データベース・システム 86－2
（１９９１．１１．８）

欠落データを含む木簡データベースのクラス階層化

上島 紳一　　園田 浩一　　大月 一弘
関西大学文学部　姫路獨協大学一般教育部　神戸大学教養部

本稿では、属性値に空値が多いデータに対して属性の欠落度に応じたクラス階層構成法を提案する。この構成法は、各属性の値の空値性に着目してデータを分類し、階層化を行う方法である。即ち各属性の有無のみを構造化の判断対象とし、属性値の意味は問わない方法である。本方式は、(1) 高速なデータ処理が行える、(2) データ更新、追加が容易である、(3) 既存のデータベースへ簡単に適用できる、などの特徴を持つ。また複雑な処理を実行したり、問い合わせを繰り返し実行することの多いデータに対してこの構造は有効であることを示した。

A CLASS HIERARCHY FOR WOODEN SLIP DBMS WITH INCOMPLETE OBJECTS

Shinichi Ueshima　　Koichi Sonoda　　Kazuhiro Ohtsuki
Faculty of Letters　College of Liberal Arts　College of Liberal Arts
Kansai university　Himeji-Dokkyo University　Kobe university

Yamate-cho, Suita city, Osaka 530, Japan
e-mail:ueshima@kansai-u.ac.jp

In database systems, such as wooden slip database management systems, some attribtutes of an object may have null values. In this paper, we present a simple class hierarchy for incomplete objects. All objects are classified into "class" by checking the number of null attributes. Then the "class" hierarchy is constructed through our simple procedure. It provides efficient access to data by passing null attributes.

1. はじめに

複数の専門家からインタビューによって得られる知識やフィールド調査データなどはデータの持つ属性が統一されていなかったり、また属性値が欠損している場合が多い。この様なデータはその性質上、属性値の取り扱いが複雑になる為,データを効果的に整理、分類することのできるデータベースシステムが期待されている。OODBMS（Object-Oriented DataBase Management System）[1] はデータ間の複雑な関係を階層構造として表現することができ、この様なデータベースの構築に適している。このシステムを実現する上でデータの持つ特性のクラス構造化が最も重要な課題となっている[2][3][4]。

クラス構造化に関する研究として、[2]は各々のデータの持つ属性と属性値をフレームに格納し、属性の出現頻度を指標にデータを分類する方法を提案している。これは再帰的手法を用いて属性値をクラス名に変換してデータの構造化を図っている。この方法は属性値の持つ意味をもとに分類している為に各属性の値の種類が少ない場合に有効である。しかしながら属性値の種類が多い場合にはクラス構造の記述が困難である為、適用が難しい。

著者らのグループでは東洋学研究支援システムとして研究者から得られる知識、調査データをデータベース化を試みている。木簡データは属性値の種類が多いのみならず、属性値が同じでも必ずしも両者が同一の意味を持つとは限らない。また属性値が欠落しているデータの数が [2] で見られる例と比較して極めて高く、[2] の方法は適用できない。

本稿では属性が空値である頻度の高いデータに対するクラス構成法を提案する。この構成法は、各属性の値の空値性に着目してデータを分類し、階層化を行う方法である。即ち各属性の有無のみを構造化の判断対象とし、属性値の意味は問わない方法である。本手法はデータ処理の高速化や見通しのよいデータの管理方法の実現を図ったものであり、複雑な処理を実行したり、問い合わせを繰り返し実行することの多いデータに対してこの構造は非常に有効であるものと考えられる。

2.1 において木簡データベースの構築の動機について述べ、2.2 において木簡データの属性値の欠落度について簡単に触れる。3節においては属性の空値の多いデータに対するクラス階層化手法を提案し、その性質について考察する。4節で今後の課題について検討する。

2. 木簡データベース

2.1 構築の動機

本データベースシステムは中国漢代の木簡を対象としている。長年、地中に埋もれていた為に複数の木簡をつないで構成された冊書は分断されて出土しており、多くの木簡は割れて木片となっている。これらの木簡は数の膨大さや情報の不完全さの為、大多数が未整理である。研究者は木簡に書かれた文字を判読し、そこから木簡全体の形を復原する為の有力なキーワードを抽出し、属性を付けて基礎データとする。これらのデータをもとに複数の木片間、また木簡間の意味的な関係付けを行って冊書や木簡の作者や作成時間の推定などを試みる[5][6]。

