数理解析研究所講究録
第593巻 1986年 216-233

統計データベースにおける一連検索用ファイル編成法

近藤 貴士　　打浪 清一　　手塚 慶一

Takashi Kondo　Seiichi Uchinami　Yoshikazu Tezuka

（ 大阪大学 工学部 ）

1．まえがき

近年、多目的有効利用可能な統計データベースの必要性が認識され、その実現が望まれている。データベース構築の際、ファイル編成法はその応答性に直接影響を与えるため、データの特性またその利用形態に適した手法を選ぶ必要がある。統計データは多次元性を有しているためその格納には多次元ファイル編成法が適している。さらに統計解析においては多数の項目のうち少数の項目のみに着目した一連検索が用いられる場合が多い。一連検索とはある項目に対し同じ項目値を持つ多くのレコードを一括して処理する必要がある検索をいう。しかし従来の多次元ファイル編成法はこのような検索に対し良好な特性を示さない。そこで本稿では統計データの一連検索に対し有効な多次元ファイル編成法を提案する。

2．統計データの特性と要求分析

2．1．要求分析

統計データに見られる種々の特徴のうちファイル編成法を考えるうえで必要となる事項を挙げる。

(i) 多数のレコード

通常数千～数億程度である。

(ii) 多数の項目

一般に調査項目が多い。しかし検索にはごく少数の項目しか利用しない。

(iii) 多次元性

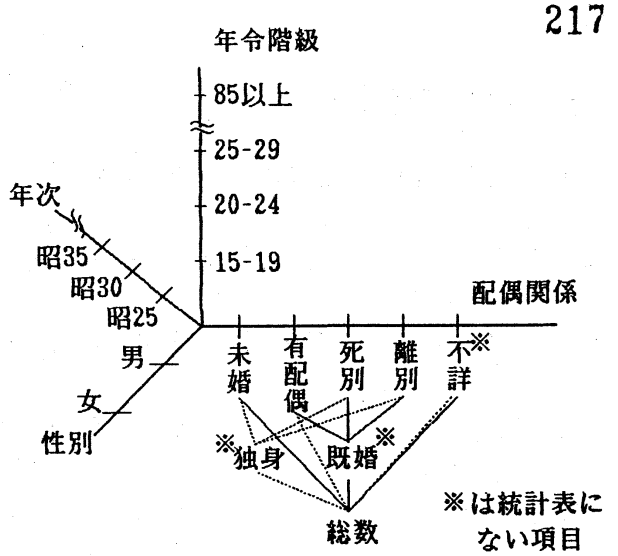


図1. 統計データの多次元性,階層性と特殊値

統計データの記述モデルとしては調査項目を軸とする多次元空間を用いると直感的に理解しやすい。利用者はこの多次元性に基づき様々なビューから検索を行う。

(iv) 項目値の階層性

各項目の項目値には階層性が存在する。階層性により上位のカテゴリを指定した検索は、下位のカテゴリに属するレコード全てが対象になる。

以上の特徴を持つデータに対する検索を必要レコード数によって次の2つに大別できる。

・ トランザクション検索(transaction retrieval)

ごく少数のレコードを処理するだけで要求を満足できるもの。

・ 一連検索(consecutive retrieval)

ある項目に対し同じ項目値を持つ多くのレコードを一括して処理してはじめて要求を満足できるもの。

統計処理では多数の多項目のレコードからなるファイル(データベース)を用いて少数項目のレコード少数からなる表を抽出する操作(要約)が多く、こ

れは一般に比較的多数のレコードを対象に行う処理で一連検索を要する。つまり一連検索は統計データを利用するうえで頻繁に行なわれる。

以上をまとめると、統計データベースには

（1） 多数の多項目レコードの格納,管理が可能

（2） 少数項目に着目した、比較的多数のレコードを対象とする検索を高速に行える。

（3） 検索条件の選びかたによるパフォーマンスの低下がほとんどない。

の条件を満たすファイル編成法が適している。

2．2．目的関数

統計データは一括入力されることが多く、更新あるいは削除されることが稀であるため目的関数Fは、F＝Tr(R)とする。ここでTrは検索に対する応答時間で、Rは検索条件である。CPUの演算コスト、仮想記憶域のトポロジ等を無視すると、F≒Bn(R)とおける。ここでBnは検索条件Rに対する答えを返すのに必要なデータを含む仮想記憶域の単位データ(バケット)の総数である。以後Bn をアクセスコストと呼び、これを小さくするファイル編成法を考える。ただしこれは検索条件・データの持つ性質などによって様々であるため、評価時はこれらに仮定を行う必要がある。

