データベースシステム 119−61
（1999. 7. 23）

帰納分類によるトランザクション動作の発見

三浦孝夫† 塩谷 勇††

本稿では、機械学習手法に基づいて時制データに対してマルコフ性を仮定し 振舞い (トランザクション) に対して帰納分類手法を適用する また定常スキーマの計算と設計方法を論じ、極限分布を生成して定常スキーマの生成・評価を行う手法を述べる。

Discovering Behaviour Based on Inductive Classification

TAKAO MIURA † and ISAMU SHIOYA††

In this investigation, we propose a classification methodology to transition behavior based on machine learning techniques. We suppose Markovian property against temporal information, then we can show stationary schemes by calculating limit values of transition probability.

1. 動　機

本稿では、データベースに対する変化を時制データととらえ、帰納分類により振舞いの法則性を見出す問題を論じる。さらにこれを拡張し、データベース設計に反映させる技法を考察する。

本来、時系列データと時制データは異なる性質を有する。前者は地震や身長などの自然現象の計測のように個々の観測値が独立しているのに対し、後者は株式市況や競馬の掛け率 (オッズ) のように過去の情報から影響された変化として捕らえられる。このように、過去の世代からの情報のフィードバックは、流行・トレンドあるいは教訓といった言葉が意味するように、情報の遷移が規則を持ち、情報内容が特定の時間にだけ依存するのではないことを表す。更に法則性を取り出すことができれば、新たな知識として統一的な動作あるいはデータベーススキーマへ反映させることが可能となる。

変化 (遷移) のマルコフ性とは、過去の遷移に共通性があり、しかも時間に依存しないという仮定である。マルコフ記憶とは何世代前までの過去のデータからの影響があるかを表す'深さ'を意味する。マルコフ性が検出できれば、例えば、特徴項目に着目し、その値がどのように変化するかを捕らえ、共通した振舞いを見出すことができるであろう。

変化の結果の範囲が予想されるとき、振舞いのマルコフ性は安定した極限状態を導くことができる。すなわち、定常的にのみ変化する振舞いの存在が'計算可能'であり、現在の状況と独立にデータの変化を確定させることができ、更に驚いたことに、極限状態をデータベース設計の立場から評価することができる。

これまでのトランザクション設計やデータベース設計は、経験的だがそれを定性的定量的に評価する統一的手法を持たない。またデータベース成長過程で検証・修正する技術が無いため、定期的に見なおすトップダウン的アプローチ以外に取りようがない。本稿では、マルコフ性を有する時制データに対して、機械学習を用いた振舞いのパターンを機能的に分類する手法を提案する。また、データの変化から定常部を検出し、これを定常スキーマとして評価する方法を論じる。本稿のポイントは、遷移性を確率的に捕らえてマルコフ性を仮定すること、およびマルコフ記憶の有効性を最大限に利用することにある。

帰納分類によるトランザクション動作・スキーマ発見を行うためには、遷移性から振舞い規則を類推し、極限分布を生成して定常スキーマの生成・評価を行う手法を述べる。次節ではデータベース設計と機械学習の関連を要約し、3 節で具体的に帰納分類と時制データの関連を示す。4 節では変化を起こす振舞い (トランザクション) の帰納分類手法を論じ、そのあとで定常スキーマの計算と設計方法を論じる。

† 法政大学工学部
Hosei Univerity
†† 産能大学経営情報学部
SANNO College

2. 準　　備

2.1 データベース設計

データベーススキーマとは，オブジェクト集合を分類管理するための知識である。この知識に従って'概念' 情報が特定され，全体として統一的な意味を持つ個々のオブジェクトは、スキーマで定められた形式に従い保持される。本稿ではデータベースをスキーマ (クラス) 記述、そのオブジェクト集合および一貫性制約集合から構成されるとし、クラス c とオブジェクト d に対して $\tau(d)$ を d が有する有限型集合 $\{c_1,..,c_k \mid c_j$ はクラス $\}$ で、$\Gamma(c)$ で型 c を有する有限オブジェクト集合 $\{d_1,..,d_l \mid d_j$ はオブジェクト $\}$ を表す。ここでオブジェクト d がクラス c に所属する (あるいは有する) とは、$d \in \Gamma(c)$ であるとし、$d \in \Gamma(\{c_1,..;c_k\})$ とは d が $\{c_1,..,c_k\}$ のすべてのクラスを有することをいう。Γ と τ は互いに他方から計算可能である。このように個々のオブジェクトが個別に所属クラス情報を保持するモデルを実現値に基づく (*instance-based*) モデルという。