木簡データベースは、これら専門家から得られた知識やデータを基本データとしてデータベースに取り込み、割れた木片を合成したり、複数の木簡をつないで冊書に復原する為の処理機能を提供する研究支援システムである。出土した木簡を１単位として基本データを作成している。各データは木簡番号、人名、地名、官吏名、年号、干支、成語などのキーワードを属性として持つ。例えば人名や官吏名は木簡の作者の名前や官吏名ではなく木簡に現われる人の名前である。また年号や干支は木簡の作成された時間を表わすものとは限らない。同様に地名も木簡の出土地の場所を表わすものではない。これらの属性は値が同じであっても意味が異なる。このように木簡データの特徴からすると属性値が同一であっても同じ意味を持つとは限らず、意味による構造化が困難である。従ってデータの構造化にあたっては意味を用いずに分類する手法が必要である。

2.2 属性値の欠落度

木簡一本ごとに見れば、人名や地名などの基礎データの属性は書かれている数文字から数十文字の中から抽出する為に空値であることが多い。また木片から得られる属性は、完全な形を持つ木簡で属性値となりうる値も文字も欠損している為に属性値とはならないこともある。

木簡データにおけるデータの属性値の特徴を見る為に[6]において敦煌漢簡約1000本を対象として属性値の欠落度を調べた。表1に各属性値の欠落度を示す。木簡の番号属性や文字の存在を示す釋文属性の属性値がほぼ充足している一方、キーワードから成るその他の属性は空値であることが多いことがわかる。

表1　木簡データにおける属性値の欠落度

属性名	欠落度(%)
番号1	0
番号2	66
形状	5
人名	80
地名	79
官史名	86
年号	92
干支	82
成語	66
釋文	3

番号1 = (出土地, 疏勒河番号,　CMX番号),
番号2 = (流沙墜簡番号)

木簡データーベースでは、これら基本データに対してデータベースの中で処理を頻繁に繰り返し、不完全なデータに新しい属性値を付加して正確なデータベースに展開していく機能を実現することを目的としている。対象となるデータは約 30000 件であり、個々のデータの比較処理や検索を行う。その為にデータベースのスキーマは各木簡の持つ属性値を見通しよく管理できることが必要である。

次節では木簡自身の持つ属性値の欠落頻度の高さを利用してデータ処理の効率を上げるクラス階層の構成法を提案する。

3. 空値を含むデータのクラス階層

3.1 構成法

本節では、属性値に空値が多いデータに対して属性の欠落度に応じたクラス階層の構成法を提案する。与えられたデータの総数をNとし、それぞれを1つのオブジェクトとする。N個のオブジェクトに現われる属性が全部で m 種類存在するものとしよう。更にこのn個の属性のうち空値を含む属性数をn個とし、$(A_1, A_2, ..., A_n)$とする。

ある属性Aiを処理対象とする場合を考えよう。通常のデータベースでは属性 A_i の値の有無によらず処理プログラムはN個のすべてのオブジェクトを処理対象としなければならない。その際、属性 A_i の値が空値であるデータに対しても空値でない場合とほとんど同様のアクセス時間を要する。

そこで属性値の欠落度の高いオブジェクトを対象とする場合は、各属性値の有無を予め調べておき、属性値の存在しないオブジェクトに対しては処理の対象からはずしてアクセスを行なわないようにすれば処理時間の効率化が図れる。これを実現する方法として以下の方法を考える。

属性値の有無によってオブジェクトを分類すると、2^n個のクラスができる。クラス階層は構造が複雑とならないよう単一継承（Single Inheritance）形式とする。

先ず、この構造の最上位クラスとしてすべての属性値が空値であるクラス C_0 を作る。以下記述を簡単にする為、i 個の属性が空値でないことを i 個の属性を持つと呼ぶ。最上位クラスの下に、深さ1に位置するクラスとして1個の属性を持つクラス $C_1, C_2, ..., C_n$ を作る。ここでは、属性の充足状況を明示するためにクラスの名前は属性の名前と対応させる。例えば、クラス C_1 は属性 A_1 のみを持ち、他の属性値は空値である。

深さ2に位置するクラスは2つの属性を持つ。これ

らのクラスはそれぞれ深さ1に属する2種類のクラスの下に作ることができる。ここでは、深さ2に位置するクラスは自身の持つ属性番号より小さい属性番号のクラスの下に作る。例えば、属性 A_1, A_2 を持つクラス C_{12} は C_1 の子クラスとして作成し、C_2 の下には作成しない。深さ i のクラスは i 個の属性を持ち、深さ i-1 の子クラスとなる。深さ i のクラスは自身の持つ属性の内で、最も大きな番号を除いた属性を持つクラスの下に置かれる。例えば、クラス C_{134} はクラス C_{13} の子クラスとなる。最後に、深さ n のクラスとして全ての属性が充足しているクラス $C_{12...n}$ を置く。このクラス構造を Σ_n と呼ぶ。属性数 n が4である場合のクラス構造 Σ_4 を図1に示す。