3．多次元ファイル編成

従来提案されている種々の多次元ファイル編成のうち代表的な手法を取り上げ、これらを統計データに適用すると以下のようになる。

(i) Inverted file

多数、多項目のレコードに対する転置リストが大きくなりすぎる。

(ii) k-d tree , Bucket Resolved Clustered Index File(BRCIF)

検索に関係する項目は全項目のうちのごくわずかであるため、レコード単位の格納形式であるこれらの手法は不要な項目の項目値へのアクセス分が本質的に多い。

(iii) Transposed file

検索はスキャンに基づいて行なわれるため不要なレコードの項目値に対するアクセスが多い。

(iv) Double Count Header Compression Scheme (DCHCS)

トランザクション処理に有効とされているが、アクセス対象データが散在するため一連検索に適さない。

検索レコード数と参照項目数によりこれらの手法が適している領域を図示すると図２となる。本稿で提案する統計データベースに適したファイル編成は＊の部分に該当する。

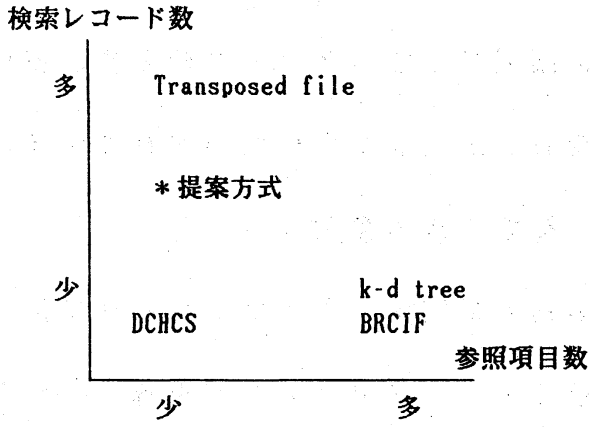


図２．　ファイル編成法の最適領域

４．本方式の設計概念

アクセスが必要なバケット数を低減させるための基礎概念は次の２つに大別できる。

(i)　データのクラスタリング

クラスタリングとは、同時に必要となる確率の高いデータ(類似データ)を物理的に近い位置に配置させることをいう。

・局所的クラスタリング

類似したデータは同じバケットにまとめて格納する。これによりアクセス対象となるバケット数を抑えられる。

・大域的クラスタリング

質問に対しアクセスが必要となるバケットが全記憶域内で集中するようにバケットの位置付けを行う。本稿ではこれについては言及しない。

類似データを物理的近傍に配置することにより、アクセス時の物理レベルでの選択・射影の効果が期待できる。

(ii) データ圧縮

計算量に多少のオーバーヘッドが生じても適切な圧縮法により記憶域が大幅に削減できるとアクセス時間や記憶域の大きさの点で有効である。

なおデータ量が多いため冗長性が高いファイル編成法は不適当である。

5. 提案手法の概要

本稿で提案する方式はレコード単位のクラスタリングを行うファイル編成に項目別データ格納の考えを導入したもので、つまり多次元空間をいくつかの部分空間に分割しそれぞれについて項目別ファイルを作成する方法をとる。

このように本手法は多次元空間$D_1 \times D_2 \times \cdots \times Dn$(nは項目数)を分割した部分空間が基本となっている。部分空間が全て直積ファイルとなるときこれら1つ1つをブロックと呼ぶことにする。ブロックは類似したレコードを管理する単位となっており、項目数個の順次ファイルとそれらへのインデックスから構成される。レコードはそれが属するブロックに、項目ごとに分割され格納される。順次ファイルにはそれぞれある項目が割り当てられており項