データベース設計とは、スキーマを確定することをいう。ここでは対象となる世界を処理するために、(1) 対象とする意味役割を簡潔に正しく記述し、(2) データを効果的・効率的に保守できる場をつくり、(3) 一貫性制約や想定される操作が容易に実現できるようにスキーマを記述する。$\Gamma(c)$ や $\tau(d)$ の効果的な管理と (1) については定性的に評価する以外に方法がなさそうであるが、(2), (3) のためには、何らかの知識機構を用いた統一的な制御が欠かせない。本稿では適切な設計の基準として、単純さ (simplicity) と妥当性 (appropriateness) を用いる [9]. 単純さとはどのオブジェクト d についても $|\tau(d)|$ が一定のサイズを超えないことをいう。妥当性とは、どのオブジェクト d についても $|\Gamma(\tau(d))|$ が一定以上のサイズを有する事を言う。すべてのオブジェクトが単純で妥当であれば、クラスの'粒度' が粗くも細かくもないことが保証される。この結果、質問の解析や型 (クラス) 検査・最適化や演繹などの知的な処理が統一的で効率よく構成できる。

データベース設計の過程は機械化することが難しく、また評価方法を得ることも難しい。経験的に、一貫性制約・演繹規則や (クラス) 階層などの知識間には相互に関連があることが多いことが分っている。例えばクラス階層関係はコンパクトな知識表現を作成するうえで重要である。しかしこれらは系統だって理解されないことが多く，設計 (知識の獲得) が大きい問題である [5]。しかし，その設計はこれまで知られた手法や知識獲得技術を用いたとしても，手作業に頼らざるをえない [20]。実際，クラス階層に関しては系統だった設計手法は知られておらず，経験的な知識や思いつくままに列挙する以外に手がない。さらに、単純さと妥当性の基準を満たすスキーマ設計は容易でなく、統一的な設計手法もほとんど知られていない。

データベースの変化とは、オブジェクトの所属するクラスの変更、特徴項目 (属性) 値の変化を意味する。ここでは、オブジェクトの生死とスキーマ変更は論じない。変化を引き起こす振舞いをトランザクションと呼ぶ。トランザクションの設計については酒井らの提案 [16] が知られるが、トップダウン分析に基づいており、特徴的な振る舞い (予兆) からデータの先読みや施錠、あるいはクラスタリング情報の抽出など、データベースの成長に伴う評価を反映させるものではない。

2.2 機械学習とデータベース設計

データベースの成長過程では、オブジェクトの担う意味も変動するため、スキーマも進化する必要がある。やっかいなことに、長期に渡って利用されているデータベースでは，設計されたスキーマが必ずしも現在の情報を記述するのに望ましいとは言い切れない。実際，新たな情報は共通の知識とする代わりに，格納データとして個別に埋め込まれることが多い。当初うまく表現できたスキーマが、次第にモデル化の意味・意義だけでなく単純性や妥当性からも適合しにくいものになるのはこのためである。このような情報は何らかの類似点や差異を抽出することでクラスタ化でき，'概念形成' に至る可能性がある [6]。データベースからの概念形成 (データベースからの知識獲得) は，より高度で精密な分野への適用がなされるにつれ，中心的で難しい問題となりつつある。この問題はトランザクションに対しても、そのまま当てはまる。

しかし、これまで知識獲得技術として知られた手法は，指数オーダの実行時間を要するアルゴリズムが多く，効率の観点からそのまま適用することが難しい。従って，現実には差分的な知識獲得や問題領域知識を活用する方式が妥当である。

著者らはこれまで、単純性と妥当性という分類基準からスキーマを見直し、クラス設計の見直し [8]、クラス階層の評価と近似手法による代替えスキーマの生成 [10]、あるいは帰納分類手法による設計支援 CSOP 手法の導入 [11] などを提案してきた。これらはスキーマに対して既に定義された既知情報を手がかりにして、クラスの構成の単純性や妥当性を評価するものであった。本研究ではオブジェクトのクラスメンバーシップ

