# Webログ分析における高頻度アクセスパターン検出を支援するデータキューブモデル

助川　貴信† 　大森　匡† 　星　守† 　蔦谷　雄一†

† 電気通信大学大学院情報システム学研究科　〒 182–8585 東京都調布市調布ヶ丘 1–5–1
E-mail: †{sukegawa,omori}@hol.is.uec.ac.jp

**あらまし**　近年，データベースに蓄積された大量のログ情報を分析し，有用な情報を抽出する試みが盛んである．一般的に，有用な情報を抽出する方法として高頻度パターンや相関性ルールを効率的に発見するアルゴリズム Apriori が知られている．しかし，単純に Apriori を用いてログ分析をするだけでは多くの情報が出てしまい, それらは必ずしも有用な情報であるとは限らない. そのため，条件を決めて分析し，それらの結果を比較しながら，理解のできる状態まで，繰り返し分析する必要がある．そこで，著者らによって，Apriori を用いた繰り返しの分析をサポートする**アイテムセットキューブ**と呼ぶ新しいモデルが提案された．本稿では，新しい分析モデルである**アイテムセットキューブ**について述べて，そのキューブを操作する実体化，ロールアップ，ドリルダウンの計算方法を述べる．

**キーワード**　Apriori ，　Itemset cube

# A Data Cube Model Supporting Web-log Mining

Takanobu SUKEGAWA†, Tadashi OHMORI†, Mamoru HOSHI†, and Yuichi TSUTATANI†

† The University of Electro-Communications, Graduate School of Information Systems　Chofugaoka 1–5–1, Chofu, Tokyo, 182–8585 Japan
E-mail: †{sukegawa,omori}@hol.is.uec.ac.jp

**Abstract**　Frequent pattern detection from a large Web-log dataset has been considered a useful method of Web-site analysis. However, a naive usage of frequent pattern detection may lead to producing another meaningless dataset, namely, too much or trivial frequent patterns about which Web-pages were visited. To overcome this difficultly, this paper proposes a new datacube model supporting interactive process of Web-log mining. This datacube is powerful when a human user continues Web-log analysis by using frequent pattern detection and under a traditional datacube approach. This paper describes our new datacube model, denoted by *itemset cube*, and then efficient data-processing algorithms of three associated operations *materialize*, *roll-up* and *drill-down* are described.

**Key words**　Apriori ，　Itemset cube

## 1. 研究背景と目的

近年，データベースに蓄積された大量のログ情報を分析し，有用な情報を抽出する試みが盛んである．Web ログを分析して有用な情報を得る手法として，高頻度パターンや相関性ルールを発見してユーザの行動パターンを分類する方法がある [1], [2]. それらの方法では，高頻度パターンを効率的に発見するアルゴリズム Apriori が用いられている [3]. 高頻度パターンや相関性ルールを発見することで，ユーザが Web を訪れた際にどのようなページの組み合わせをよく見ているかという情報を得ることができる．そのような情報を可視化して，有用な情報を模索することも行われている [4]. しかし，単純に Apriori を用いてログ分析をするだけでは多くの情報が出てしまい, それらは必ずしも有用な情報であるとは限らない. そのため，条件を決めて分析し，それらの結果を比較しながら，理解のできる状態まで，繰り返し分析する必要があると言われている [5], [6]. そこで，著者らは Apriori を用いて繰り返しの分析をサポートする**アイテムセットキューブ** と呼ぶ新しいデータキューブモデルを提案している [8], [9], [10]. 本稿では，新しい分析モデルである**アイテムセットキューブ**について述べて，そのキューブを操作する実体化，ロールアップ，ドリルダウンの計算方法を述べる．

本節では，研究背景と目的を述べた．２節では，一般的な Web ログ分析とアイテムセットキューブによる分析を非形式的に述べる．３節では，アイテムセットキューブを形式的に定義する．４節では，実体化の計算方法について述べる．５節では，ロールアップ，ドリルダウンの計算方法について述べる．

## 2. アイテムセットキューブによる Web ログ分析

この節では，一般的に行われている Web ログ分析とその問題について述べた後，それらに対して，我々の方針とアイテムセットキューブについて述べる．

一般的な Web ログ分析では，情報サーバに記憶されるログ情報を分析して，有用な情報を抽出する方法として，高頻度パターンの発見が行われている．ここで，図 2–A のような Web サイトがあったとする．Web サイトに訪れるユーザの行動は，目に見えないため，直観的にどのようなページを辿っていったかということがわからない．そのため，図 2–B のようなデータベースに記憶されたアクセスログの分析が行われている．一般的な Web ログ分析は，まずアクセスログをあるユーザが一定時間の間にどのページを見たかというアクセスレコード (セッションレコードとも呼ばれるが本稿ではこの呼称に統一する) の型に変形する (図 1)．次に調べたい状況を表す条件 (例えば，ヒット数の多いドメインで講演会のページを見た) を決めて，その条件を満たすアクセスレコード集合を抜き出し，そこから Apriori を用いてページ列の高頻度パターンを検出することが行われる．以下，このような分析の方法を「条件を決めて分析する」と呼ぶ．一般的に Apriori での分析では，アクセスレコードを構成する各要素をアイテムに対応させて，良く起きるパターンを検出する．ここで言うパターンとは，アイテムの組み合わせである**アイテムセット**に対応する．アクセスレコードの総数に対して，出現頻度がある閾値 $\theta(\%)$ を越えるようなパターンを**高頻度アイテムセット**と呼ぶ．これによって得られたページ列の高頻度アイテムセットから図 2–B のような「ユーザの振る舞い」を視覚化して，有用と思われる情報を抽出することが行われる．しかし，一度の分析でユーザの振る舞いを視覚化したとしても，その結果には意味のある情報は少ない．