目値の格納に用いる。

本方式は必要なブロック内レコードのみをアクセス対象とすることにより選択が行え、かつ選ばれたレコードのうち必要項目のみをアクセス対象とすることにより射影が行えるためアクセスコストを低減するのに有効である。

(1) 入力アルゴリズム

(i) 投入されたレコードについてその属する部分空間に対応したブロックを求める。

(ii) 求めたブロックを構成する順次ファイルに投入されたレコードの各項目の値を別々に格納する。

順次ファイルについて次のことがいえる。

・一般に項目により必要な単位データ長(項目長)が異なるため順次ファイルの大きさは異なる。

・ブロックへの分割により各項目の項目値はブロック内では限定された値しかとり得ないことを利用してデータ

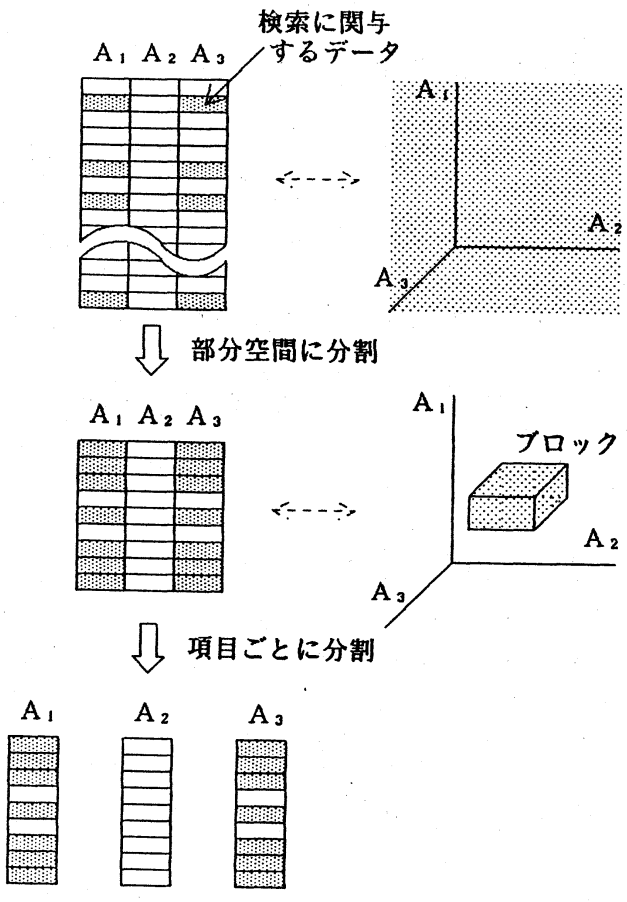


図3. 提案手法の概要(多次元空間との対比)

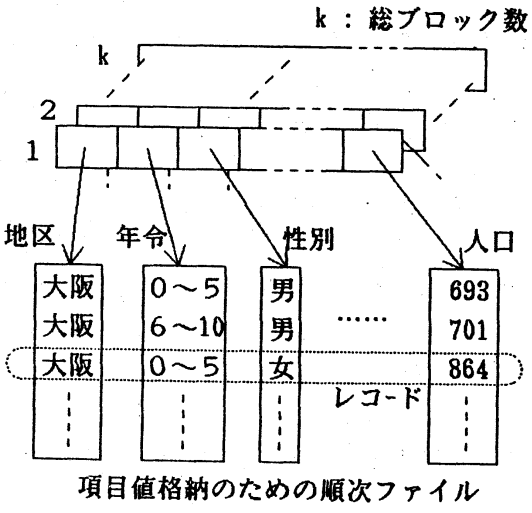


図4. ブロックの物理的構成図

圧縮が可能である。

(2) 検索アルゴリズム

ブロックは全多次元空間内で条件を与えている。ブロックiが示す条件をCiで表すことにすると、与えられた検索条件Qに対し、Ci∧Qが偽でないブロックのみに検索対象となるレコードが含まれている可能性がある。このようなブロックを検索対象ブロックという。検索対象ブロック全てに対し条件を満足するレコードをスキャニングにより特定し必要な項目値を取り出す。検索条件を与えている項目を条件項目、出力要求されている項目を参照項目と呼ぶことにすると、アルゴリズムは次のようになる。

