造形式の分類規則集合である．中間ノードには 1 つの属性が，葉ノードには 1 つのクラスが対応している．決定木はクラスが未知なオブジェクトの属するクラスを予測する．入力データが与えられたとき，根から開始された検査は，中間ノード上の属性値 $A_i =$ "a_i" に従って分岐する．これを繰り返し，葉ノードに達したとき，葉ノード上のクラスを予想する．1 つの葉に至る経路 (パス) は 1 つしかない．各パスは連言属性条件 $A_1 =$ "a_1" $\wedge ... \wedge A_k =$ "a_k" を意味しており，決定木はパスで表現される属性条件の集合を表す．所属するクラスが判明した場合には，検査を途中で打ち切ってよく，条件は必ずしも全ての属性を含むわけではない．

［例 1］ 図 1 に分類で使用されるデータ集合を示す．ここではコンピュータを購入する顧客の特徴を示す．各オブジェクトは属性 Age($\leqq 30, 30-40, 40 <$), Income($High, Medium, Low$), Student(Yes, No), Credit Rating($Excellent, Fair$) とクラスカテゴリ BuysComputer(Yes, NO) で構成される．各オブジェクトは属性値によって客の特徴を表し，クラスによってコンピュータの購入の有無を表す．例えば，Age が $\leqq 30$，Income が $High$，Student が No，Credit Rating が $Fair$ という属性を持つ客はコンピュータ購入者 (BuysComputer=Yes) であることを示す．

図 1 のデータを訓練データとして決定木を構築する．図 2 に図 1 を用いて構築された決定木を示す．ここで長方形は中間ノードを，楕円は葉ノードを表す．

未知オブジェクトが Age が $\leqq 30$，Income が Low，Student が Yes，Credit Rating が $Fair$ であるとき，決定木に従って検査が開始される．木の根が持つ Income の属性値は Low であるから，右の子に進む．その属性 Credit Rating の属性値は $Fair$ であるから，該当枝を進み葉ノードに達する．このクラス No から，入力オブジェクトのあらわす顧客はコンピュータを購入しない (クラス No) と予測する．

Attribute				Class
Age	Income	Student	Credit Rating	Buys Computer
≦30	High	No	Fair	Yes
≦30	High	No	Excellent	Yes
31～40	High	No	Fair	Yes
40<	Medium	No	Fair	Yes
40<	Low	No	Fair	No
40<	Low	Yes	Excellent	Yes
31~40	Low	Yes	Excellent	Yes
≦30	Medium	No	Fair	No
≦30	Low	Yes	Fair	No
31~40	Medium	Yes	Fair	Yes
≦30	Medium	Yes	Excellent	No
31~40	Medium	No	Excellent	Yes
40<	Low	Yes	Excellent	Yes
31~40	Low	Yes	Fair	No

図 1 訓練データ

2.2 決定木の生成

決定木の生成には，クラス分類が予め知られたデータ多重集合 T を訓練データとして利用する．決定木を生成するとき，訓練データを最も効率よく分類できるように各ノードの属性・クラスを選択する．

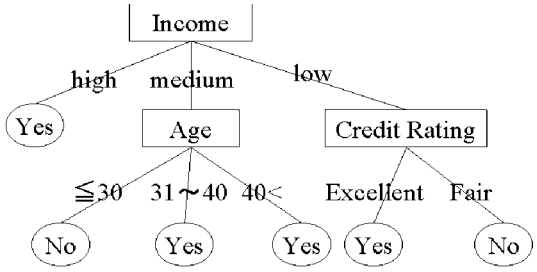

図 2　決定木

決定木の構築は訓練データ T と根ノードから開始する．訓練データのオブジェクトが全て同一クラス c に属するならば，当該ノードをクラス c の葉ノードとする．さもないとき中間ノードとし属性を与える．

属性を選択するためにエントロピ (entropy) の概念を導入する．多重集合 T には n 行のデータが含まれるが，クラス $c_j \in \mathcal{C}$ には n_j 個のデータが属するとき，$p_j = n_j/n$ として $\log(1/p_j) = -\log p_j$ をクラス c_j の T に関する情報量と言う．T に関する (クラス) エントロピ $Ent(T)$ はクラス毎の情報量の期待値を表す．

$$Ent(T) = \Sigma_{j=1}^{h} p_j \log_2(1/p_j) = -\Sigma_{j=1}^{h} p_j \log_2 p_j$$