そこで，我々は次のような方針で分析することを決めた．アクセスレコードを構成する要素であるドメイン (G1，G2，G3，...)，ページ (P1，P2，P3，...)，時間 (T1，T2，T3，....)，etc から分析する条件を決める．決められた条件から分析し，それらの結果を比較して，有用な情報を抽出する．例として図 2–C のように，ドメインに対して G1，G2，G3 と条件を決めて，見たページに対して P1，P2，P3 と条件を決めたとき，図 2–C の 3 × 3 の表のように各条件に応じた高頻度アイテムセットを計算し，それぞれの結果を比較して分析する．このように，各条件に応じたユーザの振る舞いをキューブ構造に従って管理し，分析することで初めて新たな事実がわかる．このような分析の流れをサポートするモデルがアイテムセットキューブである．以下，高頻度アイテムセットはページ列のみで構成されるとする．

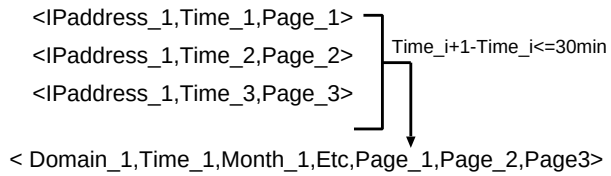


図 1　アクセスログからアクセスレコードへの変換

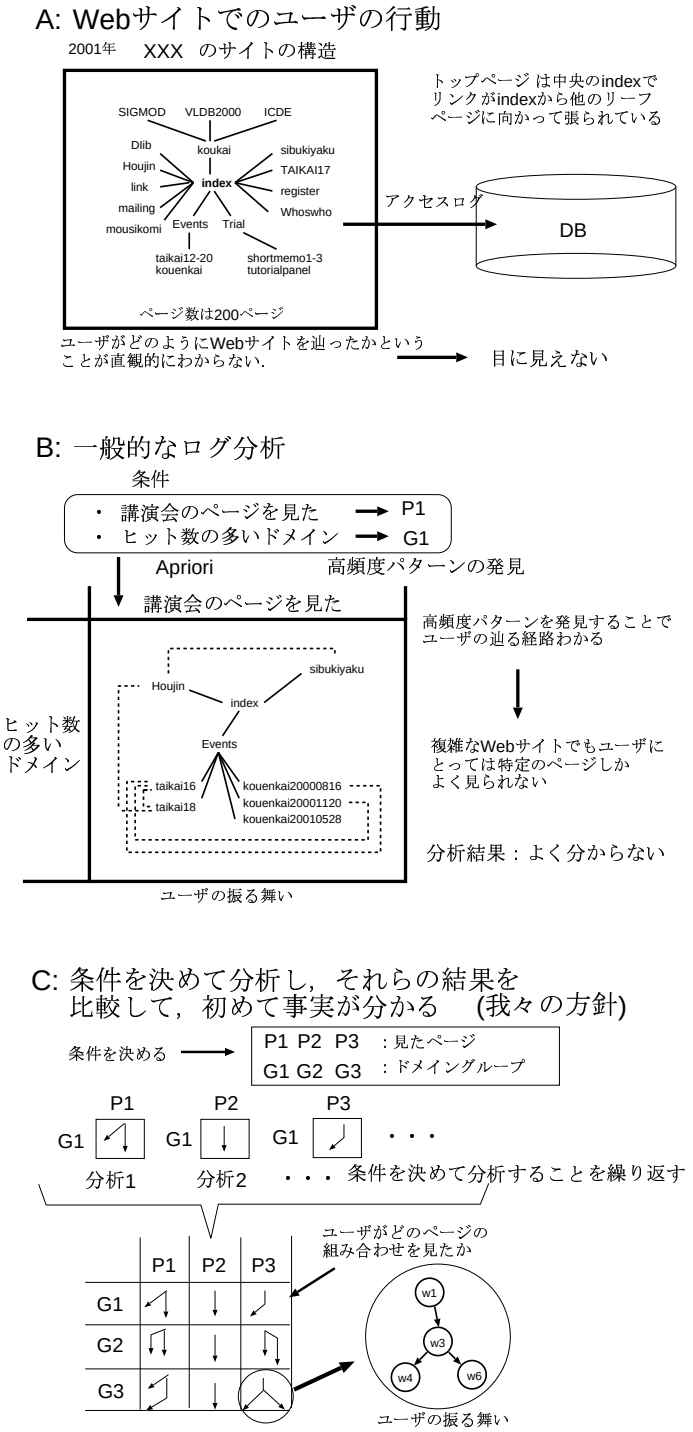


図 2　A：Web サイトに訪れるユーザの行動は直感的にわかりにくい．B：一般的に Web ログ分析では，一度の分析でユーザの振る舞い (どのようなページを見たか) を分析しても，意味のある情報は少ない．C：条件を決めて分析し，それらの分析結果を比較して，初めて新たな事実がわかる．

## 3. アイテムセットを管理するデータキューブモデル

繰り返し分析をするには，得られる情報の比較や条件の制御が簡単に行えなければならない．そこで，対話的に分析を繰り返すための分析モデルとして，[3]，[4]，[5] で提案されたアイ

テムセットを管理するデータキューブモデルについて述べる (図3).

### 3.1 アクセスレコード

本稿では，Time を時間と月に変換し, IPaddress をドメインに変換し, その他 (Web 上でのイベントの有る無し) を追加し (図1), 分析の対象として < ドメイン, 時間, 月, その他,「ページ」> の 5 属性の形のアクセスレコードを用いる. (但し, ここで言う属性「ページ」は図 1 の様にページ集合を値にとる).

### 3.2 アクセスレコードを管理する $K$ 次元空間の定義

定義域 $D$ を与えたとき，$D$ から原子的な値を 1 つ取るような属性を**関係属性**を呼ぶ．一方，$D$ から原子的な値の集合として取るような属性を**アイテムセット属性**と呼ぶ. このとき，以上の属性 $K$ 個から構成された空間 $S$ が，アクセスレコードを管理する. アクセスレコードは，$S$ の 1 点で表現される．以下，アクセスレコードをレコードと表記する.

