データベースシステム 116－33
（1998. 7. 9）

データベースからの概念学習

三浦孝夫
法政大学工学部
miurat@k.hosei.ac.jp

塩谷勇
産能大学経営情報学部
shioya@sanno.ac.jp

本稿では，データベース中のオブジェクト集合が意味する概念を生成するため，データベーススキーマと実現値集合を背景知識として利用した記述の生成方法を論じる．スキーマはデータベースにおける共通知識を表しているが，設計された時期と情報の格納時期とが一致しないことから，必ずしも意図するものとなっているとは限らない．これを補うため，本稿ではオブジェクト集合が備える特徴項目値を活かし，スキーマ定義と連動して，集合の表す概念の記述方法を提案する．具体的には，決定木手法を拡張して型の条件付き積和表現を定義し，その有効性を示す．
キーワード: 知識発見, データマイニング

Learning Concepts from Databases

Takao MIURA
Hosei University

Isamu SHIOYA
SANNO College

In this investigation, we discuss how to make *conceptual formation* from databases as background knowledge. More specifically given a set E of objects $e_1,..,e_n$, we generate *conceptual description* to capture the meaning that E has. Database schemata contain common knowledge that are carried by all the instances. However, they had been designed much before various instances are generated, then the schemata sometimes couldn't capture the exact meaning of the instances.

To adjust such conceptual differences, we utilize relevant information (attributes, attribute values) to make suitable conceptual description by means of database schemata and their expressions. In this work, we introduce *Conditional Sum of Product* (CSOP) expressions on object types and discuss how to generate conceptual description based on extended *decision tree* technique. Also, we show some experimental results.
Keywords: Knowledge Discovry, Data Mining

1 前書き

本稿では，データベース中のオブジェクト集合が何かを意味するとき，この概念をデータベースの有する知識とそれらの表現言語で記述するための方法を論じる．与えられた事例集合の意味を特定する手法は，知識獲得や概念学習としてこれまで研究されているが，本研究はデータベースを背景知識として利用する点で特徴的である．

データベースは，これまで情報の格納場所あるいはその管理方式として扱われ，本来それが有する意味を積極的には活用されていかなった．ここでいう'それが有する意味'とは，データベースがスキーマという情報の分類基準を有しそれに従って無矛盾にデータが保持されていることを表す．

データベース研究の立場からは，この情報を知識記述あるいは操作に活用しないのはむしろ奇異でさえある．データベースが膨大で無矛盾な情報を懸命になって管理しているものを，例えば概念学習の過程で利用しない手はない．候補となる概念を生成し検証，特定するのにこれらの情報はふさわしいからである．

データベースでは，すべてのオブジェクトは型と呼ばれる分類基準に従って格納されている．従って集合の意味するものを，型の表現式で構成すると考えるのは自然であろう．更に，表現式の意味(解釈)が当該オブジェクト集合をうまく記述するか，つまり表現される概念が良好な近似であるかどうかも，データベース自身を使って検証できる．本稿では，概念生成を近似プロセスとみる．更に，データベースに検証させる過程とともにこれをデータベースからの概念学習と呼ぶ．

データベースからの概念学習を考察するために解決すべき問題は，概念を記述する言語としてどのようなものが望ましいか，さらにその表現式をどのように生成すればよいかにある．本稿では，型から構成される表現，特に条件付き積和表現(CSOP 式)を，決定木手法に基づいて生成するアルゴリズムを提案し，その有効性を論じる[1]．

以下では，第 2 節でデータベースを知識と見なすことの合理性や方法を論じる．3 節ではデータベースや概念階層を定義した後，概念記述として型による積和表現の生成とその限界を示す．第 4 節では，本稿で提案する型の CSOP 式による概念記述の有用性とその生成方法を示す．5 節は実験的な結果を用いてこの方法を評価する．第 6 節では関連する研究を示す．

2 知識としてのデータベース

この節では，問題の位置付けを明確にするため，前提となる状況や本研究で達成すべきゴールを論じる．

2.1 概念の学習

本稿では，与えられたオブジェクトの集合 E が何を意味するかという問いに，データベースからの概念学習手法を用いて応える方法，言い替えると E の特徴抽出方法を論じる．E はデータベースの一部であり，E の有する新しい意味を，E だけでなくてデータベースの知識も利用する点が特徴的である．

オブジェクト集合 E には付加的な情報が備わっていると認識されていることがある．予め想定された特性項目(特徴項目という)が与えられた各オブジェクトはその情報に従って概念抽出される場合を例による学習，特徴項目が動的に決定される場合を観察による学習という．

結果がどのようなものかが予想できるとき，その処理を概念分析と呼ぶ．概念分析では，結果候補が備えるべき性質や特徴項目を用いて E の各オブジェクトを何かの方法で決定する方法を分類，特徴項目を想定せずに E の結果を決定する方法をクラスタ化という．前者の典型例が決定木，後者では統計手法に基づく因子分析が知られる．このような結果が予想できないとき概念生成という [13]．このときも同様に，本来オブジェクトが何かの特徴項目を有すると仮定して E の意味を決定する方法を概念達成，そうでない場合を概念形成という．本稿ではデータベースを背景知識とした概念達成を論じる．

2.2 背景知識としてのデータベース

データベースとは，予め与えられた分類体系に基づいて，抽象化され格納されたデータの集合である．スキーマは対象となる用途から抽出された知識であり，すべての実現値は個別の事実を表すと同時に，共通してスキーマの有する意味に従う．