T を属性 $A \in \mathcal{A}$ 上で分割したとき，各部分多重集合のエントロピの期待値を A に関する相対エントロピ (Relative Entropy) $Ent_A(T)$ という．形式的には，T 上で属性 A の値として生じるデータを $a_1, .., a_w$，$T_1, .., T_w$ を T の空ではない部分多重集合への分割で，各 T_j の要素数を n_j，それぞれが $A = "a_1", ..., A = "a_w"$ を満たすとき，T の A に関する相対エントロピ $Ent_A(T)$ は，各 $Ent(T_j)$ の期待値と定義され，次のようにあらわされる．

$$Ent_A(T) = \Sigma_{j=1}^{w}(n_j/n)Ent(T_j)$$

中間ノードのための属性選択を行うため，T についてエントロピを求め，各属性ごとに相対エントロピを算出する．この差が中間ノードを設けたことによる利得であり，これを最大にする属性 G を選択する．

$$G = \mathtt{MaxArg}_A(Ent(T) - Ent_A(T))$$

属性 A が選択された結果，その属性値の数だけの子ノードが生成され，各枝に属性値 a_j が対応する．各子ノードにはオブジェクトの部分多重集合 T_j が対応する (T_j の各オブジェクトは A 上で値 a_j をとる)．この処理を繰り返しすべてのノードが葉ノードとなったとき，決定木生成を終了する．

[例 2]　図 1 の顧客データ T を訓練データとして決定木を生成する．最初に根ノードを作成する．このノードにはすべての訓練データが対応する．このノードに対応するエントロピは，$Ent(T) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940286$ である．各属性に関して相対エントロピを求めると，

$Ent_{Age}(T) = 0.836393$
$Ent_{Income}(T) = 0.775339$
$Ent_{Student}(T) = 0.924174$
$Ent_{CreditRating}(T) = 0.85001$

となるので，最大利得を与える属性は `Income` であるとわかる．これを用いて T を分割し，その属性値 $High, Medium, Low$ に対応するノードを新たに作成する．これを繰り返して，図 2 の決定木を生成する．

2.3　決定木の評価

一般的に，分類器は評価のためのデータ (評価データ，テストデータ) を用いて評価される．決定木を用いてテストデータを分類し，オブジェクトが持つクラスと決定木の予測するクラスを比較する．評価の方法としては誤分類率が利用される．これはクラスと決定木で分類して得たクラスが不一致であるオブジェクトの割合を言う．誤分類率が低ければ低いほど，その決定木はオブジェクトを正確に分類できる．

3.　相関性とエントロピ

本章では，決定木生成で重要な役割を担うエントロピと，属性値間の相関性との関連を論じる．

3.1　相 関 規 則

項目集合 $\mathcal{I} = \{e_1, ..., e_N\}$ が与えられ，その上の事例集合 $D = \{W_1, ..\}, W_i \subseteq \mathcal{I}$ を考える．D 上の相関規則 (Association Rules) とは $X \Rightarrow Y$ の形の記述をいう．ここで $X, Y \subseteq \mathcal{I}$ かつ $X \cap Y = \emptyset$ である．相関規則は，事例に含まれる項目組にある共起 (co-occurrence) パターンをあらわす．すなわち,「もし，事例中に X の全ての項目が含まれれば，その事例には Y の全ての項目も含まれることが多い」という内容を意味する．

規則 $X \Rightarrow Y$ が支持度 (support) s を持つとは，D の $s\%$ の事例が各々に $X \cup Y$ を含むときをいう．規則 $X \Rightarrow Y$ が確信度 (confidence) c を持つとは，D の事例の各々で X を含めば Y も含むものが $c\%$ あるときを言う．予め与えた s, c に対して，相関規則の支持度と確信度が共に s, c 以上であるとき，規則は D で有効であるという．支持度・確信度の最小値を小さく設定すれば，対象となる相関規則候補数が多くなり，候補規則を多く抽出できるが，反面，意味のない規則を生成することになり，探索コストも増加するというトレードオフがある．

大量のデータから相関規則を探索的に抽出するには，従来手法では多くの計算時間を要する．Agrawal らは相関規則を高速に分析・抽出する Apriori アルゴリズムを提案した [1]．このアルゴリズムでは，相関規則の探索中に支持度と確信度を用いて相関規則の候補を選び出す．相関規則の支持度が高いほど，規則が適用できる可能性が多くの事例中にあり，確信度が高いほど信頼性の高い結論を導くことができる．Apriori アルゴリズムでは，これら支持度と確信度について，最小値を充たさない相関規則候補を相関性が低いものとして逐次的に評価対象から除外する．ここでは項目集合の要素数順に候補を生成し，妥当性を検証しながら大きな候補集合を求める．当初，ほとんど全ての項目が候補になるが，急速に収束するため，結果的に探索空間が縮小することが知られている [3].