(i) 条件を満足しうるブロックを全て求める。

(ii) 求めたブロックに対し、条件項目について項目値ファイルを先頭からスキャンしていき、条件が満たされたレコードの参照項目の値を出力する。

アクセスコストは、項目値順次ファイルを構成するバケット数を、対象ブロック全てと条件項目、参照項目全てにわたって総和をとった値となる。実際にはデータ圧縮が行えるためこれより少し小さい値となる。このコストは分割法によって様々になる。より分割数を大きくするほど不要なデータへのアクセスを減少できるが、逆にバケット内のメモリ利用率低下により結果としてアクセスが増大する。このように分割にはトレードオフが存在するため適切な分割を定める必要がある。

レコードが存在しうる全空間をブロックへ分割する方法としては種々考えられるが、ここではまずファイル作成時に分割を決定する方法について述べ

る。これを静的ファイル編成法とよぶ。説明のために以下の記号を定義する。

n　：全項目数　　Ai (1≦i≦n)：項目

N　：総レコード数　　Di：項目iの領域(項目値の全体集合)

6．静的分割法

6．1　静的分割法の概要

ブロックの定義をレコード入力の前に行う。ブロック管理が最も簡単になる方法は全定義域を塞の目状に分割し1つ1つをブロックとするものである。この例を図5に示す。

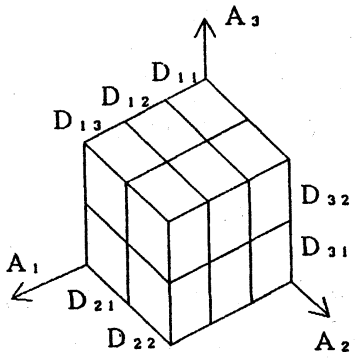


図5．静的分割法の例

静的分割法の評価のための準備として分割に対する評価法について述べる。

6．2　静的分割法の評価法

ここでは確率を用いた評価法を採用した。

レコードはあるブロックへ確率p で落ち込むとする。前述したとおり、評価対象はアクセスされるべきバケットの総数であるが、これは検索対象ブロックについて条件項目と参照項目の項目値を格納するバケット数X＝⌈r･ℓ/B⌉の総和をとったものである。ここで rはブロック内レコード数(確率変数)，ℓは項目長，Bはバケット容量である。以下では1つの項目を代表させ、そのバケット数を考える。

レコードが全部でN個投入されるとすると、あるブロックのレコード数がr となる確率P(r)は2項分布B(N,p)に従う(図6)。ここでN･pが十分大きいと仮定できるとすると2項分布B(N,p)を正規分布N(Np,Npq)に近似可能

となる。そこで正規化を行い、誤差関数erf(x)を用いると、Xの期待値E(X)は簡単に計算できる。

さて、確率変数の平均については公式、

$$E(X_1+X_2+\cdots+Xi)$$
$$=E(X_1)+E(X_2)+\cdots+E(Xi)$$

が成立するため、ある項目に絞って見たときある検索条件に対しアクセスが必要となるバケット数は、平均値を考える限りバケットごとに算出される平均値の和をとるだけで計算可能である。これを検索に関与する項目全てについて和をとると求めるアクセスコストが得られる。