データベースはその歴史的な出現理由から，データ独立性を達成するために，宣言的な記述に基づく．スキーマは型集合 T と型付き述語記号の集合 P で記述され，個々のデータは基礎項または基礎式で表現する．単純な多類論理と異なり，基礎項は複数の(しかし有限な)型を有してよいとする．一貫性制約を述語論理の閉式集合 IC で表し，$\langle T, P, IC\rangle$ をスキーマシステムとよぶ．データベース D は IC を充足する T, P の解釈とみなす．

データベースからの概念学習では D をその解釈のままに(新たな)スキーマシステムを生成する過程であり，(帰納的論理プログラミングのように)モデル計算をするのではない．データ操作(質問と呼ぶ)も論理式で記述され，D という解釈を用いた述語計算を意味とする．

2.3 データベースからの概念学習

ここでは，本稿で扱う問題を定式化する．データベースの記述に従い，E を基礎項の有限集合で表す: $E = \{e_1, .., e_n\}, \infty > n > 0$ ．E の要素 e_i はデータベースに格納されている情報であり，E を構成することによって生じる新しい意味を探ることが目的である．E の持つ意味とは，E を的確に表す論理的な記述，つまりスキーマシステムに基づく'表現式' κ をいう．式 κ はスキーマから(ある方法で)構成される表現であり，その解釈 $[\kappa]$ は E に一致するのが望ましいが，$[\kappa] \supseteq E$ でもよい．このとき κ は E を記述するという．両者の差が大きいときは，κ は E を記述するのに一般的すぎて有用とはならない．本稿では両者の要素数の比 $|E|/|[\kappa]|$ を記述度 (*degree of description*) と定義し，結果の評価に用いるものとする．記述度を 1 に近づけるほど正確に E を表す．反面そのために複雑で長大な式を要し，理解しやすさが失われがちになる．

表現式を得るためにスキーマを基本として用いる．概念学習を支援しやすい記述方法(言語)としては，閉論理式や開論理式(質問式)などの他に，部分クラスで直観的な意図を表した言語もある．

データベース分野では，既にこれと類似した目的を持つ研究が知られる [23]．内包解 (Intensional Answers) は，質問に対して大量の解の概念的な意図を理解し易い形で表

[1] この論文の完全な版は [21] を参照．

した記述であり，通常(スキーマに基づく)新たな質問式を生成する．

この内包解技法を概念学習手法として直接用いることができる．実際 E を質問 Q の解答とみなし，Q からより明確な意図を表す Q' に書き換えると考えれば良い．しかし，質問式を生成する過程では，E から Q に変換する必要があり，この変換過程や質問言語の記法，表現力に依存して概念学習が進行することになり，結果が想定されるものの組み合わせになりがちである．本研究では，E をそのまま扱ったままで κ を生成する効果的で効率良い二つの方法を提案する．

3 データベースからの概念生成

この節では，データベースからの概念生成を論じるため，初めに背景知識となるデータベースと本稿で扱う問題の正確な定義を与える．次に，問題を明確にするため，特徴項目を何も仮定しないとき型の積和表現による概念生成について論じる．

3.1 データベースと型階層

データモデルは，典型的にはオブジェクト (*object*) または実体 (*entity*) と連想 (*association*) に基づいて定義される [6]．前者は対象世界の'もの'を代表する概念であり，表現方法，属性などと独立に定義される概念である．後者はオブジェクトの間の結びつきを表す概念であり，ひとつの結びつきはひとつの連想と対応する．オブジェクトの型 (*type*) は，オブジェクト集合に共通の性質を捉える内包概念であり，このときオブジェクト e は型 t を持つと言う．

一般的に e は t 以外にも型を持つことがある．以下ではオブジェクト集合(有限とする)を $\mathcal{E}$，型集合(有限とする)を $\mathcal{T}$，型 t を持つオブジェクトの集合を $\Gamma(t)$ と表す．これを型 t の解釈という．すべてのオブジェクトを表す型が存在しないときは，これに代わる仮想的型 $\mathbf{OE}$ を考える．型 t_1, t_2 に対して $|\Gamma(t_1)| > |\Gamma(t_2)|$ となるとき，t_1 は t_2 よりも大きいという．他の多くの型に対して大きい型は，大多数のオブジェクトを要素に持つから，よく利用されている型であるといえる．型スキーマを修正して，よく利用される型に集約する発見的手順を既に提案している [16]．

連想に関しても同様に，型やオブジェクトに対応して，それぞれ型付けられている述語 (*predicates*) や連想 $p(e_1 \ldots e_n)$ を考えることができる．特に，E_1, E_2 上の述語 $p(E_1 E_2)$ で E_1 から E_2 への関数対応をとるとき，E_2 を E_1 の属性 (attribute) と呼ぶ．連想についてはこれ以上論じない．詳しくは [15] を参照されたい．

各オブジェクトが型を保持するとき，実現値に基づく (*instance-based*) と呼び，意味モデルなど多くの例がある [9, 6]．オブジェクト e に対してそれが有する型を $\tau(e)$ で表す: $\tau(e) = \{t \in \mathcal{T} \mid e \in \Gamma(t)\}$．これを e の型スキーマ (*type schema*) と呼ぶ．$\tau(e)$ は有限であるとする．すべての $\tau(e)$ はデータベースの型付け情報を表すものとなり，これをデータベースの型スキーマと呼ぶ．

データベースが常に正しい意味を捉えるには対象世界の変化に追随する必要があるため，型スキーマは変化せざるを得ず，オブジェクトの記述や操作は複雑になるであろう．

例題 1 ここでは共通例としてアジアの国々に関する情報を用いる [18]．アジア地域を記述するために 8 個の型を導入する: Country (C), FarEastAsia (FE), WestAsia (W), SouthEastAsia (SE), Asia (A), Oceania (O), Pacific (P), Kanji (K) がそれぞれ表す意味は明きらかであろう．