3.2 エントロピと相関規則

エントロピ $Ent(T)$ の性質を分析するために，T のクラス分布 $dist(T)$ を次のように定義する．

$$dist(T) =< n_1/n, ..., n_h/n >$$

ただし T の要素数を n, T の要素でクラス c_j に属するものの数を n_j とする $(n = \Sigma n_j)$．属性 A 上の値を a とするとき $T_{A(a)}$ を次のように定義する．

$$T_{A(a)} = \{t \in T | t[A] = "a"\}$$

ここで $t[A] = "a"$ は属性 A 上で値 a となるデータを意味する．このとき次の性質が成り立つ．

[定理 1] $dist(T) = dist(T_{A(a)})$ であるとき，$Ent(T) = Ent(T_{A(a)})$.

(証明)

$T_{A(a)}$ の要素数を m, そのうちクラス c_j に属する要素の数を m_j とすれば，$dist(T) =< n_1/n, ..., n_h/n >$, $dist(T_{A(a)}) =< m_1/m, ..., m_h/m >$ である．$n_j/n = m_j/m$ であるから $m_j/n_j = m/n$ が成り立つ $(j = 1, .., h).m/n = s$ とおけば $(0.0 \leqq s \leqq 1.0)$，$m = s \times n, m_j = s \times n_j$ であるから，次式を得る．

$$\begin{aligned} Ent(T_{A(a)}) &= \log m - (1/m)\Sigma m_j \log m_j \\ &= (\log s + \log n) \\ &\quad -(1/s)(1/n)\Sigma(sn_j)(\log s + \log n_j) \\ &= \log n - (1/n)\Sigma n_j \log n_j \\ &= Ent(T) \end{aligned}$$

□

この定理の主張することは，T と子ノードのクラス分布が同じときエントロピも同じであることにある．

T の各行 $< a_1, .., a_k >$ を，項目集合 $\{A_j(a_i^j) | i, j = 1, ..\}$ 上の事例 $\{A_1(a_1), ..., A_k(a_k)\}$ に対応させれば，T から相関規則を抽出することができる．

[定理 2] (1) $T_{A_1(a_1)}$ 上で相関規則 $A_1(a_1) \Rightarrow A_2(a_2)$ が成り立ち，支持度・確信度がそれぞれ s, c のとき $s = c$ である．以下では $A_1(a_1) \Rightarrow A_2(a_2)[s]$ と表現する．

(2) $T_{A_1(a_1)}$ 上で相関規則 $A_1(a_1) \Rightarrow A_2(a_2)$ および $A_1(a_1) \Rightarrow A_3(a_3)$ が成り立ち，支持度をそれぞれ s_2, s_3 とする．このとき $A_1(a_1) \Rightarrow "A_2(a_2)$ または $A_3(a_3)"$ の支持度は $s_2 + s_3$ である．

(3) $T_{A_1(a_1)}$ 上で相関規則 $A_1(a_1) \Rightarrow A_2(a_2)$ および $A_1(a_1) \Rightarrow A_2(a_2')$ が成り立ち，支持度をそれぞれ s_2, s_2' とする $(a_2 \neq a_2')$．このとき $A_1(a_1) \Rightarrow "A_2(a_2)$ または $A_2(a_2')"$ の支持度は $s_2 + s_2'$ である．

(証明)

(1) $T_{A_1(a_1)}$ が要素数 m であるとき，どの行も A_1 上では値 a_1 をとるので，属性 A_2 上 a_2 となる $T_{A_1(a_1)}$ の要素数 m_2 に対して，この規則の確信度 s は m_2/m となる．またこれは，属性 A_2 上 a_2 かつ A_1 上 a_1 となる $T_{A_1(a_1)}$ の要素数と一致するので，支持度 c は m_2/m であり，これは s と等しい．

(2,3) 同様に要素数は $m_2 + m_3, m_2 + m_2'$ となることがわかる．

□

この定理の主張することは，相関規則を抽出する手間が不要になることにある．すなわち，相関規則が得られるならば支持度と確信度が一致することから，$T_{A_1(a_1)}$ での $A_1(a_1) \Rightarrow A_2(a_2)[s]$ の抽出は，A_2 上で a_2 である $T_{A_1(a_1)}$ のデータを数えることと同じである．行データを事例形式に変更し Apriori 計算を行う必要はない．

[定理 3] T の部分多重集合 $E = T_{A_1(a_1)}, E_{A_2(a_2)}$ を考える．

(1) $dist(E) = dist(E_{A_2(a_2)})$ であり，E 上で相関規則 $A_1(a_1) \Rightarrow A_2(a_2)[s]$ が成り立てば，$0 \leqq Ent(E) - Ent_{A_2}(E) \leqq (1 - s)Ent(E)$ が成り立つ．

(2) $dist(E) = dist(E_{A_2(a_2)}) = dist(E_{A_2(a_2')})$ とする．E 上で相関規則 $A_1(a_1) \Rightarrow A_2(a_2)[s]$ および $A_1(a_1) \Rightarrow A_2(a_2')[s']$ が成り立てば，$0 \leqq Ent(E) - Ent_{A_2}(E) \leqq (1 - (s + s'))Ent(E)$ が成り立つ．