### 3.3 属性に対する分割の定義

任意の属性 $A$ において， $A$ に対する分割 ($segmentation$) を定義する. $A$ に対する分割 $P$ とは，$A$ の値に対するブール述語 $p_1$， $p_2$， ...， $p_k$ である．任意の $p_i$，$p_j$ に対し，$p_i$ と $p_j$ は互いに素である場合も， そうでない場合も許す．互いに素とは，任意の $p_i$，$p_j$ に対して，$p_i$ と $p_j$ の両方を満たすような値が存在しえない場合である.

表記方法

属性名は < > で表記する．属性に対する分割は { } で表記する．属性の値に対するプール述語を述語と呼び，述語は「 」で表記する.

例 : 属性に対する分割

- < 月 > 属性の分割を {「３月から６月」,「５月から７月」} とする.
- < 月 > 属性の分割を {「３月」,「４月」,「５月」} とする
- < ページ > 属性の分割を {「ページ１を見たか」,「ページ２を見たか」,「ページ３を見たか」} とする.

我々のデータキューブモデルでは, 属性の分割に含まれる述語をアイテムに対応させる.

### 3.4 データキューブとセルの定義

$k$ 個の属性 $A_1$，$A_2$， ...，$A_k$ が与えられたとき，各 $A_i$ $(1 \leqq i \leqq k)$ に対して適当な分割 $P_i = p_{i1}, p_{i2}, ..., p_{il_i}$ を与えるとする．この時、組 [ $A_1$，$P_1$，$A_2$，$P_2$，...，$A_k$，$P_k$ ] をキューブ構造と呼ぶ．このキューブ構造において，**セル** [ $p_{1j_1}$, $p_{2j_2}, p_{3j_3}, ..., p_{ij_i}$ ] は, 各属性 $A_i$ において述語 $p_{ij_i}$ $(1 \leqq j_i \leqq l_i)$ を満たすという条件を示す．このセルに，その条件を満たすレコード集合から計算したある値 $v$ を与える．**データキューブ**とは，組 [ $A_1$，$P_1$，$A_2$，$P_2$，...，$A_k$，$P_k$ ] で構成されるセルに計算したある値 $v$ を持たせた多次元配列のことである．以後，[ $A_1$，$P_1$，$A_2$，$P_2$，...，$A_k$，$P_k$ ] と表記した場合，データキューブのことを示す.

**＜スカラキューブ＞**

セルの値 $v$ として，当該セルを満たすレコード集合から求めたスカラ値を用いる時，そのようなデータキューブを**スカラキューブ**と呼ぶ.

**＜アイテムセットキューブ＞**

セルの値 $v$ として，当該セルを満たすレコード集合から求めた高頻度アイテムセットを用いる時，そのようなデータキューブを**アイテムセットキューブ** と呼ぶ．ただし，セルの値を相互に比較するために，各セルの計算する情報の基準を一定にする (セル毎の足切り値率 $\theta$ を揃える).

### 3.5 演　　算

データキューブを操作する演算の定義をする.

**[実体化]** $k$ 個の属性 $A_1$，$A_2$， ...，$A_k$ が与えられたとき，各 $A_i$ $(1 \leqq i \leqq k)$ に対して適当な分割 $P_i = p_{i1}, p_{i2}, ..., p_{il_i}$ を与えるとする．この時，キューブのセル [ $p_{1j_1}, p_{2j_2}, p_{3j_3}, ..., p_{ij_i}$ ] からある値 $v$ を計算し，その値 $v$ をセルに与えてデータキューブ [ $A_1$，$P_1$，$A_2$，$P_2$，...，$A_k$，$P_k$ ] が計算される.

**[ロールアップ]** データキューブ [ $A_1, P_1, A_2, P_2, ..., A_k, P_k$ ] の属性 $A_i (1 \leqq i \leqq k)$ の分割 $P_i = \{p_{i1}, p_{i2}, ..., p_{il}\}$ に含まれる述語の論理和を用いて, 属性の分割 $P_i' = \{p_{i1} \cup p_{i2} \cup ... \cup p_{im}$ ， ... ... ， $p_{in+1} \cup p_{in+2} \cup ... \cup p_{il}\}$ と作り変える $(1 \leqq m \leqq n \leqq l)$.

**[ドリルダウン]** データキューブ [ $A_1, P_1, A_2, P_2, ..., A_k, P_k$ ] の属性 $A_i (1 \leqq i \leqq k)$ の分割 $P_i = \{p_{i1}, ..., p_{il}\}$ に含まれる述語を用いて，属性の分割 $P_i = \{p_{im_1}, p_{im_2}, ...,$ ... ... $, p_{in_1}, p_{in_2}, ...\}$ と作り変える $(p_{i1} = p_{im_1} \cup p_{im_2} \cup ...$ ， $p_{il} = p_{in_1} \cup p_{in_2} \cup ...)$ .

例 : ロールアップ

- < 月 > 属性の分割 {「３月」,「４月」,「５月」,「６月」} の述語を変えて，分割 {「３月から４月」,「５月から６月」} と作り変える.
- < ドメイン > 属性の分割 {「ドメイン１」,「ドメイン２」,「ドメイン３」} の述語を変えて，分割 {「ドメイン１とドメイン２」,「ドメイン２とドメイン３」} と作り変える.
- < ページ > 属性の分割 {「ページ１を見たか」,「ページ２を見たか」,「ページ３を見たか」,「ページ４を見たか」} の述語を変えて，分割 {「ページ１またはページ３を見たか」,「ページ２またはページ４を見たか」} と作り変える.