ここではアジア地域の 14 の国で扱うを C 以外の 7 つの型で特徴付ける．

object	*typeschema*
$Japan(J)$	FE, A, P, K
$Korea(K)$	FE, A, P, K
$Taiwan(TW)$	FE, A, P, K
$China(C)$	A, K
$HongKong(H)$	SE, A, P, K
$Philippines(PH)$	SE, A, P
$Singapore(S)$	SE, A, P
$Malaysia(M)$	SE, A, P
$Indonesia(IS)$	SE, A, P
$Thailand(TH)$	SE, A
$Australia(A)$	O, P
$NewZealand(NZ)$	O, P
$India(IN)$	W, A
$Pakistan(PK)$	W, A

□

型を制御する典型的な機構のひとつが汎化 (*generalization*) であり，これは ISA ともよばれる．型 t_1，t_2 に対して，t_2 が t_1 の汎化(または t_1 ISA t_2) とは，$\Gamma(t_1) \subseteq \Gamma(t_2)$ が制約条件として与えられているときを言う．また t_1 ISA t_2 が成り立つとはデータベースにおいて当該制約条件が充足されるときをいう．ISA を複数回適用して得る反射推移閉包を ISA 階層 (または型階層) という．この階層によって ISA を正しく推論することができる，つまり t_1 ISA t_2 と t_2 ISA t_3 から t_1 ISA t_3 を導くことができる．型 t_1，t_2 に対して t_2 が t_1 の直接汎化 (*direct generalization*) または直接 ISA，直接親とは，t_1 ISA t_2 が成り立ち，かつ t_1 ISA t_3 で t_3 ISA t_2 となる t_3 が (ISA 階層に) 存在しないことをいう．同様に直接子も定義できる．

t が 2 つの型 t_1, t_2 の祖先であるとき，t がその近汎 (least general) 型であるとは，t_1, t_2 の祖先 t' で t' ISA t ならば t' と t は同一となるときを言う．3 つ以上の型に関する近汎型も同様に定義できる．型集合 $V = \{t_1, .., t_k\}$ とするとき $LG(V)$ によって $t_1, .., t_k$ のすべての近汎型集合を表す．同様に集合 $LF(V)$ を V の要素の祖先になる型をすべて取り除いたものとする : $LF(V) = \{t \in V \mid t' \text{ISA} t \text{ となる } t' \in V \text{ はない}\}$．また $LGF(T) = LG(LF(T))$ と定義する．

オブジェクト e が型 t を持ち，t ISA t' ならば e は t' も有するに違いない．言い替えると，$t \in \tau(e)$ かつ t ISA t' から $t' \in \tau(e)$ が成り立たねばならない．これは型スキーマが矛盾を含まないための条件であり，型無矛盾性 (*type consistency*) と呼ぶ．以下では到るところで型スキーマの型無矛盾性が成り立つとする．

例題 2 例題 1 で用いる ISA 階層として，9 つの直接汎化から構成され，どの型も高々 2 つしか親を持たないものを用いる．図 1 はこの ISA 階層を示している．

各国 e に対して $LF(\tau(e))$ を計算する．このとき ISA 階層を用いる．同じ LF 集合となるものを集めて表示してある各オブジェクトは ISA 階層から計算できるすべての型を有していることに注意したい．

types	*objects*	Γ(*types*)
FE, K	J, TW, K	
K	C	C, J, TW, K, H
SE, P, K	H	
SE, P	PH, S, M, IS	
SE	TH	TH, PH, S, M, IS, H
O	A, NZ	
W	IN, PK	
P		$J, TW, H, PH, S, M, IS, A, NZ$

□

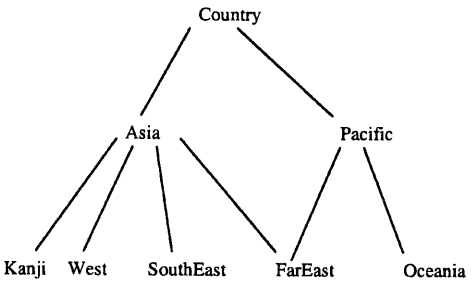

図 1: Asia 型の国の ISA 階層

3.2 型の積和表現による概念生成

初めに，最も単純な表現方法として，オブジェクト集合 $E = \{e_1, .., e_n\}$ を解釈とするスキーマ表現 κ の型だけを用い，特に積和 (SumOfProduct, SOP) 表現とする概念生成方法を考える．型 $t_1, .., t_m$ に対して $t_1 \wedge ... \wedge t_m$ の形をその積 (Product, P) 表現式という．型の解釈 $\Gamma(t_i)$ を用いて積表現式の解釈を $\bigcap_{i=1}^{m} \Gamma(t_i)$ と定義する．集合 $c = \{t_1, .., t_m\}$ に対して $\wedge c$ を $t_1 \wedge ... \wedge t_m$ とする．積表現式 $c_1, .., c_m$ に対して $c_1 \vee ... \vee c_m$ の形をその積和表現式という．積表現式の解釈 $\Gamma(c_i)$ を用いて $\bigcup_{i=1}^{m} \Gamma(c_i)$ を解釈とする．集合 $c = \{c_1, .., c_m\}$ に対して $\vee c$ を $c_1 \vee ... \vee c_m$ とする．これらは型概念の交差および並存に対応しており，直観的な意味を捉え易いと考えて良い．以下では，積表現式または積和表現式のみを扱い，'表現の単純さ' は記述の長さに反比例すると考える．