(証明)

(1) $b_1 = a_2$ とする．相対エントロピの定義から

$$\begin{aligned} Ent_{A_2}(E) &= \Sigma_{j=1}(\alpha_j/\alpha)Ent(E_{A_2(b_j)}) \\ &= s \times Ent(E_{A_2(b_1)}) \\ &\quad +\Sigma_{j=2}(\alpha_j/\alpha)Ent(E_{A_2(b_j)}) \\ &\leq s \times Ent(E_{A_2(b_1)}) \\ &= s \times Ent(E) \end{aligned}$$

ここで α, α_j はそれぞれ $E, E_{A_2(b_j)}$ の要素数をあらわす．これより $0 \leqq Ent(E) - Ent_{A_2}(E) \leqq (1 - s)Ent(E)$ を得る．

(2) 同様に証明できる．

□

この定理では情報利得の計算を行っている．クラス分布が同じで相関規則が抽出できるとき，支持度 s が 1.0 に近いほど情報利得は少なくなることをあらわしている．

3.3 KL 情報量と相関規則

2 つの分布 $p =< v_1, .., v_h >, q =< u_1, .., u_h >$ に対して $(\Sigma v_j = \Sigma u_j = 1.0)$，その KL 情報量 (Kullback Leibler Divergence) $KL(p||q)$ とは

$$KL(p||q) = \Sigma_j v_j \log_2(v_j/u_j)$$

で定義される．この KL 情報量は常に 0 以上であり，分布 p が分布 q と一致するときにだけゼロとなるという性質を持つ．これは 2 つの分布の当てはまりのよさを与える尺度となることを意味し，あてはまりが良いほど小さい値を持つ．

本稿では，クラス分布 p, q を比較するために KL 情報量を用いる．すなわち，$KL(p||q)$ が小さいほど分布が似ていることを示すことから，中間ノードの属性から (属性値ごとに) 分割されたデータ集合のクラス分布を調べ，元のクラス分布と比較できる．このような属性値を多く持つ属性は，定理 3 より情報利得が小さいため，分類に効果的ではない．一般に，定理 1 および定理 3 において，極めて近い分布を選べば，同等の結果が得られることから，以下では属性値間の相関性とエントロピの関連を KL 情報量の意味で近似して用いる．

4. 属性値間の相関規則を用いた枝刈り

この章では，データの分類に効果的ではない属性を予め削除することによって，過学習を防ぎ，簡潔で信頼性の高い決定木の生成を行う手法を述べる．

基本的なアイデアは，決定木生成中のクラス分布の変化に注目することにある．中間ノードの属性を選択するために算出される情報利得は，クラス分布の変化に依存して決定される．この変化の大きい属性値を多く持つ属性はノード属性に選ばれやすい．変化の小さい属性は決定木生成に有用でないので，予め削除しておくことができる．

ノードが持つデータの多重集合 T のクラス分布 $dist(T)$ が

$$dist(T) = \langle n_1/n, n_2/n, ..., n_h/n \rangle$$

とする．ここで n は T の要素数，n_j はクラス c_j に属する T の要素数を示す．T の属性 $A \in \mathcal{A}$ 上で生じる属性値を $a_1, .., a_w$，$T_{A(a_i)}$ を T の多重部分集合で，A 上 a_i であるものだけを含むとし，$T_{A(a_i)}$ のクラス分布 $dist(T_{A(a_i)})$ を考える．

2 つの実数パラメタ α,β を $0 \leqq \alpha, 0 \leqq \beta \leqq 1.0$ とする．α は分布の近さを定義するパラメタ，β は未分類オブジェクトの許容個数の割合を表すパラメタである．$dist(T)$ と $dist(T_{A(a_i)})$ の KL 情報量を求め，

$$KL(dist(T)||dist(T_{A(a_i)}) \leqq \alpha$$

のとき 2 つの分布は類似していると言う．このとき定理 1，2，3 の適用条件が得られたことになる．

クラス分布の類似する T と $T_{A(a_i)}$ に対して，定理 3 より，$T_{A(a_i)}$ の T に対する要素数の割合が β を超えるときには，属性 A はノード属性になりえない．この結果，考察対象から A を除外してよい．