3.2 クラス階層の特徴

（メソッド記述の容易さ）

クラス構造 Σ_4 は左に片寄った構造をしており、通常の2分木、4分木と構造が異なっている[7][8]。

グラフ上では属性 A_1 の含まれているクラスは連結されて一つの木（tree）を構成しているが、属性 A_4 の含まれているクラスは複数の木からなる森（forest）になっている。クラスのオブジェクトに対するメソッドの記述は、属性 A_1 については単純に書けるが、属性 A_4 については同じメソッドを複数定義する必要がある。しかし深さ i のクラスを深さ i-1 の下に付ける際に他の配置法を用いるとメソッドの記述が更に複雑になるものと考えられる。

（処理時間の短縮）

図1において属性 A_4 を持つオブジェクトすべてを対象にした処理を行なう場合（例えば検索など）、属性 A_1 に比べてアクセス時間が大きくなる可能性がある。しかし、このような処理はクラス階層上の位置にはほとんど影響を受けず、クラスに属するオブジェクト数に大きく依存することが次の様にして確かめられる。

属性 A_i についてみる。属性 A_i を持つクラスは深

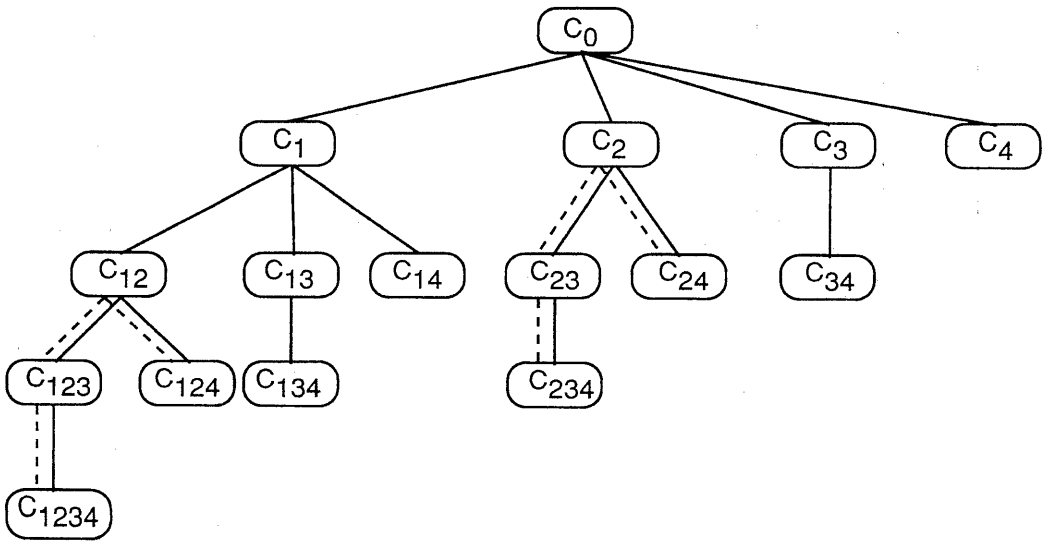

図1クスラ構造 Σ_4

さはn-i、クラス数2^{n-i}からなる2^{i-1}個の互いに同形な部分木を構成していることがわかる。図1に属性A_2を持つ2つの部分木を破線で示す。属性A_iに対する処理は空値でないA_i属性を持つクラスに対して実行する。表2に属性ごとの部分木の数と、部分木を構成するクラス数、処理対象総クラス数を示している。

クラス構造Σ_nを用いることにより、各属性に対する検索クラス数は、2^{n-1}となり、全体クラス数2^nの半分の数になる。N個のオブジェクトがすべてのクラスに平均的に分類されている場合は検索対象オブジェクト数もN/2となる。属性が含まれる部分木を移動する回数は2^{n-1}であるが、通常、N>>nの関係にある為に、それに要する時間はほとんど無視できるものと考えられる。

Σ_4において各クラスに属するオブジェクト数が各々100個であると仮定する。各属性A_i (i=1,2,3,4)に対して検索を行う場合、検索クラス総数は8、検索オブジェクト数は800となる。Σ_4を用いない場合は、1600のオブジェクトを検索しなければならない。一方、属性A_4の部分木を移動する回数は8となり、それに要する時間は小さい。従って本クラス階層を用いることにより、検索時間を大きく削減することができることが可能となる。