本稿で扱うデータベースでは，E の各要素 e_i の型スキーマ $\tau(e_i)$ は，型の無矛盾性故に e_i が持つすべての型を含んでいる．生成すべき表現 κ の解釈 $[\kappa]$ は従って型解釈 $\Gamma(\kappa)$ と定義され，$\Gamma(\kappa) \supseteq E$ を充たしつつ，記述度が 1.0 に近くてしかも単純な記述となるものを求めたい．

オブジェクト e_i に対して，積表現 $\tau_i = \wedge\tau(e_i)$ は $\bigcap_{t \in \tau(e_i)} \Gamma(t)$ を解釈に持ち，しかも e_i を含むから空ではないから，e_i を記述する候補として最も精密である．しかし τ_i が他のオブジェクト e_j を解釈に含むとは限らず，一般には $\tau_1 \wedge ... \wedge \tau_n$ は E を記述できない．

これに対して $\kappa_1 = \tau_1 \vee ... \vee \tau_n$ は E を記述し，しかも各 τ_i が e_i を精密に表すから，よい記述度になるであろう．しかし，定義から τ_i は長くなり，単純さに欠ける．なぜなら，型 t が $\tau(e_i)$ に含まれていれば t ISA t' となる t' も含まれるから．$\Gamma(\tau(e_i))$ が $\Gamma(LF(\tau(e_i)))$ と等しいから，$\tau(e_i)$ の替わりに $LF(\tau(e_i))$ を用いればよい．実際，$\kappa_2 = (\wedge LF((\tau(e_1))) \vee ... \vee (\wedge LF(\tau(e_n)))$ は κ_1 と比べて単純な記述となる．

積表現だけで E を記述できないであろうか？ $\tau(e_1) \cap ... \cap \tau(e_n)$ の要素が共通して有するすべての型であり，@E がどの型スキーマにも含まれるから，空ではない．記述を短くするために $\kappa_3 = LF(\tau(e_1) \cap ... \cap \tau(e_n))$ と定義すれば，$\wedge\kappa_3$ は E を記述する積表現となる．しかも κ_2 と比べて単純な記述を得る可能性が高い．しかし E の要素が共通して有する型は @E のように汎用的なものであることが多く，また要素の特性を無視することになるため，記述度 $|E|/|\Gamma(\kappa_3)|$ は低いであろう．つまり単純だが余りに一般的な表現となる可能性が高い．

各 $\tau(e_i)$ を十分単純に表現し，しかもそれらの和表現で記述することができないだろうか？ $\Gamma(LF(\tau(e_i)))$ の記述のために $\wedge LGF(\tau(e_i))$ を用いるというアイデアは，記述度を低下させるが単純さを増加させる積和表現を生む．実際，$LGF(\tau(e_i))$ の要素は $LF(\tau(e_i))$ の近汎型であり，適当な型階層では $\wedge LGF(\tau(e_i))$ は単純となる．$\kappa_4 = (\wedge LGF(\tau(e_1))) \vee ... \vee (\wedge LGF(\tau(e_n)))$ が候補となる記述である．

例題 3 ここでは例題 1を用いて，概念記述をどのように行うかを示す．E_1 を Japan, HongKong, China からなる集合 $\{J, HK, C\}$ とし，これを概念的に記述することを考える．定義から，κ_1 は $(FE \wedge A \wedge P \wedge K \wedge C) \vee (SE \wedge A \wedge P \wedge K \wedge C) \vee (A \wedge K \wedge C)$ であるが，簡単な計算でこれは K と同じ解釈になることがわかる．一方，κ_2 はこれより短い記述となり $(FE \wedge K) \vee (SE \wedge P \wedge K) \vee (K)$ となる．これも K と同じ解釈となる．

他方，κ_3 は K そのものである．κ_4 は $A \vee C \vee K$ であり，これは C と同じ解釈となる．

どのケースであっても，記述は型 Kanji を持つ国となり，参考となる型であることがわかる．但し，必ず Korea, Taiwan を誤りとして含む．

他の例として E_2 を Japan, Australia, India からなる集合 $\{J, A, IN\}$ とし，概念記述を行う．定義から，κ_1 は $(FE \wedge A \wedge P \wedge K \wedge C) \vee (O \wedge P \wedge C) \vee (W \wedge A \wedge C)$ であり，κ_2 はこれより短い $(FE \wedge K) \vee (O) \vee (W)$ となる．κ_3 は C そのものである．κ_4 は $A \vee O \vee W$ であり，これは $A \vee O$ の解釈と一致する．

いずれもケースも，生成される記述はあまり有用なものではなく，例えば最初のふたつは解釈することが困難でさえある．E_2 に関してはデータベースからは有用な知識を得られない．□

3.3 考察

上で示した E の意味記述の有効性を考えたとき，型の積和表現で得る記述は，E の意味を十分に反映したものとなっているだろうか？

型スキーマを手がかりとして，型の積和表現に基づく意味記述を得る手順は直観的かつ概念的で，利用者にとっても結果の妥当性を検証し易い．

しかし，E の各要素が有する型情報と背景知識となる型階層だけを用いて，その意味を抽出しようとすることが果たして可能であろうか？ データベーススキーマが個別情報の変化を捉えきれず，格納情報が (スキーマよりも) 精密な意味を表しているという状況では，型スキーマだけを手がかりとすることに問題があると考える方が自然であろう．問題は，個別情報の検査結果を，どのようにして型表現に変換するかという点にある．

4 特徴値を用いた概念生成

本節では，E の各要素が共通して有する特徴項目 $A_1, .., A_m$ を仮定し，その値の持つ意味を分析することによって E の意味する概念を生成する手法を探る．