本手法による決定木生成の属性選択プロセス KLC4 は次のようになる．

(0) パラメータ α,β を設定する．ルートノードをつくり T と共に開始する．
(1) 当該ノードを葉とするかどうかを決める．具体的には T の全てのデータが同一クラスかどうかを調べる．
(2) そのとき葉ノードとし，呼び出し位置へ戻る．
(3) さもないとき，属性が残っていれば中間ノードとする
(3-1) クラスエントロピ $Ent(T)$ とクラス分布 $dist(T)$ を計算する
(3-2) T の各属性に対して
(3-2-0) $s = \phi$ とする
(3-2-1) A の各値 a に対して $dist(T)$ と $dist(T_{A(a)})$ の KL 情報量を求める．これが α より小さければ，$T_{A(a)}$ を s に加える
(3-2-2) s の要素数の割合が β 以上なら A を削除する．さもなければ相対エントロピ $Ent_A(T)$ を求める
(3-3) 情報利得最大の属性 G を選ぶ
(3-4) T における G 上の値に対応して子ノードを作り，これに応じて T を $T_1, ..$ に分割する．各々のノードに対して再帰的にこの手順を繰り返し，完了すれば呼び出し位置へ戻る．
(4) 属性が残っていなければ葉ノードとし，最も要素数の多いクラスを当該クラスとする．

KLC4 手法では，相対エントロピを計算する前に，属性が決定木生成に不必要かをチェックすることができる．この意味で，事前枝刈りだけを考察したアルゴリズムに過ぎないが，KL 情報量と相対エントロピではクラス分布を用いるので，効率良く決定木を計算することに注意されたい．無論，事後枝刈りを組み合わせて計算することは可能であるが，本稿では議論を単純にするために考えない．

[例 3] 図 1 のデータ多重集合 T から決定木を作成する．以下では，$\alpha = 0.1$,$\beta = 0.8$ とする．

クラスが Yes であるオブジェクトは 9 個，No であるオブジェクトは 5 個 あるので，クラス分布 $dist(T)$ は

$$\langle (9/14), (5/14) \rangle = \langle 0.642857, 0.357143 \rangle$$

となる．クラスエントロピは例 2 のように $Ent(T) = 0.940286$ である．

次に各属性が決定木生成で不必要かどうかを判定し，不必要でなければ相対エントロピを計算する．属性 `Age` の各属性値の KL 情報量を計算する．各属性値が持つクラス分布は次のようになる．

$$dist(T_{Age(\leqq 30)}) = \langle 0.4, 0.6 \rangle$$
$$dist(T_{Age(30-40)}) = \langle 0.8, 0.2 \rangle$$
$$dist(T_{Age(40<)}) = \langle 0.75, 0.25 \rangle$$

各属性値が持つクラス分布を KL 情報量を用いて比較すると

$$KL(dist(T)||dist(T_{Age(\leqq 30)}))$$
$$= (0.642857) \log_2(0.632857/0.4)$$
$$+ (0.357143) \log_2(0.357143/0.6) = 0.172727$$
$$KL(dist(T)||dist(T_{Age(30-40)}))$$
$$= (0.642857) \log_2(0.632857/0.8)$$
$$+ (0.357143) \log_2(0.357143/0.2) = 0.0959279$$
$$KL(dist(T)||dist(T_{Age(40<)}))$$
$$= (0.642857) \log_2(0.632857/0.75)$$
$$+ (0.357143) \log_2(0.357143/0.75) = 0.0408096$$

$\alpha = 0.1$ 以下の KL 情報量を持つ属性値は $30 - 40$，$40 <$，この 2 つの属性値が持つデータの総数は 9 である．ノードが持つ全データに対する割合は $9/14 = 0.64285$，$0.64285 < \beta = 0.8$ なので属性 `Age` は削除の対象にならない．

他方，属性 `Student` が対象になることは次のようにしてわかる．

$$KL(dist(T)||dist(T_{Student(Yes)}))$$
$$= (0.642857) \log_2(0.632857/0.571429)$$

$$+ (0.357143)\log_2(0.357143/0.428571) = 0.0152966$$
$$KL(dist(T)||dist(T_{Student(No)}))$$
$$= (0.642857)\log_2(0.632857/0.714286)$$
$$+ (0.357143)\log_2(0.357143/0.285714) = 0.017258$$

α 以下の KL 情報量を持つ属性値が持つデータの数は 14，ノードが持つ全データの割合 は 1.0 である．$1.0 > \beta$ なので属性 `Student` は考察から除外する．これ以外の属性は決定木生成に貢献していると判断する．

結局，`Age, Income, Credit Rating` の相対エントロピはそれぞれ次のようになる．

$$Ent_{Age}(T) = 0.836393$$
$$Ent_{Income}(T) = 0.775339$$
$$Ent_{CreditRating}(T) = 0.85001$$

情報利得が最大になる属性 `Income` をノード属性に選択する．

本手法を用いて決定木を作成すると，例 2 と同じ図 2 を得る．

5. 実　　験

5.1 実験手順と実験結果

この章では，テストデータを用いて，KLC4 手法の有効性を示す．実験のために，KLC4 を用いたプログラムを開発し，C4.5 と比較する．ここではパラメータ α, β の値をそれぞれ $0.15, 0.8$ とし，また C4.5 の設定は標準値 (すなわち，枝刈りの基準を誤り率 25% (`-c25`) とし，どの検査も複数の枝が複数のデータを使って行う (`-m2`)) を用いる．なお，ゼロ除算を避けるため，クラス分布中の 0.0 値はすべて 0.001 で置き換える．

