# 決定木の緩和からの知識獲得

塩谷　勇[†]　三浦　孝夫[††]

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

**あらまし**　この論文は、概念階層を用いた決定木からの知識獲得を目的とする。我々は概念階層に沿ってオブジェクトの属性値とクラス値を緩和 (階層レベルを上げて抽象化) し、与えられた制約条件（「決定木のサイズ」と「エントロピーに基づく分類の質」）を満足する決定木を生成する。本手法を用いることで、利用者が対話的に制約条件を変えながら決定木を生成して、分類学習の結果から新たな知識発見に利用できる。
**キーワード**　決定木、概念学習、概念階層

# Knowledge Discovery of Decision Trees by relaxing Domain Knowledge

Isamu SHIOYA[†] and Takao MIURA[††]

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

**Abstract**　This paper presents a new method to discover knowledge in decision trees by utilizing *hierarchical knowledge* (semantic hierarchies among class memberships), and the method generates decision trees by relaxing the hierarchical levels along the knowledge. Then, we specify design constraints concerning the sizes of decision trees and the qualities to classify. Unlike a conventional pruning method disregards a part of decision rules for avoiding an overfitting after (or while) generating decision trees, we relax classifications along hierarchical knowledge for creating classification concepts while generating decision trees.
**Key words**　Decision Trees, Conceptual Lerarning, Background Knowledge

## 1. はじめに

決定木 (*decision tree*) は教師あり分類学習法の一つである [8], [9]。しかし、サイズの大きい決定木は概念的な理解や解釈が難しい。コンパクトな決定木を得ようとする研究 [2], [9], [13], [15] が行われているが、現象をある側面から可能な限り正確なモデルで構築しようとする試みである。複雑な現象の解析結果が再び複雑ならば、我々は理解可能な意味のあるモデルを生成することはできない。すでに、『伝統的な手法で得られる分類は概念的な解釈が難しく、意味のある分類を作り出すには概念階層が必要である』[4] と指摘されており、決定木の生成時に属性の概念階層を利用する手法が知られている [7]。この方法は属性のみに概念階層を用いた決定木の生成手法であり、クラスの概念階層が扱えないために、概念的な理解が容易な決定木の生成は一般に難しい。一方、サイズに関する制約条件を満足する決定木生成手法 [2] が提案されているが、概念階層が用いられないために複雑な決定木からのコンパクトな決定木を得るための過度な枝刈りは、一般に分類誤差の増加から決定木の分類が意味を成さなくなる。このように概念抜きに物事を語る事ができない [11]。一方、数値を区間分割して質的データと量的データを同時に扱う技法は、抽象化の一つの技法と考えられる [9], [14]。

この論文は概念階層を利用した木のサイズと分類の質に基づくコンパクトな決定木の生成手法を用いて、決定木間の順序から新たな知識発見手法を提案する。決定木の生成は以下のように行われる。属性値とクラスを階層知識に沿って緩和 (概念階層に沿って抽象化の低い記述をより高い記述に書き換える) することで、与えられた「木のサイズ」と「エントロピーに基づいた分類の質」を制約条件とした決定木を生成する [10]。生成過程で木のサイズの制約条件を満たさなくなると、分類に最も有効でない利得の最も小さい属性を概念階層に沿って緩和する。一方、同じ属性値であるが階層の上下関係にない異なるクラス

の属するオブジェクトはクラス階層に沿って緩和し、異なった分類を無矛盾な状態にクラスを書き換える。緩和によって分類の質が低下するため、動的に属性を非決定的に再選択する。このとき、如何に正確な分類をするというよりも、分類情報をできるだけ失わずに階層に沿って緩和することである。良い分類は決定木の複雑さ (すなわち木のサイズ) に反映し、階層の緩和は理解可能な意味のある分類に反映する。Garofalakis ら [2] は同様の動機で議論をしているが、概念階層を使用しないため、コンパクトな決定木を得るための過度の枝狩りが分類誤差を増大させる。