例 : ドリルダウン

- < 月 > 属性の分割 {「３月から４月」,「５月から６月」} の述語を変えて，分割 {「３月」,「４月」,「５月」,「６月」} と作り変える.
- < ドメイン > 属性の分割 {「ドメイン１とドメイン２」,「ドメイン２とドメイン３」} の述語を変えて，分割 { 「ドメイン

１」,「ドメイン２」,「ドメイン３」} と作り変える.

• < ページ > 属性の分割 {「ページ１またはページ３を見たか」,「ページ２またはページ４を見たか」} の述語を変えて, 分割 {「ページ１を見たか」,「ページ２を見たか」,「ページ３を見たか」,「ページ４を見たか」} と作り変える.

### 3.6 データキューブモデルによる分析例

上で述べたデータキューブモデルでは，属性の分割に含まれる述語をアイテムに対応させる．スカラキューブとアイテムセットキューブの各軸は<ドメイン, 時間, 月, etc, ページ>の属性とする (図 3–(a))．スカラキューブの値は，セルに対応するレコードの総数 (ヒット数) を持つ (図 3–(b))．アイテムセットキューブのセルの値は, セル計算範囲のレコード群から求められた高頻度アイテムセットの集合を持つ (図 3–(c))．これから Web 巡回グラフ (ユーザの振る舞い) を計算できる．以下では，アイテムセットキューブはセル値として，このグラフを持つと見なし話を進める.

例：分析手順

スカラキューブの値を見ること (図 4–(a)) で大まかな分析の方針を立て，ヒット件数の多いドメインと少ないドメインにグループ化 (図 4–(b)) して，アイテムセットキューブで各条件に応じた高頻度アイテムセットの計算を行なってグループ毎の行動の違いを見たり，さらに月で実体化することでそれらのグループの行動を時間軸に沿って見ることが出来る (図 4–(c)).

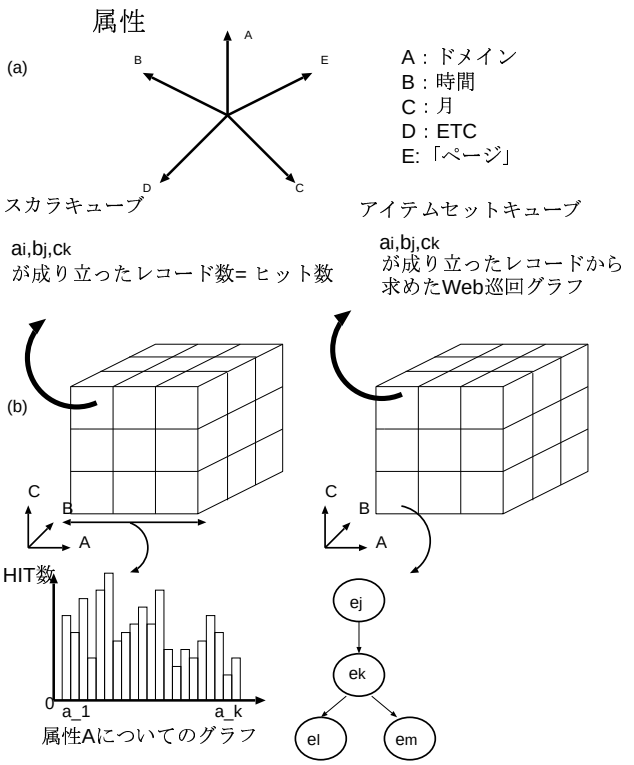


図 3　データキューブモデル

## 4. 実体化のアルゴリズム

実体化されたキューブは，属性の分割が互いに素である場合と属性の分割が互いに素でない場合に分けられる (図 5)．属性の分割が互いに素である場合, セルに対応するレコード集合は重複しないため，セル毎に個別に Apriori を用いて実体化をすればよい．この実体化の方法を Naive 法と呼ぶ．しかし，属性の分割が互いに素でない場合，セルに対応するレコード集合は重複するため，Naive 法で実体化すると，データベーススキャンと候補アイテムセットの重複が生じる．そこで，1 度のデータベーススキャンで，さらに重複する候補アイテムセットを除去して実体化する方法を以下で提案する．この方法を Cubic Apriori(CA 法) と呼ぶ.

ある属性 $A$ に対して，分割を $P = p_1, p_2, p_3, ..., p_N$ といたとき，一次元アイテムセットキューブ [A,P] のすべてのセル $[c_1, c_2, c_3, ..., c_i]$ (セル $c_i$ は属性 $A$ において述語 $p_i$ に対応する. $(1 \leqq i \leqq N)$) に対して足切り値率 $\theta\%$ で一括計算する際のアルゴリズムを説明する.

・準備

候補アイテムセットを $I$ としてとき $v(I) = [c_1, c_2, ..., c_i]$ とは, セル $c_1, c_2, ..., c_i$ での $I$ のサポート数を値に持つ配列である (図 6)．また, アイテムセットキューブに与えられる長さ $k$ の高頻度アイテムセット集合を $L(k)$ とし，長さ $k$ の候補アイテムセット集合を $C(k)$ とする.

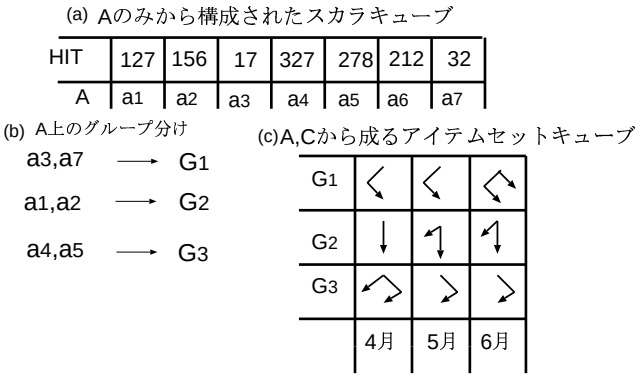


図 4　分　析　例

- **CA 法**

step1　$k = 1$ のとき，一度のデータベーススキャンで長さ 1 の候補アイテム $I \in P$ のカウントをセル $[c_1, c_2, c_3, ..., c_i]$ に対して，実行してその値を $v(I)$ に与える．もし，$v(I)$ の $j$ 番目の値がセル $c_j (1 \leqq j \leqq i)$ の足切り値率 $\theta$ を越えたなら，$I$ を $L(1)$ に追加する.