や属性値の変化に着目し、これを時制データと見て遷移の法則性からトランザクションやスキーマ設計の検証手法を提案する。本稿で提案する手法によって、特にデータベーススキーマを変化を起こさない定常部と変動部に分けることができる。前者は、時間に独立した (安定した) 意味を有し従来のスキーマ本来に相当するが、後者は時間とともに変動する遷移部であり従来はインスタンス集合に埋め込む以外にモデル化する方法が無かった。本稿で提案する方法により、遷移部からの影響を取り去った部分に対して評価することができる。

3. 帰納分類とスキーマ発見

3.1 帰納分類

データベースからの機械学習とは，スキーマで表現すべき対象世界のモデル化の再構築を支援する技術であり，その結果は常にスキーマとして捉え直される。無論，現在の情報を表すオブジェクト集合の意味するものを，過去に定義されたスキーマ記述だけで正確に表現することは疑わしく，データ操作記述や何らかの意味記述を加えたものとせねばならない。

代表的な手法がデータベースの帰納分類である。通常、オブジェクトが所属すべきクラス集合 $\mathcal{C}$ は当初のデータベース設計で与える。これと同時に、対象となるオブジェクトに備わる属性値 (特徴項目) を用いてオブジェクト集合を何かの方法で分類し、クラス $c \in \mathcal{C}$ を属性上の条件 Δ_c で表現すれば、クラス設計の有効性を意味的に検証・検査ができる。このような方法を'例題から学んだ決定方法'(帰納的な分類方法) という。

典型的な帰納分類手法に決定木 (decision tree) が知られる。これは、各オブジェクトが共通して属性 (特徴項目) $A_1, .., A_m$ を有すると仮定し，その値の持つ意味を分析することによってオブジェクト集合 E の意味する概念を生成する手法である。決定木は与えられた情報を (既に判明している) 解集合のいずれかに分類するための情報表現手法であり，特徴値による判断を行う中間節 (分岐処理) と結果をラベルとする末端節からなる。中間節では特徴項目上の値を検査し，その値に応じて子節に分岐して処理を続けるが，この操作を根節から再帰的に行う。決定木を生成するとき，対象となる情報が一つのクラスになっていれば末端節を生成するが，さもなければ未処理の特徴項目のいずれかを選んで中間節の分岐処理に対応させる。即ち，項目 A をラベルとする中間節が枝上のラベル a に沿って経路を構成するとき，経路はクラスを確定するための連言条件を表す。分岐は和条件に対応する。決定木の構成方法から，各分岐はオブジェクト集合を分割し，しかもすべてのオブジェクトを生成する。決定木は全体としてクラス c 毎のメンバシップ条件 Δ_c をある属性 $A_1, .., A_h$ に対する $A_1 = "a_1" \wedge ... \wedge A_h = "a_h"$ の形の条件の和結合で記述する。ID3 や C4.5 などのアルゴリズムでは，特徴項目選定のために，エントロピの減少量あるいはその割合を元に算出した情報利得を最大にする様に配慮される [14),15)]。エントロピは事象の発生確率を用いて定義され、問題領域固有の背景知識を用いない。(時として妙な結果を生成することがあっても) 幅広い応用を生む理由となっている。

各クラス c ごとに根からの経路を合成することでそのクラスの特徴値による記述 $\alpha_1 \vee ... \vee \alpha_k$ を得る。これはクラスの特徴・意味を表すとみなせ、前述スキーマ設計条件 (1) に対する傍証として有用な情報を提供する。また、(2), (3) のために単純性や妥当性の検証が可能であり、見直しに対する代替え情報 (分割合成候補の提示) として各 α_j と対応するオブジェクト集合が得られる [10)]。