我々は木のサイズ (高さと幅) と分類の質を制約条件とし、それらを4つのパラメータで表す。最初の2つは $\alpha_h$ と $\alpha_w$ であり、木のサイズを表し、それぞれ決定木の高さ比 (*height ratio*) と 幅比 (*width ratio*) と呼ぶ。他の2つは $\beta$ と $\gamma$ であり、分類の質の対応する。$\beta$ は緩和閾値 (*abstraction threshold ratio*) と呼び、エントロピー減少の尺度であり、階層緩和の各ステップに於けるエントロピーの極端な減少を避けるためである。すなわち、エントロピー尺度によって極端な緩和を制限する。$\gamma$ は、選択閾値比 (*selection threshold ratio*) と呼び、属性選択の制御のために、初期状態からのエントロピーの減少割合の上限である。木の生成方法は他の手法と比較してエントロピーの尺度でかなり分類情報を失う。情報を失う過程は概念階層によって導かれるため、概念階層の観点からのエントロピーに基づいた分類の質を基準として決定木が生成される。我々のアルゴリズムは属性再選択のために後戻りを伴う ($\gamma$ で制御可能) が、後戻り時に重複した計算をできるだけ避けるように工夫をしていると同時に、階層に沿って緩和するために生成した決定木は分類の誤りがない。本手法を用いることで、利用者は概念階層に沿って制約条件を変えながら理解容易な複数の決定木を対話的に生成して視覚的に表示して、決定木間の順序から新たな知識発見に利用できる。

この論文は、第2章で必要な幾つかの定義を行う。第3章で決定木の生成方法について論じる。第4章で実験結果を示し、第5章で結論を述べる。

## 2. 準　　備

オブジェクトはあるクラスに所属し、またクラスを規定する属性を仮定する。あるクラス $c$ の属するオブジェクトの集合を $\Gamma(c)$ で表す。各オブジェクトは同一の形式からなり、表 (以下、オブジェクト集合(注1)と呼ぶ) $T$ で表されると仮定する。

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

$T$ の各行は一つのオブジェクト $t_j : c_j$ を表し、$t_j$ は属性 $A_i \in \mathcal{A}$ $(i = 1, \cdots, k)$ 上の特徴値ベクトルである。$\mathcal{C} = \{c_1, \cdots, c_q\}$ $(c_i \neq c_j (i \neq j),\ q \leqq n)$ はクラスの集合である。$t(A_i)$ はオブジェクト $t : c = (a_1, \cdots, a_k) : c$ の属性 $A_i$ の値 $a_i$ を表す。

クラス階層 (*class hierarchy*) は ISA 関係の集まりで、根を除いて必ず親が一意に定まる。任意のクラス $c_1$ と $c_2$ に対して、$\Gamma(c_1) \subsetneqq \Gamma(c_2)$ ならば、$c_1$ ISA $c_2$ と表す。クラス階層が単純 (*single*) とは $\Gamma(c) \cap \Gamma(c') \neq \phi$ ならば $c$ ISA $c'$ または $c'$ ISA $c$, すなわち $\Gamma(c) \subsetneqq \Gamma(c')$ または $\Gamma(c') \subsetneqq \Gamma(c)$ を表す。この論文ではすべてのクラス階層が単純と仮定する。また、すべての属性 $A_i$ $(i = 1, \cdots, k)$ に於いても階層 $\mathtt{ISA}_{A_i}$ を仮定し、各階層がまたは単純と仮定する。

[例 1] 図 1 に *Race Condition* の例を示す。3つの属性: *Weather*, *Temperature*, *WindForce* から *RaceCondition* を決定する。クラス *Held* はレースが開催、*Half* は部分開催、*No* は中止を表す。同じ特徴ベクトルであるが異なるクラスに所属するオブジェクトが存在する。また、図 2 の ISA 階層を仮定する。

| Wea | Temp | WindF | RaceCond |
|---|---|---|---|
| Fine | Medium | Windy | Held |
| Fine | High | Very Windy | Half |
| Fine | Very High | Windless | No |
| Fine | Low | Breeze | Half |
| Fine | Low | Breeze | Held |
| Cloudy | Low | Windy | Held |
| Cloudy | High | Breeze | Held |
| Cloudy | High | Very Windy | Half |
| Cloudy | Low | Windless | Held |
| Rainy | Low | Windy | No |
| Rainy | Very Low | Very Windy | No |
| Rainy | Medium | Windless | Held |
| Rainy | Low | Windless | Held |
| Rainy | Low | Breeze | Half |

図 1　Race Condition.

| 属性またはクラス | A | B |
|---|---|---|
| Weather | Fine | NotWet |
| | Cloudy | NotWet |
| | Rainy | Wet |
| | NotWet | DontCareWeath |
| | Wet | DontCareWeath |
| Temperature | VeryHigh | Hot |
| | High | Hot |
| | Medium | Warm |
| | Low | Cold |
| | VeryLow | Cold |
| | Warm | Comfortable |
| | Cold | Comfortable |
| | Hot | NotGood |
| | Comfortable | DontCareTemp |
| | NotGood | DontCareTemp |
| WindForce | VeryWindy | Wind |
| | Windy | Wind |
| | Breeze | NoWind |
| | Windless | NoWind |
| | Wind | DontCareWin |
| | NoWind | DontCareWin |
| Race Condition | Half | Held |
| | No | DontCareRace |
| | Held | DontCareRace |

図 2　Race Condition の概念階層 A ISA B.