またk個の属性に同時に処理を行う場合、例えばk個の属性に対してAND条件による検索を行う場合、対象クラス数は2^{n-k}個、またk個の属性のどれかに処理を行う場合、例えばk個の属性に対してOR条件による検索を行う場合の対象クラス数は(2^n-2^{n-k})個である。k個の属性に対するAND条件による検索を考えると対象クラス数は、総クラス数の$1/2^k$となり、大幅な効率化ができる。

複雑な処理を実行したり、問い合わせを繰り返し実行することの多いデータに対してこの構造は非常に有効である。それぞれの時間の定量的な評価は別稿に譲る。

(柔軟なクラス数の決定)

本クラス階層はクラス数が2^nとなる為、クラス数がNP-完全となるのでnが大きい場合、実用が不可能である[7]。3.1では便宜上、空値を持つ属性をすべてクラス階層の対象としたが、実際のシステムにおいてはnの大きさを設計者が自由に選択できる。即ちクラス階層化の対象となる属性を自由に選ぶことができるからである。また他にも(1)最初に与えられる属性を検索の頻度によりグループ化する、(2)相互関連度の強い属性をグループ化する、などの方法が考えられる。木簡データベースでは人名、地名、官吏名、年号、干支、成語などを階層化の対象と考えている。

表2 処理対象となるクラス数

属性	処理対象クラス数	部分木数	クラス数／部分木
A_1	$2^{(n-1)}$	1	$2^{(n-1)}$
A_2	$2^{(n-1)}$	2	$2^{(n-2)}$
…	…	…	…
A_n	$2^{(n-1)}$	$2^{(n-1)}$	1
$\bigcap_{i=i_1}^{i_k} A_i$	$2^{(n-k)}$	$\bigcup_{i=i_1}^{i_k} A_i$	$2^n - 2^{(n-k)}$

（簡単なデータ更新）

新しいオブジェクトが追加される場合や既存のオブジェクトに新しい属性値が追加される場合なども属性値の意味を考えずにオブジェクトを機械的にクラスに割り当てることができる。

（既存のデータベースへの適用）

既存の OODBMS にこの方式を適用する場合、膨大なオブジェクトを持つクラスに対してこのクラス構造を追加実装することが可能であ。

4.おわりに

本稿では属性値の欠落度の高いデータを対象として属性値の有無に着目した処理効率の高いクラス構造を提案した。本方式は、(1) 高速なデータ処理が行える、(2) データ更新、追加が容易である、(3) 既存のデータベースへ簡単に適用できる、などの特徴を持つ。また複雑な処理を実行したり、問い合わせを繰り返し実行することの多いデータに対してこの構造は有効であることを示した。

尚、今後の課題として

・クラス階層化する属性の選択方法
・属性の相関性を利用した属性のグループ化手法
・処理時間の定量的評価
・クラス階層化する属性数の決定方法

などが考えられる。

謝辞

本稿をまとめるにあたり、日頃、ご指導、ご鞭撻頂く神戸大学小野厚夫教授、関西大学大庭脩教授に、衷心より感謝の意を表します。本研究の一部は平成3年度文部省科研費（03245109）の補助を受けて行われた。

参考文献

[1] W. Kim, F.H. Lochovsky (eds) : Object-Oriented Concepts, Databases and Applications, Addison-Wesley (1989)

[2] 大川、小泉、馬場口、手塚:"不完全情報を含むフレームの階層化"、電子情報通信学会論文誌 D-II Vol.J74-D-II No.6, pp.786-795 (1991)

[3] 打浪:"フィールド調査データ処理におけるマルチメディアデータベース"、情報処理 VOl.28 No.6, pp.773-783 (1987)

[4] 上島、園田:"木簡画像データのクラス階層"、平成３年度文部省科学研究費補助金（重点領域研究(1)）（課題番号03245109）、研究会報告書 (1991)

[5] 上島、園田:"敦煌漢簡画像データベースの構築"、平成２年度情報処理学会全国大会論文集 (1990)

[6] 大庭、上島:"コンピュータによる漢代木簡の索引作成の基礎的研究"、平成元年度文部省科学研究費補助金(一般研究 (B))(課題番号63450050)、研究成果報告書 (1990)

[7] 茨木:アルゴリズムとデータ構造、昭晃堂 (1989)

[8] T.Y. Young, K.S. Fu (eds) : "Handbook of Pattern Recognition and Image Processing", Academic Press (1986)

[9] D.J. DeWitt, P. Futtersack, D.Maier, F. Velez : "A Study of Three Alternative Workstation-Server architectures for Object-Oriented Database Systems", Proc. 16th International Conference on Very Large Databases (Brisbane), pp.107-121 (1990)