4.1 特徴値と特徴項目

E の各要素が共通して特徴項目 $A_1,..,A_m$ を有する状況は，それほど珍しいことではない．例えば，型 $E, A_1,..,A_m$ 上で定義された述語 $p(EA_1...A_m)$ は，$\langle e, a_1,..,a_m\rangle$ の形の基礎式を連想に対応させるが，e からの相対的な視点に立てば e には A_i に関して値 a_i となる組 $\langle a_1,..,a_m\rangle$ が一つ以上対応する．別の例として，型 $t, t_1,..,t_m$ かつ t **ISA** $t_i, i=1,..,m$ のとき，t_i の属性 A_i は t の要素に継承される．従って，$E \subseteq \Gamma(t)$ となる E は共通して特徴項目 $A_1,..,A_m$ を有する．

4.2 条件付き積和表現

以下で示す方法は，分類手法として知られる決定木を参考にして，特徴項目値の分類を概念生成に利用する．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手続きであり，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．**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，エントロピー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 [25, 26].

決定木では根から末端節への経路は，特徴項目に課せられたを連言条件を表している．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，条件 σ として $A=$"a" を生成すると考えれば，経路は連言条件 α を表す．ここでα を$\sigma_1 \wedge ... \wedge \sigma_k$ と定義する．末端節に付けられた結果ラベルが c であれば，これを $\alpha : c$ と表す．

決定木の構成方法から，各分岐は事例集合を分割し，しかもすべての事例が生成された CSOP の対象となる．CSOP 式の解釈 は $[\alpha_1 : c_1] \cup ... \cup [\alpha_m : c_m]$ で定義される．ここで $[\alpha : c]$ とは事例集合の要素で条件 α を充たすものの集まりをいう．本稿ではこれを更に拡大し，c を集合であっても良いとする．

4.3 概念生成の手順

本稿で提案する手続きを理解しやすくするために，はじめにどの $\tau(e_i)$ も単一集合 $\{c_i\}$ とする．このとき，E の各要素 e_i は特徴値と c_i で記述できる:

$$E = \begin{pmatrix} a_1^1 & ... & a_m^1 & c_1 \\ ... & ... & ... & .. \\ a_1^i & ... & a_m^i & c_i \\ ... & ... & ... & .. \\ a_1^n & ... & a_m^n & c_n \end{pmatrix}$$

$c_1,..,c_n$ がすべて同一の型 c であれば，汎化度 (*degree of generality*) 1.0 で E を型 c で表現できるという[2]．$T_c = \{d_1,..,d_k\}$ を $c_1,..,c_n$ の近汎型集合 $LG(\{c_1,..,c_n\})$ とする．T_c の汎化度がしきい値 ρ 以上ならば，同様に E を型 T_c で表現できるという．ただし，T_c の汎化度を $|\bigcup_{i=1,..,n}\Gamma(c_i)|\,/\,|\bigcap_{j=1,..,k}\Gamma(d_j)|$ と定義する．T_c の汎化度は 1.0 以下である．いずれでもないとき，ある項目，例えば A_1 で E をグループ化し，各グループに対して (A_1 以外の特徴項目を用いて) 同様の処理を行う．特徴項目がもはやなくなれば，E を型 c で表現できないという．

特徴項目の選定基準を考える．項目を選定する各段階で決定的に選定するために，情報の集合 E を対象となる特徴項目 A で分類して $E_1,..,E_j,..,E_l$ を生成し，E_j の汎化度 g_j と要素数 k_j の積の和 $g_1 \times k_1 + .. + g_l \times k_l$ を求める．これを E の A に関する被覆度 (coverage) という．

E にとって未処理である特徴項目の被覆度をすべて計算し，その最大の項目を選ぶものとする．最大のものを選ぶことで，損失を少なくする効果を得ることになる．

E が T_c で表現できたということは，どの過程も (汎化度のしきい値 ρ 以上で) E を被覆しており，型階層の特性 (例えば汎化を行った回数や '兄弟' 型の数) を用いていない．処理全体をみれば，E が生成された CSOP 式 $\alpha_1 : c_1 \cup ... \cup \alpha_m : c_m$ によって排他的に分割され，各枝が積表現で '十分に' 特徴つけられるものとなっている．言い換えると，E は型の CSOP 式として記述されることになる．ここで，この CSOP 式の解釈 は $\Gamma(\alpha_1 : c_1) \cup ... \cup \Gamma(\alpha_m : c_m)$ で定義される．ここで $\Gamma(\alpha : c)$ は $\Gamma(c)$ の要素集合でα を充たすものをいう : $\Gamma(\alpha : c) = \{e \in \Gamma(c) \mid e$ はαを充たす $\}$.

これまでは$\tau(e_i)$ を単一要素としたが，集合でもよい．実際，$\tau(e_i)$ の替わりに $LF(\tau(e_i))$ を用いる．E の要素がすべて同一の型集合 c であれば，汎化度 1.0 で E を型集合 c で表現できるという．$T_c = \{d_1,..,d_k\}$ を $c_1,..,c_n$ の近汎型集合 $LG(c_1 \cup ... \cup c_n)$ とする．T_c の汎化度が ρ 以上ならば，E を型集合 T_c で表現できるという．ただし，T_c の汎化度を $|\bigcup_{i=1,..,n}\Gamma(c_i)|\,/\,|\bigcap_{j=1,..,k}\Gamma(d_j)|$ と定義する．T_c の汎化度は 1.0 以下であることに注意したい．