## 3. 決　定　木

### 3.1 決定木の生成

決定木の生成法が従来の方法と異なる点は、クラス階層の利用にある。通常、クラス値 $c_1, \cdots, c_q$ は排他的と仮定され、エントロピー $-\Sigma p_i \cdot \log_2 p_i$ は排他的なクラスに関する情報量の期待値である。我々は単純なクラス階層を仮定しており、クラ

(注1)：出現頻度を考慮しているので、正しくは集合でない。

ス階層の基で決定木の生成法を再構成する必要がある。クラス階層のエントロピーは、階層のすべての分岐を合計したエントロピーと定義する。エントロピーの定義から、クラスの再分類に関して単調に増加することは明らかで、通常のエントロピーの定義の拡張になっている。

利用者の概念的な理解や解釈が容易な決定木を生成するためには、分類誤差のない良い分類のみでなく、分類の情報量を失うが抽象度の高い記述でできるだけ良い分類をする必要がある。しかし、自明な結果(注2)になるかもしれない。我々の方法は概念的に解釈可能な意味のある分類の情報を失うことなく抽象化を行う。我々はより正確な決定木を得ようとする一方で、階層に基づいた分類を行うために、かなりの情報を失う。すなわち、抽象度の高い表現により分類の有用な知識を得ようとするが、一方で、抽象化により分類の情報を失う。我々は決定木の生成中に入力表データ $T$ をしばしば参照し、動的にクラスや属性の値を書き換える。

我々は決定木の複雑さを、木のサイズから定義する。なぜならば、高さは分類条件の連元の長さに対応し、一方幅は規則数に対応する。ここでは決定木の複雑さを木の深さと幅で定義する。

表データ $T$ の属性 $A_j$ に $u_j$ 個の異なる値が現れ、決定木で $r$ 個の属性が選択されたと仮定し、$u = \frac{u_1 + \cdots + u_r}{r}$ とする。最悪の場合、結果は $u$-ary 木になり、高さが $\lceil \log_u(u_1 \times \cdots \times u_r) \rceil$ になる (我々はこれを $T$ の木の高さと呼ぶ)。しかし、$\lceil \log_u n \rceil$ と期待される。ここで、$n$ は $T$ の行数である。

高さの複雑さ比 (*height complexity ratio*) $\alpha_h$ とし、高さ制約条件 (*height condition*) $\lceil \alpha_h \cdot \lceil \log_u n \rceil \rceil$ と定義する。幅についても、$T$ の幅の複雑さ (*width complexity*) は $\lceil u \rceil$ によって定義し、各頂点の分岐数を表す。木の幅の期待値は $u^{\lceil \log_u n \rceil}$ である。同様に、幅の複雑さ (*width complexity ratio*) $\alpha_w$、幅の制約条件 (*width condition*) は $\lceil \alpha_w \cdot \lceil u \rceil \rceil$ と定義する。定義から、幅の制約条件を満たすか否かが決定木の生成中に他の枝とは独立に調べることができる。

制約条件を満足する決定木が見つからず、決定木の生成ができない場合は属性の階層レベルを上げる。数値の *High-Salary* と *LowSalary* のように、非数値の場合は階層の緩和技法 [1], [3], [5], [7] を採用する。これはクラスにも適用できる。

属性値の書き換えは概念階層が単純でなくなり、矛盾状態になるかもしれない。表データ $S$ ($S \subseteq T$ の部分) の属性値を変更した後で、２つのオブジェクト $t : c_1$ と $t : c_2$ が同じ特徴ベクトルを持つが異なるクラスに属すると仮定する。定義から、$c_1$ が $c_2$ の先祖（またはその逆）ならば無矛盾である。さもなければ、どれらの中の最小の一般の値に書き換える。