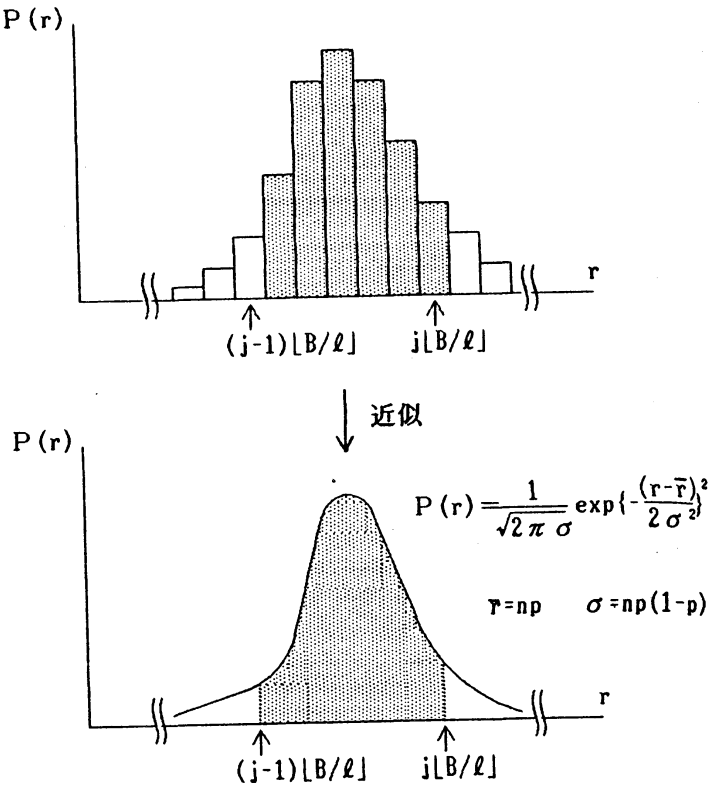


図6．　バケット数がjになる確率

静的分割法について適切な分割法を決定する一般的規則は未知であるが、与えられた分割法に対する評価を以上の方法により行うことができる。

6．3　静的分割法の評価と考察

簡単のためにつぎのように仮定したデータについて、前述した評価法に基づきアクセスコストの期待値を算出する。

・レコード数　：　１００万　　・項目数　：　５

・項目値の発生確率は一様で各項目で独立している。

・項目長　：　すべて８(bit)　　・バケット容量　：　16384(bit)

・分割は各項目同様に等分割を行う。

・検索条件は partial match queryとする。

以上の仮定を用い分割数１(分割しない),２,…,５について１項目あたりのコストの期待値を算出した結果を表１に示す。ここではデータ圧縮を無視している。

表１．アクセスコストの期待値(小数部は四捨五入)

c	例. k	1	2	3	4	5
0	(*,*,*,*,*)	489	512	652	1024	3125
1	(0,*,*,*,*)	489	256	217	256	625
2	(0,0,*,*,*)	489	128	72	64	125
3	(0,0,0,*,*)	489	64	24	16	25
4	(0,0,0,0,*)	489	32	8	4	5
5	(0,0,0,0,0)	489	16	3	1	1
	総ブロック数	1	2^5	3^5	4^5	5^5

c : 条件項目数　　k : 分割数　　* : don't care condition

計算結果の表で分割数１はtransposed fileと同じである。そこで表から提案方式をtransposed fileと比較すると次のことがわかる。

・条件項目が存在するとTransposed fileよりアクセスコストが少なくてすむ分割法が存在する。

・条件項目が多いほど必要となる１項目あたりのバケット数が少ない。

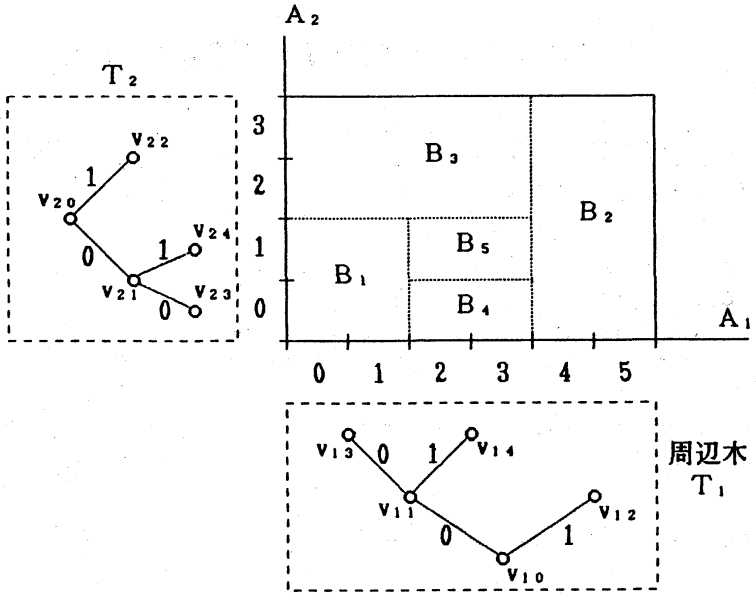
なお項目長などと密接な関係があり最適な分割法はcase by caseとなる。

静的分割法は格納されるレコードの数や分布,項目長などが既知であるときは最適分割を考えることができようが、実際には最適分割を定めるのは困難であることが予想され、データは逐次投入され分布も未知である。そこで投入されたデータにより自動的にブロック分割を行う動的分割法が実用的であると考えられる。