step2　$L(k) = I_1, I_2, ..., I_x$ から長さ $k+1$ の候補アイテムセット $J$ を次の条件で生成する:「$J$ の長さ $k$ のサブアイテムセットを $J'$ とする．そのとき，あるセル $c_j$ に対してすべての $J'$ がそのセルの足切り値率 $\theta$ を越えていなければならない」．生成された $J$ に対して，$J$ を $C(k+1)$ に追加する.

step3　候補アイテムセット $I \in C(k+1)$ のサポート数のカウントを一度のデータベーススキャンで行う．$p_j$ を満たすレコー

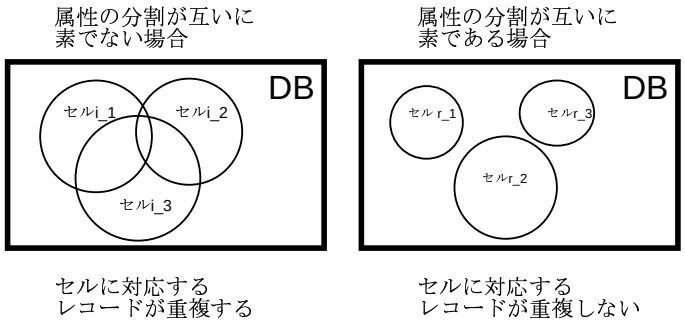


図 5　レコードの重複

候補アイテムセット AB ，AC ，BC の数げえ上げは
候補アイテムセットカウンタで行う.


候補アイテムセットカウンタとは,
各セルのサポート数を与えた配列である

図 6　候補アイテムセットカウンタ

ABC

| | AB | AC | BC |
|---|---|---|---|
| セル1 | freq | freq | freq |
| セル2 | freq | | |
| セル3 | | | freq |

ABCは候補アイテムセットになる

ABC

| | AB | AC | BC |
|---|---|---|---|
| セル1 | | freq | freq |
| セル2 | freq | freq | |
| セル3 | | | freq |

ABCは候補アイテムセットにならない

図 7　候補アイテムセットの生成条件

ド $r$ 毎に $C(k+1)$ のサポート数を次のようにカウントする．レコード $r$ で $I$ が成り立つとき，$r$ が満たしているすべての $c_j$ に対して，$v(I)$ の $j$ 番目の値をインクリメントする．

step4　$L(k) = \phi$ になるまで step2 に戻る．□

CA 法の step2 の候補アイテムセット生成の条件は図 7 になる．

### 4.1　実体化の評価

使用する人工データは, N5000，D100000，T20，I4．(N:アイテム数，D:レコード数，T:1 レコードの長さの平均，I:高頻度アイテムセットの長さ) で行った．計算機の環境は，Pentium4(1.5GHz), メモリ 256MB で実験を行った．

- 実験 1

実体化するセルに対応するレコードの重複の数を変化させて，4 セル G1，G2，G3，G4 の実体化の実験をした (図 8)．実体化する 4 セル G1，G2，G3，G4 のレコードの重複数は，セルに含ませるアイテムの数で変化させる．図 8 の縦軸は計算時間 (秒) で，横軸は，セルに含ませるアイテムの数を示す.

実験 1 より，セルに対応するレコードの重複の多い場合，データベーススキャンと候補アイテムセットの重複除去が効果的に機能したことがわかる．Naive 法に比べて，$\frac{1}{3}$〜$\frac{1}{4}$ 程度の計算時間で行えた．また，重複の少ない場合については，Naive 法と同じぐらいの計算時間で行えた.

- 実験 2

使用する人工データは,N10000，D300000，T20，I4．実体化するセルの数を 10 セル，20 セル，30 セルと変化させて実体化した (図 9)．セル同士のレコードの重複は，多い場合を想定してセルに含ませるアイテムの数は 300 とした．セルを満たすアイテムの分け方は，アイテム (0-9999) から 0-299，300-599，600-899，... とした．図 9 の縦軸は計算時間 (秒) で，横軸は，実体化するセルの数を示す.

実験 2 より，セルの数に対しても，Naive 法の $\frac{1}{5}$〜$\frac{1}{6}$ の時間で計算が行えた．CA 法は，レコードの重複が多い複数のセルを実体化するとき，計算コストを大幅に軽減できることがわかる.

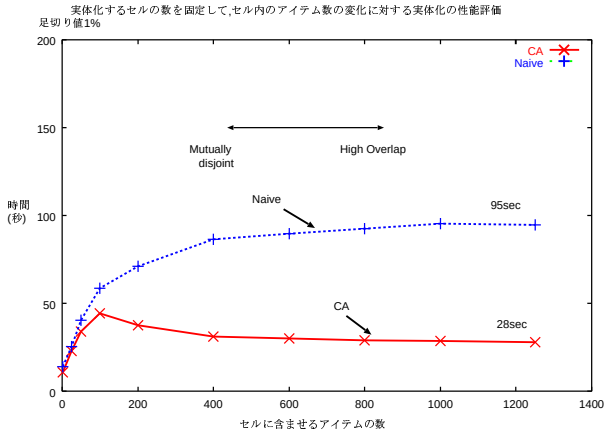


図 8　セルに対応するレコードの重複数の変化に対する実体化の性能評価

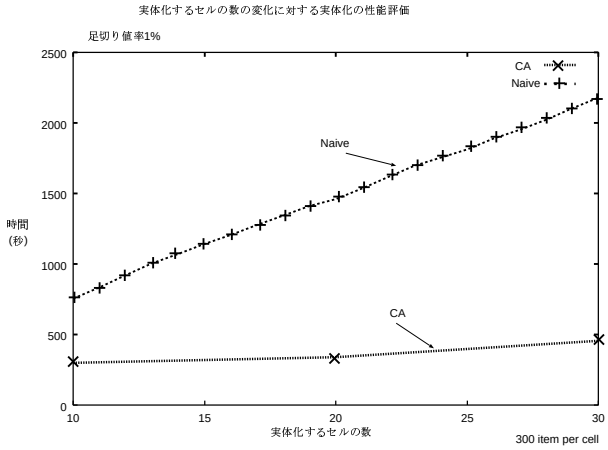


図 9　実体化するセルの数の変化に対する実体化の性能評価

### 4.2　実　　例

実際のアクセスログによる分析例を示す (図 10)．Web サイトの構造は，図 2−A である．top ページは中央の index で，index

から他のリーフページに向かって張られている.