より重要な場合は、特徴ベクトルが同じであるが $c_1$ と $c_2$ が排他的 ($\Gamma(c_1) \cap \Gamma(c_2) = \emptyset$) な場合である。$c_1$ と $c_2$ は矛盾する。この場合、オブジェクトの一つの特徴値を書き換えることで、この問題を回避する。一つの単純な階層を使うことで $c_1$ と $c_2$ の共通の最小元 $c_0$ を使う。このとき、$c_0$ への最小距離である $c_1$ または $c_2$ のいずれかを選ぶ。いまこれを $c_1$ とすると、$t : c_1$ を $t : c_0$ に書き換える。オブジェクト $t : c_1$ はクラス $c_1$ に属さないが $c_0$ に属するとする。$c_1$ `ISA` $c_0$ ならば、$c_1$ のクラス値は $c_0$ に抽象化される。このような書き換えはすでに作られている部分木の間でも生じるかもしれない。関係するオブジェクトを再度検査する必要がある。

我々の述べてきた事は、抽象化によって利得の極端な減少がないように設計者によって指定された制約を満たす範囲で可能な限り利得の意味で分類の質を保ちながら、オブジェクト集合を如何に抽象化するかである。

属性 $A$ の属性値の書き換えによってオブジェクト集合 $T$ を無矛盾に保つようにクラス値の書き換えることで、初期利得は $E(T)$ から $\frac{\Sigma_{i,j} v_{i,j} \cdot (\log_2 n_j - \log_2 n_i)}{n}$ に変化する。ここで、$v_{i,j}$ は $c_i$ から $c_j$ への変化の数である。我々は抽象化の各ステップで調べ、抽象度比 (*abstraction threshold ratio*) $\beta$ がこの範囲ならば許容する。利得の変化が大きい場合は計算を打ち切る。クラス値の変更は利得が単調に減少する。一方、属性値の書き換えは利得が減少したり増加することもあり、0.0 よりも大きいと保障できない。

我々は変化の量が最適な属性選択に比較して許容の範囲にあるかどうかを調べる。この目的のために選択閾値比 (*selection threshold ratio*) $\gamma$ ($0 \leq \gamma$) を導入し、変化の比が初期利得に比較して $\gamma$ の範囲に入っているかを検査する。その場合には再帰的に属性選択を行う。さもなければ、$\gamma$ を越えており、属性選択を行わない。

幅の制約条件を満たす決定木を如何に得るかを述べる。決定木の幅はルール数に等しい。我々の方法では複数の規則の一部を無視するのではなく、属性値の階層の緩和によって複数の規則を一つの規則に合併する。幅の複雑さの定義を思い起こすと、決定木の各頂点の枝の数についての制約である。概念的に容易に解釈できる決定木を生成することは、アンバランスな決定木、すなわち、決定木のある部分は詳細に分類され、他の部分は大雑把な決定木は望ましくない。我々は決定木を生成するときに、それを検査する。我々は幅の複雑さを満たすまで属性値を抽象化することで決定木の枝狩りを行い、すべてのオブジェクトが無矛盾になるように階層レベルを調整する。どの属性を最初に抽象化すべきか。ここでは利得に基づく手法を採用し、制約条件を満足する分類で最小の利得を与える属性値を先に抽象化する。決定木の生成の度に検査が必要となるから、事前に前処理で近似的な緩和を行う。

### 3.2 決定木生成アルゴリズム

決定木は以下のアルゴリズム $\texttt{NDT}(S, AT, AL)$ によって生成される。ここで、$S$ はオブジェクト集合、$AT$ は属性の集合、$AL$ は分岐条件の集まり (決定木) である。我々は4つの設計パラメータ $\alpha_h$, $\alpha_w$, $\beta$, $\gamma$ を使用する。オブジェクト集合 $T$ は属性 $A_1, \cdots, A_k$ 上の入力とすると、$\texttt{NDT}(T, \{A_1, \cdots, A_k\}, \phi)$ の呼び出しで計算が始まる。計算が成功すると、決定木の条件とクラス $\alpha_1(d_1), \cdots, \alpha_l(d_l)$ を出力する。

制約条件 $\alpha_h$, $\alpha_w$, $\beta$, $\gamma$ をどのように選択するか。それらは互いに独立ではない。我々の戦略は次の様になる。最初に最も