3.2 時制データとスキーマ発見

帰納的なクラス記述は、データベースの成長に伴って変化すると考えられる。ここでは成長の各段階 t を時制オブジェクト集合 E_t で、成長過程 (つまりオブジェクトの遷移) を時制データの動態 (振舞い) $E_t \rightarrow E_{t+1}$ とみなせば、時制データの帰納分類は、変化を引き起こす動作 (トランザクション) の帰納分析が重要なポイントとなる。以下では有限・離散的な非数値データを属性値の離散変化を対象とする。つまりオブジェクト d の特定が常に可能な状態で、その属性値 $A(d)_t$ の時制的な変化 $A(d)_t \rightarrow A(d)_{t+1}$ だけを扱い、オブジェクトの生死を考えない。時制的な変化は任意の '記憶の深さ' n で考察できる。実際 $\mathcal{E} = E \times A_1 \times ... \times A_m$ の時間 t でのデータを $\mathcal{E}_t$ とすれば、$\mathcal{E}_1 \times ... \times \mathcal{E}_n$ でのクラスメンバーシップに関する帰納分類は過去 n の深さに遡った考察を意味する。また、(クラス集合 $\mathcal{C}$ に対して $\mathcal{C}_1 \times ... \times \mathcal{C}_n$ を新たな超クラス集合とみなせば) クラスメンバーシップの所属変化 (トランザクションの振舞い) も帰納分類することができる。

4. 時制データのマルコフ性

4.1 エントロピと時制データ

決定木による帰納分類では、帰納性 (予測) の根拠としてエントロピを用いる。エントロピは、事象 i の発生確率を $p_i > 0, i = 1, .., k$ (k は有限) とするとき、$-\Sigma_{i=1}^{k} p_i \log_2 p_i$ と定義される。この式は、いずれの事象が発生したかを確定するために必要な 2 進決定

木の深さ平均を表す (大きいほど不確実さを表す)[18]。決定木生成の手順では、この算出のためにクラス c_i のオブジェクト数 (発生頻度) n_i の発生頻度比 n_i/N (N はオブジェクト数) が確率とされ、属性決定で考察されるオブジェクト集合内での (各時点での) クラスごとの要素数が予測結果に大きい影響を与える。

時制データ $E_1,..,E_t,..$ において、各オブジェクトのクラスメンバーシップは時間とともに遷移する。すなわち、クラス c_i から c_j へ変化したオブジェクト数を $n_{i,j}$ で表せば、遷移の確率 $p(i \to j)$ は $n_{i,j}/n_i$ で表せる。通常これは時間 t に依存し、事象のパターンをとらえることはない。一般的に変化 (遷移) がマルコフ的であるとは、遷移確率が時間 t に依存しないことをいう。影響を受ける過去のデータの世代数を (マルコフ遷移の) 記憶の深さという。深さ 1 (これを単純マルコフ遷移という) の場合クラス数 K とすれば遷移確率は K 次正方行列 $p(i \to j), i,j = 1..K$ で、深さ n の場合は $K^n \times K$ 次行列 $p(i_1,..,i_n \to j) = n_{i_1,..,i_n,j}/n_{i_1,..,i_n}$ で定義される ($n_{i_1,..,i_n}, n_{i_1,..,i_n,j}$ はそれぞれクラス $c_{i_1},..,c_{i_n}$ と遷移してきたオブジェクト数および、さらにそれが c_j に遷移した数を表す)。

遷移が過去の状態に依存しない (つまり独立事象) ならば $p(i_1,..,i_n \to j) = q(j)$ である。ここで $q(j)$ はオブジェクトがクラス c_j に所属する初期確率を表す。クラス c_i のオブジェクトが次の時間で c_j に所属する確率 $p(i,j)$ は $q(i) \times p(i \to j)$ である。また $p^{(t)}(i,j)$ により、クラス c_i のオブジェクトが t 時間後に c_j に変化する確率を表す。これは $\Sigma_{k=1}^{K} p^{(t-1)}(i,k) \times p(k \to j)$ で定義される。$q^{(t)}(j)$ は t 時間後にクラス c_j になる確率 $\Sigma_{k=1}^{K} q(k) \times p^{(t)}(k,j)$ を表す。単純マルコフ遷移の遷移行列 $P = [p(i \to j)]$ を仮定したとき、時間 t でクラス c_i に所属する確率 $q^{(t)}(i)$ をベクトルで表した $\vec{q}^{(t)}$ はクラスメンバシップ初期確率 $q(k), k = 1,..,K$ のベクトル表現 $\vec{q}$ と遷移確率の積 $\vec{q} \times P^n$ で決定される。