例題 4 アジアの国々の例題 1 を続ける．ここでは，更に主たる宗教と東京にある観光局の所在地という特徴項目を加える．

object	$R(religion)$	$L(office\ location)$
$Japan(J)$	B	*Ginza*
$Korea(K)$	F	*Ginza*
$Taiwan(TW)$	B	*UchiSaiwai*
$China(C)$	F	*ToranoMon*
$HongKong(H)$	B	*MarunoUchi*
$Philippines(PH)$	C	*NampeiDai*
$Singapore(S)$	C	*UchiSaiwai*
$Malaysia(M)$	M	*Ginza*
$Indonesia(IS)$	M	*Akasaka*
$Thailand(TH)$	B	*Ginza*
$Australia(A)$	C	*KioiCho*
$NewZealand(NZ)$	C	*Shinjuku*
$India(IN)$	H	*Ginza*
$Pakistan(PK)$	M	*Azabu*

ここで B は仏教，H はヒンドゥー教，M はイスラム教，F は儒教，C はキリスト教を表す．

このとき集合 $E_2 = \{J, IN, A\}$ は次の CSOP 表現を使って記述できる: $(L =$"$Ginza$"$\wedge R =$"B"$: \{FE, K\}) \vee (L =$"$Ginza$"$\wedge R =$"H"$: \{W\}) \vee (L =$"$KioiCho$"$: \{O\})$. 各項はそれぞれ J, IN, A に対応した記述になっている．集合 $E_3 = \{J, K, TH\}$ で十分に小さい ρ ならば，CSOP 表現 $(R =$"B"$: \{A\}) \vee (R =$"F"$: \{FE, K\})$ はこの集合を表し，最初の項が $\{J, TH\}$ を，後が K を意味している．

集合 E_4 を $\{J, K, C\}$, $\rho = 0.5$ としよう．このオブジェクトの型スキーマ (LF 集合) はすべて異なるものであり，また共通の近汎型は *Asia* である．これは 12 個の実現値を含む．

[2] 実際 c を c 自体で汎化している．

L 上でグループ化を行う．$L =$"$Ginza$" のグループは J, K で共通の型スキーマ $\{FE, K\}$ を持ち，汎化度も 1.0 となる．もうひとつのグループも同様であり，結果的に被覆度 3.0 を得る．R 上でグループを作るならば，$R =$"F" が K, C の 2 ケ国を含むが，それぞれの型スキーマが FE と K であることから，近汎型 $Asia$ に集約され，汎化度は $5/12.0 = 0.42$ になる．別のグループは J だけを含むので，全体としての被覆度は 1.83 になる．

このことから L 上でのグループ化を行い，対応する CSOP 表現 ($L =$"$Ginza$" : $\{FE, K\}$)$\vee$($L = ToranoMon$" : $\{K\}$) で E_4 を記述する．図 2 を参照．□

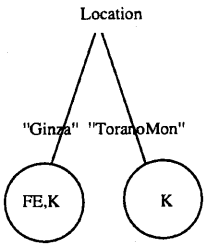

図 2: CSOP 表現

4.4 生成アルゴリズム

上に示したアルゴリズムを要約する．あらかじめ，各オブジェクトの型スキーマ $c_1, .., c_n$ はすべて $LF(c_1), .., LF(c_n)$ で入れ換えておく．

(1) $c_1, .., c_n$ がすべて同一の型集合 c であれば，E は表現可能であり，汎化度を 1.0 として終了する．
(2) そうでないとき $c_1, .., c_n$ の近汎型集合 $LG(\{c_1, .., c_n\})$ 計算し，$T_c = \{d_1, .., d_k\}$ とする．汎化度 $|\bigcup_{i=1,..,n} \Gamma(c_i)|/|\bigcap_{j=1,..,k} \Gamma(d_j)|$ を計算し，これが ρ 以上ならば終了する．
(3) いずれでもないとき，分割処理のために特徴項目 A を選定する．この項目で E をグループ化し，各グループに対して (A 以外の特徴項目を用いて) 再帰的にこの処理を行う．特徴項目がもはやなくなれば，E は表現できない．

特徴項目を選定するために，次の操作を行う．

(1) 対象となる各項目 A に対して情報の集合 E を A で分類し，$E_1, ..., E_l$ を生成する．E_j の汎化度 g_j と要素数 k_j から A に関する被覆度を計算する．E が分類できないときは E を表現することができない．
(2) (1) で得た値の内で最大の項目を選ぶ．

5 実験結果

本節では，CSOP アルゴリズムの実験結果を示す．実験に用いたデータは，日経バイトマガジンの 1994 年 2 月号に示された 1985 年から 10 年間の記事の索引情報 **Nikkei Byte Article Keyword Index** を用いている．この索引には 2315 件の記事に関する情報 (その内 2 件は誤りであり，キーワードが無いインデックスなどを除いた実際には 2009 件) が含まれ，各情報は表題，掲載刊号，ページ，主題，および幾つかのキーワードを含む．以下に記事の一部を示す．

キーワード数	記事数
0	304
1	1371
2	513
3	106
4	18
5	1
(合計)	2313

図 3: **NIKKEI** 索引概要

```
.....
(580) Demonstrating by IBM and MS - OS,Multimedia,
  Pen:1991.12:no.93:p.144:Trends:OS Trends:
  Multimedia:Pen computer
.....
(666) Page Printer on Windows - PART III Basic
  Benchmark - Graphics:1991.09:no.90:p.256:Topics:
  Windows:Printer:Benchmark test
(667) Page Printer on Windows - PART IV Application
  Benchmark:1991.09:no.90:p.274:Topics:Windows:
  Printer:Benchmark test
.....
```