(注2)：例えば、すべての分類が同じクラス (根) である。

緩い条件 $\alpha_h = \alpha_w = \beta = \infty$ であるが、$\gamma = 0$ によって属性の再選択を許さない条件で決定木を生成する。得られた出力 (決定木) の高さが高ければ、$\alpha_h$ を調整して利用者によって理解しやすい適度な高さにする。第２に、属性選択の順序を変更する事で、より高い分類を得ようと試みるために $\gamma$ を大きい値に変更する。このとき、$\beta$ は各緩和のステップで最も緩い条件でエントロピーの減少を許しており、より小さい値 (例 2 の $\alpha_h$=0.7 の場合) に変更する。また、木の幅が大きければ、$\alpha_w$ を調整する。多くの決定木はこの様な設計プロセスを決定木の生成プロセスに組み込んでいない。我々の方法は分類が緩和されるが、分類誤差がない。我々は満足する決定木が得られるまでこのプロセスを対話的に繰り返す。

NDTA(S,AT,AL)
(1) $S$ のすべての特徴ベクトルが一つのクラス $d$ ならば、成功として $AL(d)$ を返す。$S$ のすべての特徴ベクトルは同じであるが異なるクラスに属するならば、$S$ のクラスを共通な最小のクラスに書き換え、クラス変更の個数を数える。クラス変更の総数による利得の変化が $\beta$ を超えるならば、失敗を返す。
(2) 未選択のすべての属性 $A$ の利得 $E_A(S)$ を計算する。
(3) 高さ比が $\alpha_h$ を超えなければ、
(3.1) 分類の最も有効でない最小の利得を与える属性 $A$ を選ぶ。
(3.2) $A$ の特徴値を書き換えて、矛盾がないように調整する。
(3.3) 同じ特徴値であるが排他的なオブジェクトはクラスを書き換えて、属性値の変更に伴うクラス変更の数を数える。すでに選択されている属性 $B$ について、$E_B(S)$ が $\gamma$ を超えるならば、このプロセスを決定的にする。エントロピーの変化量の比が $\beta$ を超えるならば、失敗とする。
(3.4) $S$ が無矛盾であるように調整し、(1) へ。
(4) 最大利得を与える属性 $A \in AT$ を選ぶ。
(4.1) $A$ 上の属性値に従って $S$ をグループ化して、これを $S_1, \cdots, S_r$ とする。グループ数が条件 $\alpha_w$ を超えるならば、(2) へ。ここで、各 $S_j$ は $S$ から $A$=“$a_j$” によって生成される。
(4.2) 各 $S_j$ に対して、再帰的に NDTA$(S_j, AT - \{A\}, AL \cup \{A =$ “$a_j$”$\})$ を呼び出す。呼び出しの返り値が再選択ならば、(2) へ。
(4.3) すべての結果を集め、$AL_1(d_1) \vee \cdots \vee AL_r(d_r)$ を返す。
(4.4) $A$ は処理済みとし、$E_A(S)$ を記憶する。
(5) $AT$ が空ならば、失敗を返す。

決定木生成アルゴリズム NDTA の計算量は制約条件に大きく依存する。実際、制約が無い場合 ($\beta$ と $\gamma$ が大きい) は多くの時間が必要とする。実際、決定木の生成に属性選択のすべての可能性を試み、緩和の制限がない。しかし、$\alpha_h = \alpha_w = \infty$、$\gamma = 0$ の場合は概念階層の緩和がなければ (分類誤差なし)、枝狩りのない C4.5 の答えと等しい。こ場合、利得に基づいた最適の属性選択を行う。$\gamma$ を大きくすると、得られた決定木は利得に基づいた最適な属性選択ないかもしれないが、大域的には最適な選択に近づく。

$\alpha_w$ は前処理によって近似的にオブジェクト集合を書き換えることができる。このとき、緩和する属性値はエントロピーに基づき、得られる解は近似的であるが決定木を生成する時間を大幅に節約できる。

また、我々は重複した計算を避けるために、近似式を使用している。しかし、近似のためにしばしば誤差が増大し、近似を用いないで再計算する必要がある。

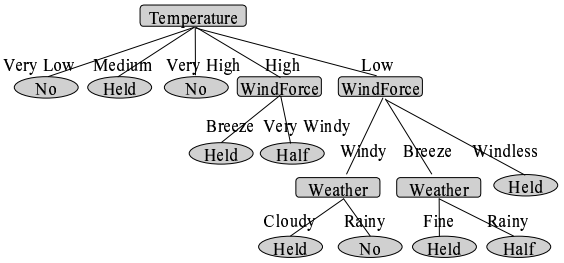


図 3 The decision tree: Race Condition. $\alpha_h = 1.1$, $\alpha_w = \infty$, $\beta = 0.02$, $\gamma = 0.0$.