実験では *UCL ML data archives* を用いる [14]．他研究と比較できるようにいくつかの *breast-cancer-wisconsin* (Breast), *Credit Screening*, *Car Evaluation Database*, *German*, *Vote* および *Molecular Biology Databases* についての実験の要約を示す．

Data	Training	TestData	Class	Attrs	(Real)
DNA	2000	1186	3	180	0
CarEval	1383	345	4	6	0
German	667	333	2	20	6
Credit	465	188	2	15	6
Breast	455	228	2	9	0
Vote	200	100	2	15	0

Data	Paths	IntermNodes	ErrorRatio
DNA	10 (64)	9(63)	0.13 (0.07)
CarEval	67 (98)	23 (40)	0.177 (0.13)
German	4 (26)	1 (10)	0.288 (0.252)
Credit	2 (32)	1 (9)	0.16 (0.165)
Breast	76 (46)	18(5)	0.039 (0.057)
Vote	8 (3)	4 (1)	0.07(0.07)

数字は，KLC4 手法の結果を，また (..) は C4.5 の結果を表す．各項目は *Data* (データ種別)，*Training* (訓練事例数)，*TestData* (テスト事例数)，*Class* (クラス数)，*Attrs* (属性数)，(*Real*) (そのうち実数属性数)，*Paths* (パス数、これは葉ノード数に対応)，*IntermNodes* (中間ノード数)，*ErrorRatio* (誤分類率) を示している．以下では代表的に *Credit Screening* および *Molecular Biology Databases* (Splice-junction Gene Sequences Database，DNA Database) について実験の詳細を述べる．

5.2 Credit Screening

Credit Screening はいくつかの欠損値を含む　690　件のデータを持ち，2 つのクラス $+, -$ と 15 個の属性 $A1, .., A15$ からなる。Credit Screening はクレジットカード審査に関わるため，すべての属性と値はデータの守秘性を守るために無意味な記号に置き換えられている．本実験では，欠損値を含むデータを削除した後にトレーニングデータとテストデータに分割した．その結果、データ数は 653 件となり，トレーニングデータ 465 件，テストデータ 188 件に分割する．また，連続値で構成される属性を平均で分割し，平均より大きい，平均より小さいという 2 つの属性値に置き換えた．Credit Screening を構成する属性の値は `A1` (b, a)，`A2` $(< 31.58, 31.58 \leqq)$，`A3` $(< 4.96, 4.96 \leqq)$，`A4` (u, y, l, t)，`A5` (g, p, gg)，`A6` $(c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff)$，`A7` $(v, h, bb, j, n, z, dd, ff, o)$，`A8` $(< 2.45, 2.45 \leqq)$，`A9` (t, f)，`A10` (t, f)，`A11` $(< 2.6, 2.6 \leqq)$，`A12` (t, f)，`A13` (g, p, s)，`A14` $(< 177.014$, $177.01 \leqq)$，`A15` $(< 1166.05, 1166.05 \leqq)$ となる．

以下に Credit Screening に対する実験結果を示す．

	パス数	ノード数	誤分類率
KLC4	2	1	0.16
C4.5	32	9	0.165

KLC4 で生成した決定木と C4.5 で生成した決定木を比較すると，誤分類率はほぼ変わらない値を示しているが，パス数は 1/15 と大幅に減少している．図 3 と図 4 を比較すると，C4.5 による木では上位ノードで出現している属性が，KLC4 手法による決定木ではルートノード生成時に削除されている．また，ルートノードとその子ノードで多くの属性を削除されている．これが，木の大きさを削減させる要因になっている．

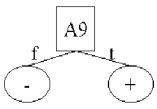

図 3　KLC4 による Credit Screening の決定木

5.3 DNA Database

DNA の接合とは DNA 列上の点を表しており，この部分でタンパク質の生成過程中に余分な DNA が取り除かれる．このデータ集合では与えられた DNA 列に対して exon (DNA の接合の後にでも保存される DNA 列の一部) と intron (保存されない部分) の境界を見出すことを目的としている．従って，問題は 2 種類あり，exon/intron 境界 (*EI site* と呼ばれる) の検出と，intron/exon 境界 (*IE site*) の検出にある．遺伝子工学分野では，後者は acceptor，前者は donor と呼ばれている．DNA Database は DNA クラス (EI(25%),IE(25%)，どちらでもない (50%)) からなる 180 個の特徴値属性 $A0, .., A179$ を持つ 3186 件のデータからなる．本稿では属性 $A0, .., A179$ をそれぞれ属性 0,.., 属性 179 という．すべての属性が持つ属性値はバイナリ $(0, 1)$ である．これを 2000 件の訓練データと 1186 件のテストデータに分ける．なお，訓練データにおける属性値