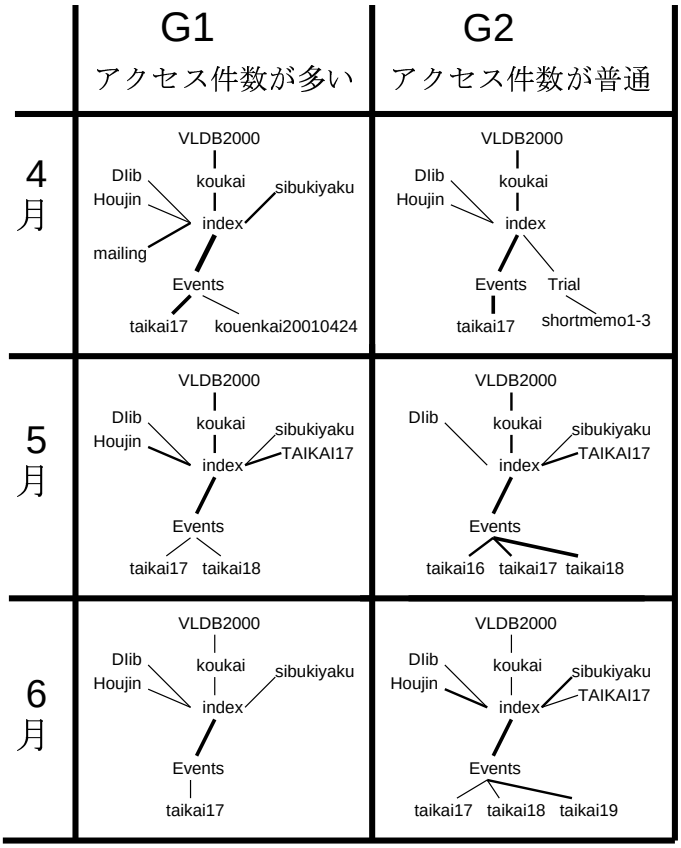


図 10　実　　例

## 5. ロールアップ, ドリルダウンの高速化

### 5.1 方針の概略

ロールアップとドリルダウンとは，実体化されたキューブに対して属性の分割を規定する述語を変えてキューブを変形させる演算である．例として，< 月 > 属性の分割を {「1 月」,「2 月」, ...,「12 月」} と与えて実体化したキューブに対して、その分割を {「春 (4〜6 月)」，夏 (7〜9 月)」,「秋 (10〜12 月)」,「冬 (1〜3 月)」} と変えたい場合がロールアップ，その逆がドリルダウンになる．この例のように，属性の分割が互いに素である場合，セルに対応するレコード集合は重複しない．この場合，ロールアップについては，元のセルの高頻度アイテムセット集合をマージして数え直せば良い(注1)．しかし，ページ属性の分割のように，互いに素でない分割が与えられた場合には，この方法は使えない.

例：アイテムセットキューブにおいて、< ページ > 属性の分割を {「講演会のページを見た」,「大会のページを見た」,「ダウンロードのページを見た」,「リンク集のページを見た」} と与えて実体化したとしよう．この分割は互いに素ではない．Web サイトのディレクトリ構造に沿った階層構造がページ属性に与えられているとすると、この階層構造に沿ってこのキューブを変形する操作がロールアップとドリルダウンとなる.

このように，アイテムセットキューブのある属性に互いに素でない分割が与えられている時のロールアップとドリルダウンの効率良い計算手法は自明ではない.

1 つの統一的な解決法は，ロールアップ/ドリルダウンの処理ごとに毎回，もう一度最初のデータ集合から実体化をすることである．この手法を，以下，**初等的手法**と呼ぶ．これは明らかに効率が悪い．そこで以下では，属性の分割が互いに素でない場合で，かつ，属性に概念階層があらかじめ与えられている場合についての効率の良いロールアップとドリルダウンの実行方法を述べる [11].

### 5.2 ロールアップ

ある属性 $A$ に対して，分割を $P = p_1, p_2, p_3, ..., p_N$ といたとき，一次元アイテムセットキューブ [A,P] をロールアップする場合を考える ($N$=4 で以下説明する)．今，アイテム $p_1, p_2, p_3, p_4$ に図 11 のような概念階層があったとする．階層 3 に $p_1, p_2, p_3, p_4$ が属する．階層 2 には，$p_1, p_2$ の上の $p_1 \cup p_2$ と $p_3, p_4$ の上の $p_3 \cup p_4$ が属する．階層 1 には，さらにその上の $(p_1 \cup p_2) \cup (p_3 \cup p_4)$ が属する．図 11 でロールアップする状況として，まず階層 3 を実体化し，その後に階層 2 へ次に階層 1 へとロールアップする場合を考える (図 11)．このとき，最初に実体化する階層 3 を実体化階層と呼び，その実体化階層を基準にしてその上の階層 1，2 を上位階層と呼ぶ．実体化階層のセルと上位階層のセルにおいて足切り値 $\theta\%$ が一定により，上位階層のセルの足切り値に相当するレコード数は，実体化階層のセルのそれより大きくなる．さらに，セル同士のレコードの重複は極めて多いため，実体化対象のセルを実体化する際に，上位階層のセルを含めて実体化しても，ほとんど計算負荷にならない (4.1 の実験 2)．そこで，図 11 の階層 3 を実体化する段階で，上位階層のすべてのセル (図 11 の階層 1，2) を含めて実体化することで，ロールアップの際に高速化が行えるはずである．評価実験に用いるデータセットは，$N = 16$ として，図 11 のように 2 分木で概念階層を与える．階層 1 には 1 個，階層 2 には 2 個，階層 3 には 4 個，階層 4 には 8 個，階層 5 には 16 個のアイテムが属する．このデータセットに対して，階層 5 を実体化する段階で，上位階層 (階層 4，階層 3，階層 2，階層 1) のセルを含めた実体化を行うときの計算時間を実験した (図 12)．これにより，実体化の計算時間のオーバヘッド 10〜20%程度を経ることで，ロールアップの計算時間を消滅することができる.