例えば 666 番目の記事は表題が Page Printer ... で，掲載刊号 (no.90)，ページ (pp.256)，主題 (Topics)，キーワード (Windows, Printers, Benchmark Test) であることを表す．このとき，各索引は**@E** 型のオブジェクト (この場合 "記事" と対応する) と見なせるが，更にキーワードも型と考える．索引には 198 個のキーワードが定義されており，実際には 146 個のキーワードが現れ，52 個は現れない．索引全体で，延べ 2792 回生じる．

索引のキーワード上の **ISA** 階層は，キーワードの組情報 ⟨ キーワード, 親キーワード ⟩ の集合として記述する．階層には 288 個の直接汎化関係があり，各キーワードは最大でも 4 つの親を持つにすぎない．次は階層構造の一部である．

```
....
x86 MicroProcessor:MicroProcessor
A/D converter:Component
AI:@E
APL:language
AX:IBM compatible:Personal computer
Ada:language
Apple II:Apple product:Personal computer
Apple product:@E
B16/32:HITACHI product:Personal computer
BASIC:language
C/C++:language:Object oriented
....
```

図 3に記事のインデックスの記事数を示す．

実験では特徴項目として主題，年，第 1 および第 2 のキーワードを用いた．索引記事情報から **software** というキーワードを含む記事を学習するオブジェクト集合 E

として選んだ．この場合，233 個の記事が見つかる．例えば，記事番号 1044 は以下であり，特徴項目間はセミコロンで区切られている．

```
Byte network: 1989: Word processor/word processor
    software: Table calculator software
```

記事番号 1044 は以下のキーワードを持っている．

```
LAN: Word processor/Word processor software: Table
   calculator software
```

E に現れるキーワードは以下のものからなる．

```
Application linkage, Benchmark test,
Communication software, C/C++, Data exchange,
Database management software, DTP, Editor,
Extended Memory, Facsimile adapter,
FEP, File management software, File format,
Graphical software, Hypertext, Idea processor,
LAN, Macintosh, Music software, Mouse, OS/2,
Personal computer communication,
Table calculate software, Translation software,
Unify software, Utility software, Windows,
Word processor/Word processor software
```

$\rho = 0.5$ として実験を行なった．E の汎化度のしきい値が ρ よりも大きい必要があり，その結果，以下に示す E 中の 14 個のキーワードが削除された．

```
Application linkage, Benchmark test,
C/C++、 Data exchange, Editor,
Extended memory, Facsimile adapter,
FEP, File format, Hypertext,
Idea processor, Mouse, OS/2,
Translation software
```

他の 14 個のキーワードは E 中のオブジェクトに残っている．

```
Communication software, Database management software,
DTP, File management software, Graphic software,
LAN, Macintosh, Music software,
Personal computer software, Table calculate software,
Word processor/Word processor software,
Unify software, Utility software Windows
```

第 1 キーワードの各特徴項目値によって，E はグループ化され，それらの特徴項目値は次に示されている．

```
Communication software, Database,
Database management software, DTP,
Editor, Facsimile adapter, File format,
File management software, Graphics software,
Hypertext, LAN, Mouse, Macintosh,
Music software, Table calculate software,
Translation software, Unify software,
Utility software, Windows,
Word processor/Word processor software
```

特徴項目値 Database management software によってグループ化されたオブジェクト集合は次に示す特徴項目‘主題’によってグループ化される．

```
Byte network, Byte report, Byte seminar,
Introduction, Product file, Product report,
Review, Special issue, Survey, Trend,
U.S.Insight
```

特徴項目値 Byte network によってグループ化されたオブジェクト集合は特徴項目‘年’によって 6 グループ作られる．

1984, 1985, 1986, 1987, 1988, 1989

1989 によってグループ化されたオブジェクト集合は第 2 キーワードの特徴項目によって 3 個にグループ化される．

```
Database management software, Table calculate
software, Word processor/Word processor software
```

この実験によって，E はうまく記述できる．

第 2 の実験では $\rho = 0.1$ に設定した．2 つの異なるオブジェクトの同じ特徴項目 (特徴項目値) を持つオブジェクトの集合が存在する一方で，型が異なる．このため，E は記述できない．例えば，一つのオブジェクトが型 LAN を持ち，一方，Database management software は同じ特徴項目を持つ．

この実験を通じて，次の所見が得られた．

1. 型の最小汎化の処理過程で，最小なものとして根@E が得られる時がある．@E は型が有用でないことを意味し，記述の失敗である．一方，汎化度が低いと特徴項目の不足から記述の失敗となる．
2. 汎化度を高くするためには，Eのある型を削除した．CSOP 表現が短くなるというよりも，記述の度合いが小さくなる．逆に，汎化度が低いと良く E を近似することができる．しかし，時々記述の失敗に陥る．
3. オブジェクトの型を無視しないと，同じ特徴項目を持つが異なる型であるオブジェクトに遭遇する．このために E を記述できない場合がある．

6 関連研究

データベースからの概念学習は現在データベース分野で最も大きい研究テーマのひとつであり，知識獲得 (あるいは機械学習問題) とデータベース設計を結ぶ境界問題である．記述を分析することで新しい型を生成することは意味世界の記述の生成であり，知識発見処理をデータベースに適用した知識獲得技術は，現在多くの研究者の注目を浴びるものとなっている [1, 2, 3, 4, 7, 8, 10, 24, 27]．例えば，ヘルシンキ大学 H.Mannila 教授らのグループはこの問題に対して積極的な研究を展開している [14]．