4.2 遷移の振舞いと帰納分類

これまでの議論を重ねると、トランザクション集合から動作のパターンを帰納的に分類できる。トランザクションとは各オブジェクトに対する変更操作であり、具体的には属性項目 A_i の値を a_i から a_i' に変化させしかも所属クラスを c_i から c_j に変更するものである：$\langle A_1 : a_1 \to a_1', .., A_1 : a_n \to a_n'; c_i \to c_j \rangle$。単純マルコフ遷移であると仮定しているから時間に依存せず、しかも各属性について 1 回の変動だけを記憶すればよい。

このとき、$A_1 \times ... \times A_n$ 上のテーブルで各オブジェクトは個々の行で変化状況が記述され、このときクラス変化 $c_{i,j}$ ($c_i \to c_j$) が対応する。この情報を決定木で分類することにより、クラス変化に対する分類が可能となる。

例題 1 CPU のクロック数およびハードディスク容量の変化で サーバとしての役割の変化が生じるとする。

CPU	ハードディスク	サーバ
500:550	6:13	Web:File
500:550	6:6	Web:Web
400:400	6:8	Web:Web
400:450	6:6	Web:Web
300:400	6:6	Web:File

この変化はつぎの決定木で表現される:

ハードディスク	CPU	サーバ
6:6	500:550, 400,450	Web:Web
	300:400	Web:File
6:8		Web:Web
6:13		Web:File

例えば Web:Web クラス変化はハードディスク容量が 6:6, 6:8 で期待できる可能性が高いため、Web クラスに対する予備的な施錠を実施することが考えられる □

5. 定常スキーマと遷移スキーマ

この節では、単純マルコフ遷移を仮定して定常スキーマ・遷移スキーマの検証技法を述べる。

5.1 定常分布の計算

エルゴード理論に従えば、十分に大きい t をとれば どのような i,j についても $p^{(t)}(i \to j) > 0$ となるとき (つまりどのクラス間の遷移も常に保証されれば)、$\lim_{t\to\infty} q^{(t)}(i), \lim_{t\to\infty} p^{(t)}(j \to i)$ が存在し、両者は一致して 0 ではない[3]。これは $K+1$ 個の連立方程式 (極限方程式という) の解として得られる:

$$X_j = \Sigma_i X_i \cdot p(i \to j), j = 1,..,K$$
$$\Sigma_i X_i = 1$$

極限分布 $q^{(\infty)}(i)$ は変化を繰り返した後に、オブジェクト集合が到達するクラス分布を意味する。この極限分布は遷移の極限でもあるから独立事象であり、オブジェクト集合のクラス分布が時間に独立な '定常的な' 枠組みを有することを表す。究極的なスキーマかどうかの検証が計算可能であると言ってよい。スキーマ設計の立場からみれば (1) 普遍的な意味の検証、(2) 究極的な単純性・妥当性の判定、(3) 遷移の妥当性を計算でき、データベーススキーマの有効性を検証する場を提供する。

しかし、現実には常に時間 $t < \infty$ でのデータを扱うため、$q^{(t)}(i) = q^{(\infty)}(i) + v^{(t)}(i)$ における変動部 $v^{(t)}(i)$ が存在する。従って極限分布との差には、(1) 時間に依存する変動誤差 (遷移部)、(2) $q^{(t)}(i \to j) > 0$ が仮定できない (遷移しない) ことによる誤差、(3) 単純マルコフ遷移という仮定による誤差、(4) 確率を扱う故の必然的な誤差などが含まれる。(4) は確率論に基づく必然的なものであり避けられないが、モデル化の過程でこれらを分離する必要がある。

遷移確率の極限分布の存在を保証するには、条件 $p^{(t)}(i \to j) > 0$ が必要であるが、(遷移部分を含めた) 算出には $\mathcal{Z}$ 変換を利用した方が手っ取り早い [4]。関数 $f(t)$ に対して、時間 t で値 $f(0), f(1), ..$ が与えられるとき、$F(z) = \Sigma_{i=0}^{\infty} f(t) z^t$ を f の $\mathcal{Z}$ 変換という。この変換は $|z| < 1$ のとき、一意的に逆変換が存在する。極限方程式を書き直しベクトル係数の $\mathcal{Z}$ 変換式を $F(z)$ とすると、$z^{-1}(F(z) - \vec{q}) = F(z) \cdot P$ を満たし、これより $F(z)$ は一般に $\vec{q} \cdot (\vec{I} - zP)^{-1}$ と表せる。逆行列 $(\vec{I} - zP)^{-1}$ は必ず存在するので、t 次の係数を求めれば定常部と変動部を得ることができる☆: $q^{(t)}(i) = q^{(\infty)}(i) + v^{(t)}(i)$。このように $\mathcal{Z}$ 変換手法を用いることで、極限分布誤差に関するはじめの 2 つの問題に対応できる。