[例 2] 例 1 について、$\alpha_h = 1.1$、$\alpha_w = \infty$、$\beta = 0.02$、$\gamma = 0.0$ と仮定すると高さが 3 である。このとき、初期エントロピーは 1.16658 であり、最小の利得 0.637596 を持つ属性 *Temperature* を選ぶ。属性 *Temperature*=“*High*” を持つオブジェクトについて、NDT はエントロピー 0.0 による属性 *WindForce* を持つ。規則 *WindForce*=“*VeryWindy*” (*Half* Class) と *WindForce*=“*Breeze*” (*Held* Class) を生成する。属性 *Temperature*=“*Low*” を持つオブジェクトに対して、NDT はエントロピー 0.452846 を伴って *WindForce* で３つの分岐を持つ。$\alpha_h$ = 1.1 に対して、$\beta = 0.01$ に対して決定木を生成できない。なぜならば、エントロピーの意味で抽象化の範囲を制限しているからである。一方、$\beta = 0.02$ ならば図 3 の決定木を得る。同様の傾向が $\alpha_h = 0.7$、$\alpha_h = 0.5$ の場合も見られる。制約 $\alpha_h$=0.7、$\alpha_w$=0.5、$\beta$=0.4475、$\gamma$=0.7 の下で決定木を生成すると図 7 が得られる。属性に関して抽象化がされる。このとき、56% のエントロピーが減少する。

我々のアルゴリズムの最初のステップで、決定木を生成する属性選択の順序は $\gamma = 0$ かつ緩和がなければ C4.5 と同じ結果になる。属性値やクラス値を緩和することで、属性選択の順序が変わってしまう。我々の方法が単なる詳細に分類する決定木の概要を得ているわけではなく、設計制約の下で意味のある分類を得ようとする。我々の提案はすでに得られている決定木の枝狩りではなく、設計制約の下で、階層を抽象化し、属性選択の順序を変えながら、設計制約を満たすまで計算を試みる。しかし、すべての試みをしても、意味のある分類が得られな場合は属性やクラスの階層を疑う必要がある。

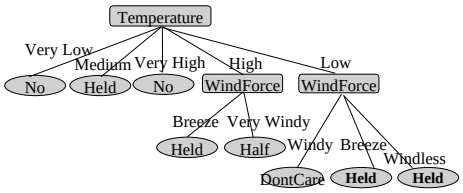


図 4　Race Condition の決定木。$\alpha_h$=0.7, $\alpha_w$=$\infty$, $\beta$=0.24, $\gamma$=0.0.

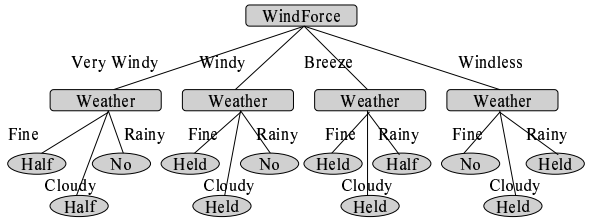


図 5　Race Condition の決定木。$\alpha_h$=0.7, $\alpha_w$=$\infty$, $\beta$=0.14, $\gamma$=0.5.

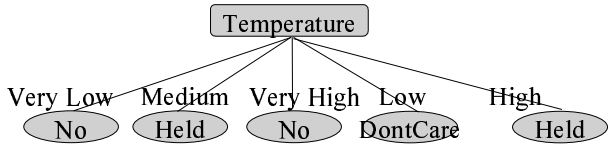


図 6　Race Condition の決定木。$\alpha_h$=0.5, $\alpha_w$=$\infty$, $\beta$=0.64, $\gamma$=0.0.

### 3.3　決定木の順序

決定木 $DT_1$, $DT_2$ に対して、すべてのオブジェクト $t : c \in T$ が次の条件を満足するならば、、$DT_2 \geq DT_1$ と定義する。$DT_1$ の根からの $t : c$ を分類する経路パス上のラベルを $a_1$ = “$x_1$”, $a_2$ = “$x_2$”, $\cdots$, $a_m$ = “$x_m$” がクラス $c'$ ($c$ ISA$_=$ $c'$ : $c = c'$ または $c$ ISA $c'$) に分類されると仮定する。このとき、$t : c$ は $DT_2$ 上ではパス $b_1$ = “$y_1$, $b_2$ = “$y_2$”, $\cdots$, $b_n$ = “$y_n$” によって $c''$ に分類されたと仮定する。このとき、$m \geqq n$ で、$b_i$ を並び替えた部分列 $b'_j$ は $a_i$ の部分列で、対応する $y_j$ についても $x_i$ ISA$_=$ $y'_j$ $(i = 1, \cdots, n)$、$c$ ISA$_=$ $c'$ ISA$_=$ $c''$。