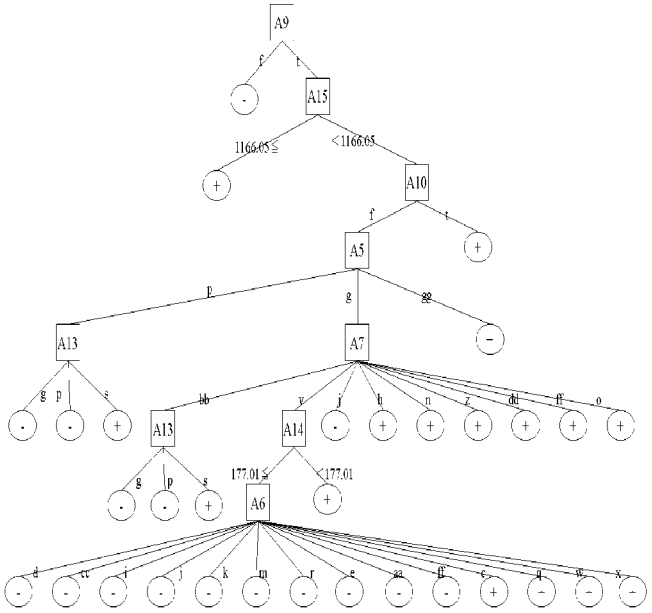

図 4　C4.5 による Credit Screening の決定木

間の統計的相関性は -0.236 ないし 0.119 の範囲のため，正相関 (1:1, 0:0) も負相関 (0:1, 1:0) も見られない．

以下に DNA データの実験結果を示す．

	パス数	ノード数	誤分類率
KLC4	10	9	0.13
C4.5	64	63	0.07

ルール数，ノード数が共に 1/6 になっており，極めてコンパクトになったことを得ている．半面，誤分類率が悪化しているが有効性を失うほどではない．図 5 に 本手法での決定木の構築結果を示す．また，図 6 に C4.5 での決定木の構築結果の一部を示す．

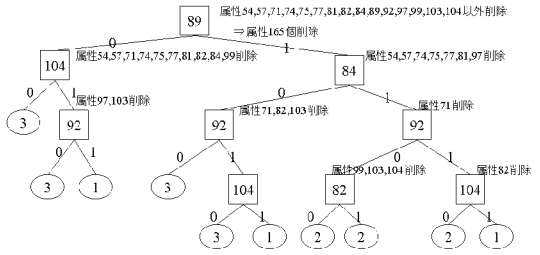

図 5　KLC4 による DNA の決定木

本手法により生成された決定木を調べると，根ノードで 180 個中 165 個の属性が除去され，これ以降のノードでも同様にいくつかの属性が除去されている．最終的に属性 89 を根ノードとして構成するが，この属性値が 0 のとき，クラス分布 $< 73, 2, 766 >$ となり，属性値が 1 のとき，クラス分布 $< 391, 483, 285 >$ を得るので両枝は展開される．たとえばノード 89 値 1，ノード 84 値 0 下のノード 92 では値 0 で分布 $< 0, 0, 135 >$，値 1 で分布 $< 151, 0, 64 >$ を得る結果，値 0 の枝をクラス 3 の葉として停止し，値 1 の枝は展開される．このように，全体として検出しにくい局所的な値のばらつきを吸収

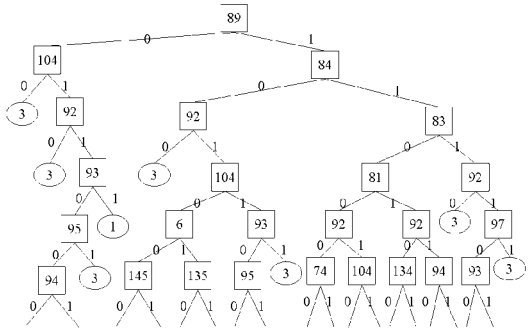

図 6　C4.5 による DNA の決定木

することができるため，特に初期段階で多くの属性を削除しており，決定木の生成も効率よく行われたことを示している．

本手法は初期段階で多くの属性を削除しており，決定木の生成も効率よく行われたことを示している．削減結果からわかるように，本手法で作成した決定木はパス数を大幅に減少させており，しかも誤分類率が低い．図 5 と図 6 の比較から，C4.5 による木では上位ノードで出現している属性が，本手法による決定木では出現していない．実際，本手法では根ノード検査時に削除しており，木の大きさを大幅に削減させる要因となっている．