### 5.3 ドリルダウン

ある属性 $A$ に対して，分割を $P = p_1, p_2, p_3, ..., p_N$ といたとき，一次元アイテムセットキューブ [A,P] をロールアップする場合を考える ($N$=4 で以下説明する)．ロールアップのときと同様にアイテム $p_1, p_2, p_3, p_4$ に図 13 のような概念階層があったとする．図 13 でドリルダウンする状況として，まず階層 1 を実体化し，その後に階層 2 へ次に階層 3 へとドリルダウンする場合を考える (図 13)．このとき，最初に実体化する階層 1 を実体化階層と呼び，その実体化階層を基準にしてその下の階層 2，3 を下位階層と呼ぶ．計算方法として 5.2 節の方法でドリルダウンにおいても高速化が行えるがそのまま適用すると著し

(注1)：ドリルダウンの場合はこの方法は使えない.

く遅くなる．それは，図 13 のようにアイテムに階層構造がある場合，足切り値 $\theta$% が一定により，下位階層のセルの足切り値に相当するレコード数は，実体化階層のセルのそれより小さくなる．つまり，実体化階層のセルを実体化する際に，下位階層のセルを含めて実体化すると，下位のセル毎に新しいアイテムセットの数え上げをすることになるからである．そこで，下位階層 (図 13 の階層 2，3) については，候補アイテムセットのオーバーラップが高くドリルダウンする際に最も計算負荷のかかる長さ 2 の高頻度アイテムセットの生成までを行う．よって，図 13 の階層 1 を実体化する段階で，下位階層のすべてのセル (図 13 の階層 2，3) については，長さ 2 の高頻度アイテムセットまでを計算する．これにより，ドリルダウンする際には，当該セルの長さ 3 の候補アイテムセットの生成から実行するため，最も計算負荷のかかる処理を回避できる．よって，初等的なドリルダウンの $\frac{1}{2}$ 程度の時間で計算することができる (図 14).

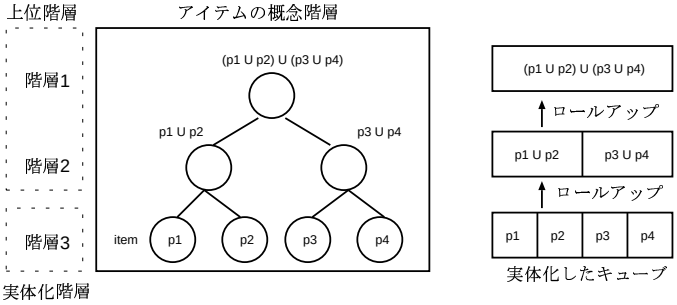


図 11　ロールアップ実験におけるアイテムの概念階層

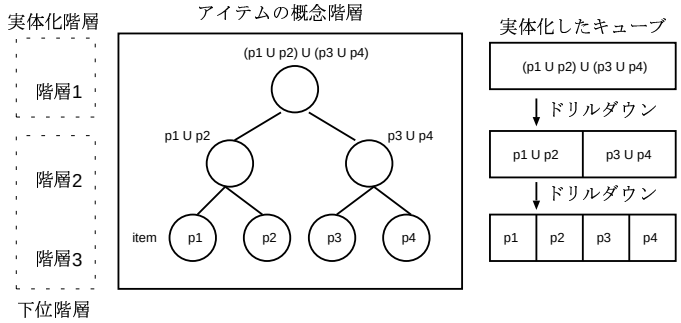


図 13　ドリルダウン実験におけるアイテムの概念階層

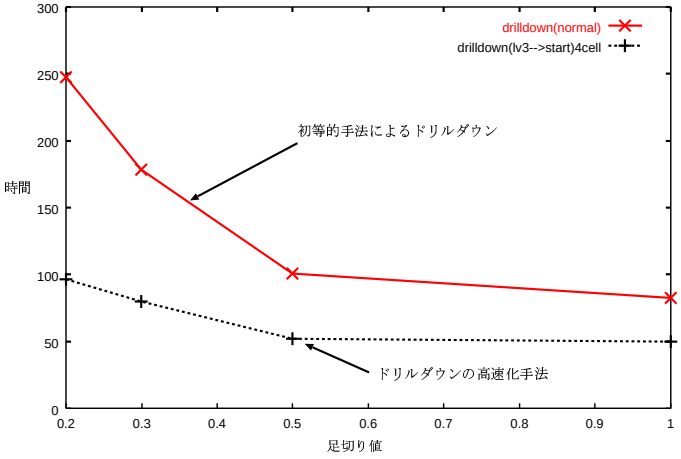


図 14　ドリルダウンの高速化の評価実験　横軸：足切り値　縦軸：時間 (秒)