過去データベース研究では，スキーマ進化に関する議論がなされてきたが，これらは主としてデータベースシステムにどのような柔軟な機構を必要とするかというトップダウン的な視点を有している (例えば [5])．筆者らの理解する限り，実現値からのフィードバックを捉え，現在のデータベースを記述するスキーマを得ようとする技術は知られていない．スキーマ発見はメタモデル操作のための一般的な枠組みとも捉えられる [12]．データベースに対する制約の質問を規則や制約発見と見なせば，これはスキーマ発見の知識獲得そのものである．

筆者らの研究は，一貫してスキーマと実現値集合から現状を記述するのにふさわしいスキーマ記述を得るものである．これまで型スキーマ [16, 17]，述語スキーマ [19]，複合オブジェクトスキーマ [20] が統一的なスキーマ発見のパラダイム [18] に基づいて展開されていることを明らかにしてきた．過去の知識獲得技法と異なりデータベース的な問題とする根拠は，‘大部分のデータベースは効果的であ

るが，精密にみれば問題が生じており新たなスキーマの発見が差分的に決定できるに違いない'，という視点による．本研究では新たな知識獲得の枠組みを提案したが，依然としてこの観点からデータベース的な問題と考えている．

7 結論

本研究では，データベースから新たな概念を生成するための手法を提案した．得られた結果はデータベース再構築へのフィードバック情報として重要な役割を果たす．この目的を達成するため，型の CSOP 式を導入し，データベース知識 (ISA 階層) を用い，決定木に基づいた概念生成の方法を提案した．提案する方法は発見的ではあるが，効果的であることを実験を通じて明らかにすることができた．

謝辞 本研究に対して激励と貴重なコメントを頂いた上林弥彦教授 (京都大学), 増永良文教授 (図書館情報大学) に感謝します．なお，本研究は文部省科学研究費補助金 (重点領域研究高度データベース，課題番号 09230219) より一部援助を頂いた．

参考文献

[1] Agrawal, R., Ghosh, S. et al.: An Interval Classifier for Database Mining Applications, *Very Large DataBase* (VLDB), pp.560-573 (1992)

[2] Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules, *Very Large Database* (VLDB), pp.487-499 (1994)

[3] Cai, Y., Cercone, N. and Han, J.: An Attribute-Oriented Approach for Learning Classification Rules from Relational Databases, *Intn'l Conf. on Data Eng.* (ICDE), pp.281-288 (1990)

[4] Cai, Y., Cercone, N. and Han, J.: An Attribute-Oriented Induction in Relational Databases, in [27], pp.213-228

[5] Elmasri, R. and Navathe, S.B.: Fundamentals of Database Systems, *Benjamin* (1989)

[6] Embley, D. et al.: Object Oriented System Analysis, Yourdon Press (1992)

[7] Frawley, W.J., Piatetsky-Shapiro, G. and Matheus, C.J.: Knowledge Discovery in Databases – An Overview, in [27], pp.1-30

[8] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. (Eds): Advances in Knowledge Discovery and Data Mining, *MIT Press* (1996)

[9] Hull, R. and King, R.: Semantic Data Modeling, *ACM Computing Surveys* 19-3, ACM, pp.201-260 (1987)

[10] Han, J., Cai, Y. and Cercone, N.: Knowledge Discovery in Databases : An Attribute Oriented Approach, *VLDB*, pp.547-559 (1992)

[11] Han, J. and Fu, Y.: Discovery of Multiple-Level Association Rules from Large Databases, *VLDB*, pp.420-431 (1995)

[12] Imielinski ,T. and Mannila, H.: A Database Perspective on Knowledge Discovery, *CACM* 39-11, ACM, pp.58-64 (1996)

[13] Langley, P.: Machine Learning and Concept Formation, *Machine Learning* 3, pp.99-102 (1989)

[14] Mannila, H.: Methods and Problems in Data Mining, *Intn'l Conf. on Database Theory* (ICDT), pp.41-55 (1997)

[15] Miura,T: Database Paradigms Towards Model Building, *Object Roll Modelling*, pp.228-258 (1994)

[16] Miura, T. and Shioya, I.: Mining Type Schemes in Databases, *Conference and Workshop of Database and Experts Systems Applications* (DEXA), pp. 369-384 (1996)

[17] Miura, T. and Shioya, I.: Knowledge Acquisition for Classification Systems, *Tools with Artificial Intelligence*(ICTAI), pp.110-115 (1996)

[18] Miura, T. and Shioya, I.: Paradigm for Schema Discovery, *Intn'l Symposium on Cooperative Database Systems for Advanced Applications*(CODAS), pp.101-108 (1996)

[19] Miura, T. and Shioya, I.: Differentiation for Schema Discovery, *Intn'l Database Workshop*, pp.62-77 (1997)

[20] Miura, T. and Shioya, I.: Examining Complex Objects for Type Schema Discovery, *Conference and Workshop of DEXA*, pp.462-477 (1997)

[21] Miura, T. and Shioya, I.: Concept Formation From Databases, *Conference of DEXA*, to appear (1998)

[22] Mitchell, T.: Version Space - A Candidate Elimination Approach to Rule Learning, *IJCAI*, 1977

[23] Motro, A.: Intensional Answers to database Queries, *IEEE Trans. Knowledge and Data Engr.* 6-3, 444-454 (1994)

[24] Ng, R. and Han, J.: Efficient and Effective Clustering Methods for Spatial Data Mining, *VLDB*, pp.144-155 (1994)

[25] Quinlan, R: Induction of Decision Trees, *Machine Learning* 1-1, pp.81-106 (1986)

[26] Quinlan, R: Learning Logical Definition from Rules, *Machine Learning* 5-3, pp.239-266 (1990)

[27] Piatetsky-Shapiro, G. and Frawley, W.J. (Eds): Knowledge Discovery in Databases, *MIT Press* (1991)