7．動的分割法

7．1　動的分割法の概要

本方式は最初全レコードを格納するブロックが1つだけ定義されており、レコード格納によりブロック内レコード数が大きくなり後述の基準を超えたとき、ブロック内レコードを後述の計算により定まる項目の項目値によって



(a)　周辺木とブロック

ノード	子ノード 左	子ノード 右	ブロック 1	2	3	4	5
v10	v11	v12	0	0	0	0	0
v11	v13	v14	0	0	1	0	0
v12	—	—	0	1	0	0	0
v13	—	—	1	0	0	0	0
v14	—	—	0	0	0	1	1
v20	v21	v22	0	1	0	0	0
v21	v23	v24	1	0	0	0	0
v22	—	—	0	0	1	0	0
v23	—	—	0	0	0	1	0
v24	—	—	0	0	0	0	1

v10={0,1,2,3,4,5}
v11={0,1,2,3}
v12={4,5}
v13={0,1}
v14={2,3}
v20={0,1,2,3}
v21={0,1}
v22={2,3}
v23={0}
v24={1}

(b)　ビットマップファイル

図7．　動的分割法のブロック管理法

２分割する方式である。

本方式はブロック管理にn個の周辺木とビットマップを用いる。周辺木Ｔｉは２分木で、項目iの項目値の集合に対応するノードから構成される。 根は項目iの全項目値に対応する。あるノードjがレベルhで、ある項目値集合Ｓｊに対応するとき、ノードjの左子ノードはＳｊのうち第hビットが“0”である項目値部分集合に対応し、ノードjの右子ノードはＳｊのうち第hビットが“1”である項目値部分集合に対応する。このようにノードは項目値集合をあらわす。またこれに従い、左子ノードへのアークを“０”,右子ノードへのアークを“1”とラベル付けする。

全てのブロックはこれらノードが示す項目値集合を一辺とする超直方体に対応させる。このような、ノードとブロックの対応づけを行うのがビットマップである。つまりあるノードv_{ij_i}のビットマップで“１”であるブロックは項目iのとりうる項目値はv_{ij_i}である。

ブロックが特定されるのに用いた周辺木Ｔi上の経路は０，１系列で示せる。ブロック内のレコード全ては項目iの項目値はこの系列を上位に持っているため本方式ではこの系列を省略して格納することによりデータ圧縮を行う。

（１） 入力アルゴリズム

（ｉ） 全ての項目について、投入されたレコードを構成する項目値のビット系列に従って各周辺木を根から順に葉までたどる。

（ii） 各項目について（ｉ）で求めた経路上のノードのビットマップの論理和をとる。これによりブロックの集合を得る。

（iii） 求めたブロック集合の積集合をとるとただ１つのブロックが定まる。

（iv） 求めたブロックが分割基準に達したときは分割を行う。これについては後述する。

（v） 求めたブロックにレコードを項目ごとに分割格納する。

以上の方法によると、Ai＝a であるレコード全ては、周辺木Tiをその根から aのビット系列に従ってビットマップをその論理和をとりながら葉に到着するまでたどっていき、結果が“1”であるビットに対応するブロックに格納されていることになる。