決定木を順序の系列から、視覚的に新たな知識発見に利用する。特に、分類の概要から詳細までの複数の系列が一般にできるために、系列の意味するもの、または、同じ手法による複数の観点からの分類を見出すことができる。

[例 3]　例 1 について、分類の質とサイズに関する平面上に決定木を配置すると、図 8 のようになる。矢印が決定木間に順序の関係があることを示している。この例では２つの系列の決定木が生成され、大きく 3 通りの解釈が可能であることを示している。特に、この中で図 7 は分類の概要をよく捕らえており、

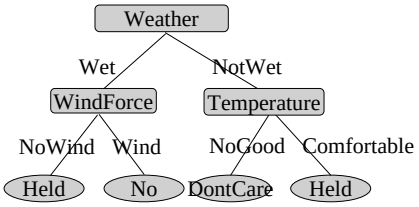


図 7　Race Condition の決定木。$\alpha_h$=0.7, $\alpha_w$=0.5, $\beta$=0.4475, $\gamma$=0.7.

レース開催 (“holding race”) の概要は

- $Wet$ の場合は、レース開催 “holding race” は風力 force of wind によって決定される。
- 雨が降らない ($NotWet$) 場合は、気温が適度ならば開催される、さもなければなんともいえない。

選択される属性の順序は属性選択の後戻りを通じて再選択される。

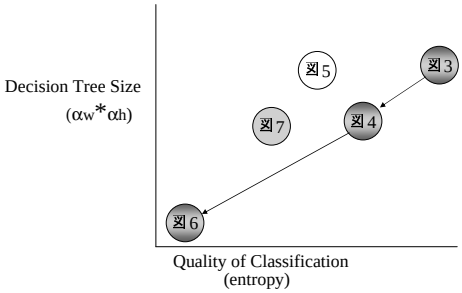


図 8　決定木の分類の質とサイズ

## 4.　実　　験

http://www.mi.sanno.ac.jp/~shioya/ に示される概念階層に基づいて UCI の car evaluation と nursery データについて制約条件を変えながら決定木を生成すると、決定木の分類の質のサイズの平面上に決定木の順序を描くと、いくつかの観点からの決定木の解釈の分類をすることができる。特にこの中で、図 9 と図 10 が興味深い分類結果として得られる。

car evaluation の決定木 (図 9) から、４人乗りの車で安全度

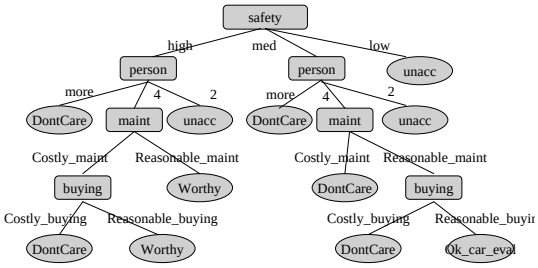


図 9　car evaluation の決定木.

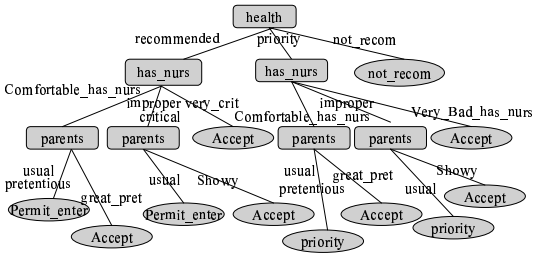


図 10　nursery の決定木.

が高い場合と中程度の場合の車の評価の違いが理解できる。すなわち、安全度が高ければ維持コストが高くても車の価格が適度ならば車の評価は良い。一方、車の安全度が中程度の場合は評価が分かれる。車の維持コストが適度な場合は、安全度が高ければ評価が良いが、安全度が中程度ならば維持コストに加えて車の価格が適度ならば車の評価が良い。すなわち、車の安全度が高くて4 人乗りならば車の価格が高くなければ維持コストが高くても良いと評価される。一方、車の安全度が中程度ならば維持コストと車の価格の両方が適度ならば評価が非常に良い。それ以外はさらに詳細に他の属性を調べる必要がある。車の安全度が高い場合と中程度の場合の車の評価の微妙な差異が決定木から読み取れる。nursery の場合も、健康状態が良い (recommended) 場合と優先順位が付けられている (priority) 場合のクラス分類の微妙な差異が、決定木 (図 10) から読み取れる。