6. 関 連 研 究

近年，決定木研究は分類精度向上を目指した研究が盛んに報告されている．一般には，先ず大きな木を生成し何かの基準でこれを縮小するという事後枝刈り手法が主流である [2], [10]．枝刈りに使用される基準は，カイ二乗検定などの統計手法，誤り予測 (C4.5 等の悲観的な予測)，木の複雑さに関するもの (MDL 等) が用いられる．これらは，いったん生成された木から”有害”部分を除去するという考え方による．このような事後枝刈りに対する疑問は，近年多く指摘されており大量のデータにたいしても単純な規則が正確な分類を与える報告がある [4]．反面、決定木の事前枝刈りに対する疑問も古くから指摘されており [9]，統計的な仮定を予め与えにくい，あるいは近視 (local minimum) 的な結論を得やすい等の理由で，これまで注目を浴びてこなかった．

事前枝刈りの例として，PreC4 アルゴリズムが代表的である [8]．決定木生成の前に，訓練データを交差検定して大域的誤り予測を行い，この値を越えると期待できる部分の生成を抑制する．決定木はこの割合に近づくように誤差二乗測定を繰り返しながら生成される．特に DNA Database に対する分類精度に関しては C4.5 とほとんど同じ値であったが，決定木の大きさではおおよそ半分の大きさになったと報告されている．また，事前枝刈りと事後枝刈りを組合わせた方法も提案されている [5] が，決定木に特化したものではない．

相関規則を用いた決定木の生成に関してもいくつかの報告がなされている．寺邊らは，属性間の相関規則をもとに属性を統合し訓練データに付加することによりデータの記述を変更する方式を提案した [13]．相関規則の抽出に Apriori アルゴリズムを採用しこれを前処理段階で適用する．Car Evaluation Database に対して 6 属性上で 10 回の交差検定で 173.2 ノー

ドである木が 18 属性 88.7 ノードまで減少し，誤分類率も 34% の向上が見られたという．

Liu は，属性間の相関規則を用いて分類器を導出するアルゴリズムを提案し，C4.5 よりも良い分類性能を得ることを示した [7]．しかし，導出される分類器は決定木形式ではないため，直接の比較を行うことができない．

7. 結　　論

本論文では，決定木の事前枝刈り手法 KLC4 を提案し，木の生成に効果的ではない属性を発見し，これを除外するアルゴリズムを提案した．このため，局所的な相関規則の検出とクラス分布の類似性の判定を用いて，属性の除去手順を構築した．いくつかの実験を行い，KLC4 手法により誤分類率の有用性を残したまま大幅に枝刈りできることを示した．

今後の課題として，数値属性および欠損値に対する対応，クラス階層を用いた決定木生成への応用 [11], [12] がある．

また，本論文で設計したアルゴリズムは我々の研究室のホームページによりフリーウェアとして公開予定である．(http://www.dbl.k.hosei.ac.jp)

謝　　辞

本研究の一部は文部科学省科学研究費補助金 (課題番号 14580392) の支援による．

文　　献

[1] Agrawal, R., Srikant,R.: Fast Algorithms for Mining Association Rules, VLDB 1994

[2] 古川康一，尾崎知伸，植野研: 帰納論理プログラミング, 共立出版 (2001)

[3] Han, J. and Kamber M.: Data Mining - Concepts and Techniques, Morgan Kauffman (2000)

[4] Holte, R.C.: Very Simple Classification Rules Perform Well on Most Commonly Used Datasets, *Machine Learning* 11-1, 1993

[5] Fürnkranz,J.: Pre-Pruning and Post-Pruning, *Machine Learning* 27-2, 1997, pp.139-171

[6] Kohonen, T.: Self Organizing Maps, Springer-Verlag (1995)

[7] Liu,B.,Hsu,W., and Ma,Y.: Integrating Classification and Association Rule Mining, proc. *Conference on Knowledge Discovery and Data Mining*, pp.80-86, 1998

[8] Pfahringer, B.: Inducing Small and Accurate Decision Trees, *Technical Report* TR-98-09, Oesterreichisches Forschungsinstitut für Artificial Intelligence, Wien, 1998.

[9] Quinlan, J.R.: C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993 (古川康一 訳: AI によるデータ解析, 株式会社トッパン (1995))

[10] Safavian, S.R. and Landgrebe, D.: A survey of decision tree classifier methodology, IEEE *Transactions on Systems, Man, and Cybernetics* 21(3), pp. 660-674, 1991

[11] Takamitsu, T., Miura, T. and Shioya, I.: Testing Structure of Decision Trees, proc.*Info. Syst. and Engr.*(ISE), 2002

[12] Takamitsu, T., Miura, T. and Shioya, I.: Decision Trees using Class Hierarchy, proc. *Hybrid Intelligent System* (HIS), 2003

[13] 寺邊正大, 片井修, 槻木哲夫, 鷲尾隆, 元田浩: 相関ルールにもとづく属性生成手法, 人工知能学会論文誌 15-1, 2000

[14] UCI Machine Learning Repository, `http://www1.ics.uci.edu/ mlearn/`