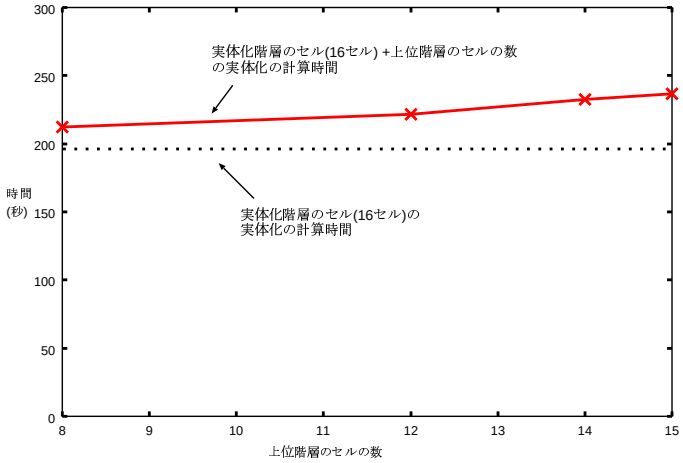


図 12　上位階層のセルを含めた実体化の計算負荷　横軸：上位階層のセルの数　縦軸：時間 (秒)

## 6. ま と め

本稿では，Web ログ分析における複雑な分析条件の変更と分析の繰り返しを支援することを目的として、ログデータ分析時の高頻度パターン数え上げをデータキューブモデルの枠組に沿って行う方式を提案した．

提案されたデータキューブモデルは，分析を行いたい $k$ 個の属性 $A_i$ とその属性に関する分割 $P_i$ $(i = 1, \ldots, k)$ を与えることで構成される．分割とは，当該属性に関する相異なる分析条件を規定するブール述語の集合である．この時，データキューブの各セル，いわゆるキュボイド (cuboid，本稿ではセルと呼んだ) には，値として，そのセルが表す分析条件 (すなわち，分析対象となる各属性に関するブール述語の連接) を満たすようなログレコードの集合から計算された高頻度アイテムセットが与えられる．本稿では，こうして定義されたデータキューブを**アイテムセットキューブ**と呼んでいる．

ログの分析プロセスは，指定したアイテムセットキューブをログデータから計算する演算である**実体化**，および，分析軸についての分割の変更，すなわち，**ロールアップ**や**ドリルダウン**の繰り返しによって支援される．本稿では，アイテムセットキューブと各演算を定義した後に，実際の Web サイト (数百ページ程度) に適用した事例を示した．次に，実体化演算の高速実行手法として，キューブ特有の性質を使った Apriori 法の変形版である Cubic Apriori を提案した．10000 アイテム・30 万件の人工データを使った性能評価では，特に，相異なるセルを同時に満たすようなログレコードが多く存在する場合において、Cubic Apriori 法は，従来手法 (単純にセルごとに Apriori を使う手法) よりも大幅に高速化できることがわかった．

また，本稿では，実体化されたアイテムセットキューブの変形，すなわちロールアップとドリルダウンの実行手法についても，属性について概念階層が与えられている場合について提案した．提案した方法は，概念階層の下ではセルを規定する分析

条件が上述した Cubic Apriori の早くなる状況になっていることを利用して，実体化の時に後続の変形操作に向けた事前処理を極めて低い負荷で計算する．その結果，実体化時のオーバヘッド 10〜20%程度を経ることによってロールアップ時の計算処理は消滅することを示した．また，ドリルダウン時の計算時間は，直接的に実体化する場合に比べて 2-3 倍程度に高速化できた．

現在の試作システムは上記機構を全て組み込んでおり，実際の Web ログ分析を行うことができる．ただし，キューブの変形処理においては，あらかじめ概念階層を与えて指定した属性一つについてしかロールアップ，ドリルダウンはできていない．複数の属性の概念階層を考慮した演算の高速化，および，より大規模なログデータ分析への適用，高頻度パターンを入力とする分析用アプリケーションサーバとの接続などが現在の課題である．

## 文　　献


[1] 松原 健志, ネットパーセプションズのリアルタイム・パーソナリゼーション, ACM SIGMOD 日本支部 第 17 会大会講演論文集, pp. 21-48, 2000.

[2] 小池 寛, 電子ショップにおける優良顧客向けパーソナリゼーション, ACM SIGMOD 日本支部 第 17 会大会講演論文集, pp. 1-20, 2000.

[3] R. Agrawal, et al.， Fast Algorithms for Mining Association Rules, Proc. 20th VLDB, pp.487-499, 1994.

[4] B.Prasetyo , et al.，Naviz:User Behavior Visualizations System using Web Access Log, 情報処理学会第 64 回全国大会 6X-01, 2002.

[5] U. Dayal, Web Content and Usage Mining for E-Commerce Applications, Proc. DBWEB2000, pp.311-316, 2000.

[6] R.Kohavi, et al, KDD cup 2000 organizers' report:peeling the onion, SIGKDD explorations vol.2(issue 2)，pp86-93，Dec，2000.

[7] K.L.Wu, Speed Tracer A Web Usage Mining and Analysis Tool, 電子情報通信学会 データ工学研究専門委員会データウェアハウス・データマイニングの実践フィールド チュートリアル資料集, pp. 3-11, 1998.

[8] 蔦谷雄一, Web ログ分析における対話的パターン検索のためのデータキューブモデル, 電気通信大学大学院情報システム学研究科修士論文, 2002

[9] 蔦谷雄一, 大森匡, 星 守, Web ログ分析における対話的パターン検索のためのデータキューブモデル, 情報処理学会第 64 回全国大会, 6x-04, 2002

[10] Ohmori.T, Tsutatani.Y, Hoshi.M , A Novel Datacube Model Supporting Interactive Web-log Mining, 2002 IEEE Int.Symp. Cyber Worlds:Theory and Practices(CW2002) , 419-427, 2002

[11] 助川貴信, 大規模データベースから高頻度パターンを計算するデータキューブエンジンの高速化に関する研究, 電気通信大学大学院情報システム学研究科修士論文, 2003