通常、決定木の学習に於いて、学習データとテストデータの２つのグループに分けて学習結果の評価を行う。我々の目的は正確な分類を行う決定木を生成するのではく、学習データから概念的な理解が容易な決定木を生成する手法について言及しており、テストデータによる分類性能の良い決定木を生成する事が目的でないため、テストデータの分類性能による評価は意味をなさない。この理由は、一つは概念階層を用いており、概念階層の観点から見た分類に偏っており、客観的な評価は難しい。二つは生成された決定木の概念的な理解が容易とは主観的な基準であり、データ発見 (Data Mining) のように「これまでに知られていない知識が得る」の目的に類似している。car evaluation と nursery データから得られた決定木を見る限り、上記で述べた分類の微妙な差異が決定木から得られる。

## 5. おわりに

概念階層を利用して木のサイズと、エントロピーに基づいた属性再選択の基準パラメータを満足する決定木を生成し、決定木間の順序に基づいて分類の質と木の幅のサイズの関す平面状にそれらの関係を描くことで新たな知識発見手法について提案し、実験を行った。その結果、これまで多くの決定木生成手法と異なり、様々な観点からの決定木の分類を得ることができ、それらの決定木の間の半順序の関係から、新たな知識発見に利用できることが明らかになった。

本研究から、決定木の類似部分木検出や類似決定木の検出が新たな知識発見に重要と思われる [12]。決定木の部分木が同一であるか否かは枝の数に比例した時間で調べることができる。一方、決定木が類似しれいるか否かは『類似しているとは何か』を定義する必要があり、今後の検討課題である。

## 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] Hussein Almuallim, Yasuhiro Akiba, and Shegeo Kaneda. An efficient algorithm for finding optimal gain-ratio multiple-split tests on hierarchical attributes in decision tree learning. In *AAAI-96, revised*, 1997.

[2] Minos Garofalakis, Dongjoon Hyun, Rajeev Rastogi, and Kyuseok Shim. Efficient alorithms for constructing decision trees with constraints. In *Proceedings of KDD, Boston, MA, USA, August, 2000*, pp. 335–339. ACM, 2000.

[3] Jiawei Han and Y. Fu. Discovery of multiple-level association rules from large database. *Proceedings of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95)*, pp. 420–431, 1995.

[4] Robert E. Stepp III and Ryszard S. Michalski. *Conceptual Clustering: Inventing Goal-Oriented Classifications of Structures Objects*. 471–498 in [6], 1986.

[5] Ke Wang Senqiang Zhou Shiang Chen Liew. Building hierarchical classifiers using class proximity. In *Proceedings VLDB-99*, pp. 363–374, 1999.

[6] Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, editors. *Machine Learning, Vol. II*. Morgan Kaufmann, 1986.

[7] Marlon Nunez. The use of background knowledge in decision tree induction. *Machine Learning*, Vol. 6, pp. 231–250, 1991.

[8] J.R. Quinlan. Induction of decision trees. *Machine Learning*, Vol. 1, No. 1, pp. 81–106, 1986.

[9] J.R. Quinlan. *C4.5 - Programs for Machine Learning*. Morgan Kauffman, 1993.

[10] Isamu Shioya and Takao Miura. Knowledge pruning in decision trees. *Proceedings ICTAI-2000*, pp. 40–43, 2000.

[11] 吉田健一, 元田浩. 推論過程からの概念学習 (1)-類型的推論過程の抽出-. 人工知能学会, Vol. 7, No. 4, pp. 119–129, 1992.

[12] 大西健介, 吉田哲也, 西田正吾. 決定木の相関関係に基づいた概念相違検出手法. 電情通論誌, Vol. J85-D-I, No. 8, pp. 784–897, 2002.

[13] 寺邊正大, 片井修, 椹木哲夫, 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, Vol. 15, No. 1, pp. 187–197, 2000.

[14] 荒木大, 小島昌一. 数値データによる決定木の帰納学習. 人工知能学会誌, Vol. 7, No. 6, pp. 992–1000, 1992.

[15] 樋雄介, 稲積宏誠. 複合属性による領域分割を用いた決定木 dtmacc. 人工知能学会, Vol. 17, No. 1, pp. 44–52, 2002.