例題 2 P としてつぎの遷移行列を考える: $\begin{pmatrix} 3/4 & 1/4 \\ 0 & 1 \end{pmatrix}$

この結果 $\vec{I} - z \cdot P$ は次の式になる:

$$\begin{pmatrix} 1-(3/4)z & -(1/4)z \\ 0 & 1-z \end{pmatrix}$$

この逆行列 $(\vec{I} - z \cdot P)^{-1}$ は次式で得られる:

$$(1/1-z)\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} + (4/4-3z)\begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}$$

時間 t での項 $\vec{q}^{(t)}$ は、この式を級数展開すれば次のようになる:

$$\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} + (3/4)^t \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}$$

□

5.2 定常スキーマの評価

ここでは定常スキーマの設計を論じる。本稿ではオブジェクトの生死を仮定しないため、オブジェクト数 N は一定である。従って、例えば定常スキーマにおけるクラス c_i のオブジェクト数 $n_i^{(\infty)}$ は $q^{(\infty)}(i) \times N$ で計算できる。また遷移要素数 $n_{j,i}^{(\infty)}$ は $p^{(\infty)}(j \to i)^2 \times N$ に等しい。

☆ $p^{(t)}(i \to j) > 0$ を仮定しなくても $F(z)$ を計算できるが、この仮定の下では $\lim q^{(t)}(i) = 0$ となり、完全エルゴード的といわれる。

定常スキーマから、クラススキーマと各クラス c_i のオブジェクト集合サイズ $n_i^{(\infty)}$ を得ることができる。従って定常スキーマを定量的に評価 (単純性・妥当性の視点から) できる。

定常スキーマに従うデータベースでは、時間 t 後にクラス c_i に遷移するオブジェクト数が c_i のオブジェクト数と常に一致するという、特殊な性質を有する。しかも時間を経るにつれて現データベースが定常スキーマに収束することが期待できるという意味で、極限的な役割を担う。従って、定常スキーマでのクラスの定性的な分析 (決定木による属性値によるクラス記述) や定量評価 (単純性・妥当性) は、現時点でのスキーマ評価を大きな単位で捕らえている。

極限分布に対するエントロピは次で得られる:

$$\log_2 N - (1/N)\Sigma_i n_i^{(\infty)} \log_2 n_i^{(\infty)}$$

この極限分布の考え方を用いれば、決定木による帰納分類技法がそのまま定常スキーマに対しても考察できる。極限決定木が存在すればスキーマを定性的に表現でき、クラスの合併や分割が考察対象になる。

決定木の生成過程で考察されるすべての確率は要素の発生頻度として計算されている。しかし、個々のオブジェクトの極限状態を計算するものではないから、厳密な意味での極限決定木を生成することはできない。そこで、帰納分類で生成された決定木を参考に、要素の発生頻度を極限分布で比例配分し直し 決定木の構成時に作成された要素数 (従ってエントロピを) 再計算するという方法をとる。

例えば条件 $A =$ "a" を満たすオブジェクト集合 W の要素数 n, W の要素でクラス c_i の要素数を n_i とする。このとき、極限分布での要素数を $n^{(\infty)}, n_i^{(\infty)}$ と表せば $n_i^{(\infty)}/n_i^{(\infty)}$ を $(n_i/n) \times (N_i^{(\infty)}/N_i)$ と定義する。ここで $N_i^{(\infty)}, N_i$ は クラス c_i の極限分布および現在のデータベース中での総オブジェクト数を示す

定常スキーマが望ましいものでないならば、クラス設計の見直し過程で帰納分類で生成された決定木を参考に、合併や分割が考察対象になる。(クラス c_j に限らず) 決定木の生成過程で考察されるすべての確率は要素の発生頻度として計算されているから、極限方程式を再計算せずに対応できる。

例題 3 先の例では、第 1 項が定常部分第 2 項は時間 t に依存する変動部を表す ただ、この例では極限確率は存在するが c_1 の定常確率は 0 となっており最終的には消去できるクラスであることを示しているこの決定木は自明である