（2） 分割アルゴリズム

分割対象ブロック番号をp、未使用のブロック番号をfとすると、選定された分割項目dから分割は図８のように行えばよい。

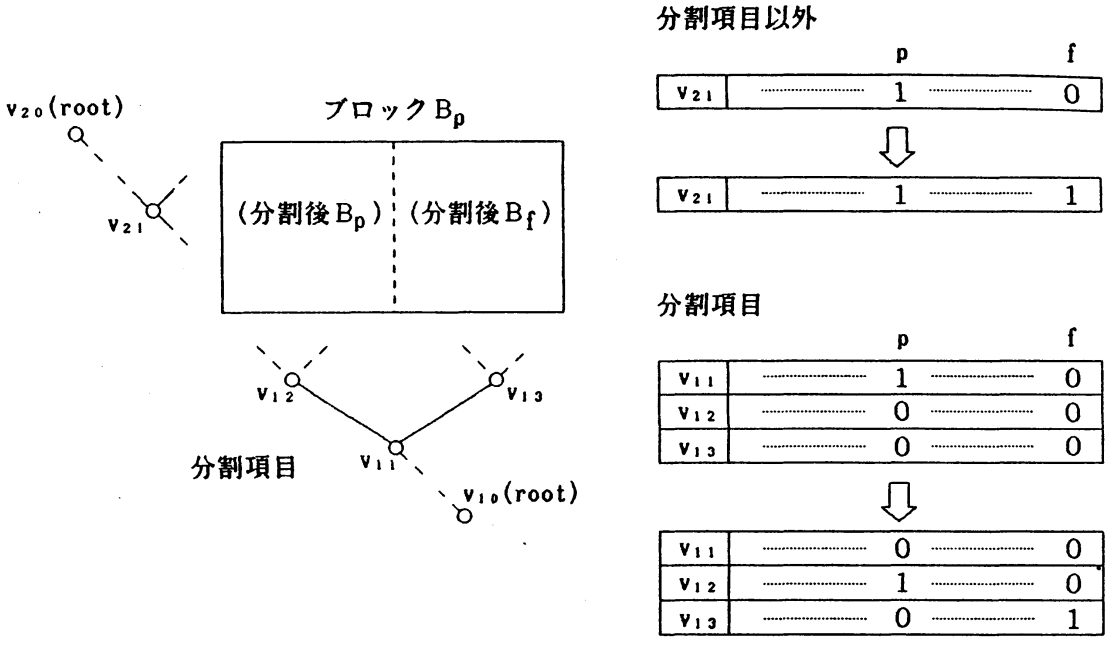


図８． ブロック分割によるビットマップの変化

(3) 検索アルゴリズム

(i) 条件項目全てについて、その項目値に対応するノードを求め、ビットマップからブロックの集合を得る。

(ii) 求めたブロック集合の積集合をとる。

(iii) 得られたブロック全てに対し、条件を満たすレコードの参照項目を出力する。

本方式では分割は項目値のビット系列で行うが、分割を行う時期やどの項目の値に着目して分割するかは非常に重要な問題点となるので次に分割基準について述べる。

7．2 分割基準

分割はアクセスコストを減少させるのが目的であるがその方法を検討するため分割によるアクセスコストの変化を考える。このためにqueryを次のように仮定する。

(i)検索条件はQ(Ai=a)の形に限定する。ここで条件項目i はランダムに等確率で選ぶとし、a はDiからランダムに等確率で選ぶ。

(ii)参照項目jは1つでランダムに等確率で選ぶ。

全てのi,jについて検索コストを考え、これらの平均値を最も減少させる項目dによる分割を局所的最適分割とする。

いま分割を考えるブロックB_0を分割対象ブロックと呼ぶことにし r個のレコードがあるとする。分割は項目dで行うこととし、分割を行うとレコードはブロックB_1にr_1個、ブロックB_2にr_2個と分かれるとする。

アクセスコストは検索対象となるバケット全てについて、条件項目に対す

るアクセスコストと参照項目に対するアクセスコストの和をとったものである。ある項目が条件項目になる確率、また参照項目になる確率は共に1/nで、ある項目iのある項目値が条件として選ばれる確率は1/{n·card(Di)}であるから平均コストは、

$$\sum_{i} \sum_{a \in Di} \sum_{j} \mathrm{ACOST} / n^2 \mathrm{card}(Di)$$

である。ここでACOSTは条件Q(Ai=a)に対し参照項目jの値が要求されたときの全ブロックに対するアクセスコストである。

分割により変化するのは分割対象ブロック分だけであり、これは

$$(1/n^2) \sum_{i} \sum_{a \in Di} \sum_{j} \mathrm{COST}/\mathrm{card}(Di) \quad \cdots ①$$