□

6. 関連研究

機械学習から帰納推論はデータベースからの知識発見 (KDD) と強い関連を有する [2] が、スキーマ情報だけを利用しインスタンスや背景知識と関係を持たない KDD は現在最も精力的な研究テーマの一つである [7] が、知識獲得の 1 分野として捕らえられデータベースのスキーマまたはインスタンスから背景知識を得る点で特徴的である。

ただ、この視点からの時制オブジェクトに関する研究はほとんど無い。伝統的に時系列解析は経済学分野で議論されてきたが 非数値領域に関する研究は少なくまた統計的な解析結果を解釈することが容易ではないため、関連をつかめないでいた 機械学習を時系列データ分野に応用する試みはうまく行っているようで、この分野の発展は有望である [1]。時制データのための決定木の差分的な修正については研究がある [19] が、構造の変更を少なくすることが狙いである。エントロピの差分計算を目的とするものは筆者らによる [12],[13]

7. 結　び

本稿では、機械学習手法に基づいて時制データに対してマルコフ性を仮定し 振舞い (トランザクション) に対して帰納分類手法を提案した。特に定常スキーマの計算では、トランザクションの帰納分類と一致し 同時に設計方法を論じることになることを示した 具体的には極限分布を生成して定常スキーマの生成・評価を行う手法を提案した

本方式の問題はマルコフ記憶の深さの効率良い見直しにある。本文では単純マルコフ遷移を仮定したが一般マルコフ遷移にすることは容易である ただ、現状のデータベースの遷移状況を観察し、マルコフ記憶の長さを検出するための効果的方法を構築することが急務である 記憶の深さの変更では再計算が必要であるが差分計算可能かどうかを含め現在この方法の有効性を検証している。

謝辞 本稿の考察にあたり貴重なコメントを頂いた Mike Boronowsky 博士 (ブレーメン大学) に感謝します。

参考文献

1) Boronowsky, M.: Automatic Measurement Interpretation of a Physical System with Decision Tree Induction, *Conference of IDEAL* (1998)
2) Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. (Eds).: Advances in Knowledge Discovery and Data Mining, *MIT Press* (1996)
3) フェラー, W.: 確率論とその応用 1 (上下)、現代経営科学全集 5, 紀伊国屋 (1960)
4) ハワード, R.A.: ダイナミックプログラミングとマルコフ過程、培風館 (1971)
5) 石塚: 知識の表現と高速推論, 丸善出版 (1995)
6) Langley, P.: Machine Learning and Concept Formation, *Machine Learning* 3 (1989)
7) Mannila, H. Methods and Problems in Data Mining, *Intn'l Conf. on Database Theory* (ICDT) (1997)
8) 三浦, 塩谷: データベースにおける型スキーマの発見, 情報処理学会論文誌 38-6 (1997)
9) Miura, T. and Shioya, I.: Paradigm for Scheme Discovery, Intn'l Symposium on Cooperative Database Systems for Advanced Applications *(CODAS)* (1996)
10) Miura, T. and Shioya, I. Learning Concepts from Databases, *Conference and Workshop of DEXA* (1998)
11) 三浦、塩谷：スキーマ発見のための型近似、情報処理学会論文誌 39-4 (1998)
12) Miura,T. and Shioya, I.: Inductive Classification of Temporal Objects, to appear in *PACRIM* (1999)
13) Miura,T. and Shioya, I.: Incremental Update of Decision Trees for Temporal Objects, to appear in *KRDB* (1999)
14) Quinlan, J.R: Induction of Decision Trees, *Machine Learning* 1-1 (1986)
15) Quinlan, J.R.: C4.5 - Programs for Machine Learnings, Morgan Kauffman (1993)
16) Sakai, H. et al.: A Method for Behavior Modeling in Data Oriented Approach to Systems Design, *ICDE* (1984)
17) Piatetsky-Shapiro, G. and Frawley, W.J. (ed.): Knowledge Discovery in Databases, *MIT Press* (1991)
18) 滝：情報論 I, 岩波全書 306 (1978)
19) Utgoff, P.E.: Incremental Induction of Decision Trees, Machine Learning 4-2 (1989)
20) Wu, X.: Knowledge Acquisition from Databases, Ablex Publishing (1995)