である。ここでCOSTは条件Q(Ai=a)に対し参照項目jの項目値が要求されたときの、分割対象ブロックの項目jの順次ファイルを構成するバケット数であるが、これはi,j,dの選びかたによって様々である。

詳細は省略するが(分割後の①式)−(分割前の①式)つまり分割による①式の増加は、ℓが大,nが大なる仮定を置き近似すると、

$$(1/n^2) \left[\sum_{i \neq d} Pi \left\{ \sum_{j} I(j) + (n-1) I(i) \right\} - Pd \left\{ \sum_{j} L(j) + (n-1) L(d) \right\} \right] \cdots ②$$

と表せる。ここでL(i)は分割前の項目iの項目値格納のためのバケット数,I(i)は分割による項目i格納のためのバケット数の増加である。②式の[]内の第1項は分割を行うことによるアクセスコストの増加すなわちデメリット、第2項はメリットをあらわしている。

そこで本方式では、レコードの格納のために新たなバケットが必要となるとき、

$$Div(d) = \sum_{i \neq d} Pi\{\sum_{j} I(j) + (n-1)I(i)\} - Pd\{\sum_{j} L(j) + (n-1)L(d)\} \cdots ③$$

を計算し、負となるdが存在するとき③式を最小にするdで分割するという分割法を採用する。これにより平均コストを最小にする準局所的最適分割を与える。

8．あとがき

統計データの特徴を分析し、その一連検索に適したファイル編成としてブロック分割による多次元ファイル編成法についてその基礎概念および概要を述べた。まず静的分割法についての評価法として確率を用いた方法を示し、Transposed fileと比較して有効である分割法が存在することを示した。これにより分割の有効性が示せた。また動的分割法は対象分野のデータに応じた分割を動的に行える有効な手法であり、その分割法、分割基準とその基準の理論的根拠を示した。本ファイル編成法は大量データを対象とし、多数の項目のうち少数項目に着目した一連検索が多用される分野、例えば文献検索システム、実験値計測システムなどに有効であると考えられる。動的分割法の特性評価のためのシミュレーションによる他の編成法との比較などが今後の課題として残されている。

参　考　文　献

[1] D.S.Batory "On Searching Transposed Files" ACM Transaction on Database Systems, Vol.4, No.4, pp.531-544(1979).

[２] Susan J.Eggers,Arie Shoshani "Efficient Access of Compressed Data" A LBL Perspective on Statistical Database Management, pp.179-189 (1982).

[３] 行政管理庁行政管理局統計主幹 "統計調査総覧" 全国統計協会連合会 (1976).

[４] 近藤貴士 "統計データベースにおける一連検索用ファイル編成法について" 信学AL研資, AL85-68 (1986).

[５] W.C.Lin,R.C.T.Lee,H.C.Du "Common Properties of Some Multi-attribute File Systems" IEEE Transaction on Software Eng. SE-5,2 pp.160-174 (1979).

[６] J.Martin著,国友他訳 "データベース" 日本コンピュータ協会(1983).

[７] 小菊他 "リレーショナルデータベースを用いた統計検索機能と実現方式" 情処DB研資42-2(1984).

[８] 大沢裕,坂内正夫 "良好な動特性をもつ多次元点データ管理構造の一提案" 信学論Vol.J66-D, No.10,pp.1193-1200(1983).

[９] 坂内正夫,大沢裕 "画像データベースにおけるデータ表現・管理方式" 信学論(D), J68-D, 4, pp.434-441 (1985).

[10] 総理府統計局 "第三十二回日本統計年鑑"(1982).

[11] 田中譲 "複数属性ファイルの適応型最適ファイル分割法" コンピュータソフトウェア, Vol.2, No.2, pp.15-28 (1985).

[12] 特定研究「多目的統合統計データバンクの開発」総括班事務局 "STATISTICAL DATA BANK" Vol.1, No.1-4 (1983～1984).

[13] M.J.Turner,R.Hammond,P.Cotton "A DBMS for Large Statistical Databases" VLDB '79, pp.319-327.

[14] 打浪,手塚 "統計用DBMSの構成に関する一考察" 信学総全大(1984).

[15] 上田尚一 "統計データの見方・使い方" 朝倉書店(1981).