D-021

高頻度アイテムセットによる多次元的なログデータ分析を支援するデータキューブ機構*

大森 匡　　成瀬 正英　　星 守

電気通信大学大学院情報システム学研究科

1 はじめに

近年、コンピュータシステムから生成される大量のログデータの分析は重要性を増している。著者らは、ログデータ分析を高頻度アイテムセットに基づいて行う場合を対象に、多次元データ分析で使われる数値用データキューブを拡張した「アイテムセットキューブ」と呼ぶシステムを提案している。このシステムでは、時間やイベント種別などの多次元上で場合わけした空間内で任意の切り口に応じて高頻度アイテムセットによるログデータのパターン分析を行う。本稿では、本システムにより、従来のデータキューブで成功してきた実体化、スライス、ロールアップによるオンライン的な多次元分析手法が、アイテムセットに基づいたログデータ処理においても効果的に行えることを述べる。

2 アイテムセットキューブによる多次元ログ分析

2.1 前提

本稿では、ログデータ分析の代表的な事例としてWebサーバのアクセスログ分析を用いる。

通常、アクセスログは[IPアドレス、ユーザ名、アクセス時刻、ページ]の形を持つ。さらに、このログデータ集合は、「同一ユーザから一定時間内に連続してアクセスされたページの列」を表すセッションレコードに変換されて分析される。

以下、セッションレコードを、[ユーザドメイン、アクセスした時刻、アクセスした月、ページ集合]の4属性で表現し、そのうち、ユーザドメイン(の種類)、アクセスが生じた月(1月、2月、...)、アクセスされたページ集合、の3属性に着目したログ分析を考える。

以下、レコードという時は、セッションレコードを指す。

また、説明のために、ユーザドメインはG_i $(i = 1, 2, 3)$の3種類に分類する。ただし、相異なるG_i, G_jの間には共通するユーザドメインは存在しない。このような分類のことを、「互いに素な分割」と呼び、個々のG_iのことを、(分割における)セグメントと呼ぶ。

「月」属性は、1月から12月に分割する。

また、ページ集合は、E_i = 「あるイベントE_iに関するページ」$(i = 1, 2, 3)$に分割する。ただし、一般に、1つのページは複数のイベントに関連する。そのため、相異なるE_i, E_jに同一のページが属す可能性がある。このような分割を「互いに素でない」と呼ぶ。

*A New Data Cube System for Multi-Dimensional Log-Data Analysis Using Frequent Itemsets, T.Ohmori, M.Naruse, M.Hoshi, U.Electro-Communications

2.2 従来手法: 数値データキューブによる分析

ログ分析を多次元的に行う手法の主流は、データキューブを使ったOLAPである。以下、簡単にその特性を述べる。

図1に、「月」「ユーザグループ」「ページ集合」の3次元からなるデータキューブを示す。キューブの次元1つは、属性1つとその属性に与えられた分割に対応して区切られる。キューブのアトミックな構成要素であるセル(キュボイド)とは、各次元について1つのセグメントが指定された状況、すなわち、これらセグメントの表す論理述語の全次元にわたってのAND積を表す。セルには、そのセルが記述する条件を満たすレコードの総数が入る。

OLAPでは、各次元に与えられる分割は互いに素なものに限られるのが普通である。

OLAPでは、このようなデータキューブをあらかじめ計算しておく。利用者の問い合わせは、「4月におけるE_1, E_2, E_3別のヒット数を求めよ」というように、必ずしもデータキューブの構造とは一致しない。そのため、キューブから必要な部分をスライスで取りだし、必要でない属性(この場合は「ユーザグループ」)について集計(ロールアップ)することで、所望の結果を計算する。

2.3 数値データキューブとアイテムセット分析の組み合わせの問題点

従来の数値用のデータキューブでは、問い合わせに応じて、スライスとロールアップにより、事前に実体化したキューブを変形する。ロールアップ時に元データのスキャンがいらないこともあり、アドホック問い合わせへのオンライン応答に有効である。

ログ分析の立場から見ると、上記のデータキューブによって、任意の多次元空間上の度数分布を計算できる。この度数分布を見た結果、ある状況1においてどのような事象の組み合わせが多いのか、を調べる問題を考える。

例えば、図2のように、数値データキューブによって4月のページ種類別ヒット数を計算した結果を見た時、次のようなログ分析を行いたい：

- Q1. 「ヒット数の多かったイベントに関するページ」にアクセスしたセッションレコードにはどのような事象の組み合わせが多いのか、を調べたい。

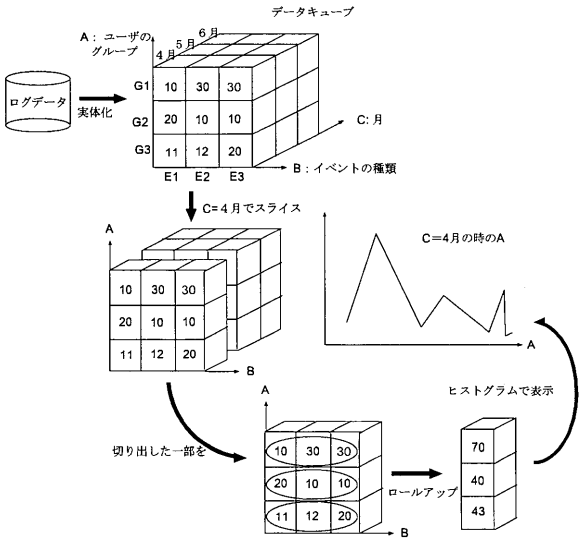

図 1: OLAP/数値データを有すデータキューブ

- Q2. 他のイベントの場合と比べて、Q1 はどのような事象の組み合わせが違うのか、
- Q3. その違いは、時間が経過するとどう変わるのか。

従来、上記のような分析を行うためには、調べたい状況を表すレコード集合から、調べたい対象となる事象を 1 アイテムとして、高頻度アイテムセットを計算する。

今回は、調べたい事象は、「どのページをアクセスしたか」であるので、これを 1 アイテムとして、同一レコードに高頻度に出現するアイテム集合を求めることになる。しかし、図 2 のように、考えうる多様な場合について毎回レコード集合を求めて、そこからアイテムセット計算を行うのでは、オンライン分析にはなっていない。

2.4 提案: アイテムセットキューブによるログ分析

そこで著者らは、高頻度アイテムセットを各セルの値として持つようなデータキューブモデルを提案している。これを「アイテムセットキューブ」と呼ぶ。

アイテムセットキューブは、2.1 節の前提と用語の下で、以下のように定義される：

1. 各次元の分割としては、互いに素でない分割も許す。
2. データキューブの各セルは、従来の数値データではなく、そのセルの条件を満たすようなレコード集合から計算された高頻度アイテムセットを持つ。ただし、支持率 s の下でアイテムセット I が セル c で高頻度であるとは、I が c を満たすレコード総数 N_c に対して、そのセルで $N_c \times s$ 回以上出現することを指す。s は全セルについて共通の定数として与える。

図 3 は、ユーザグループ、月、イベント種別ページ、の 3 属性についてのアイテムセットキューブの例である。このキューブを用いて、「イベント E_3 に関するページがアクセスされた場合において生じる高頻度アイテムセットを、月別に示せ」という問いは、「ページの種類＝ E_3」でスライスを行い、「ユーザドメイン」次元を All へロールアップすることで計算される (図 4)。

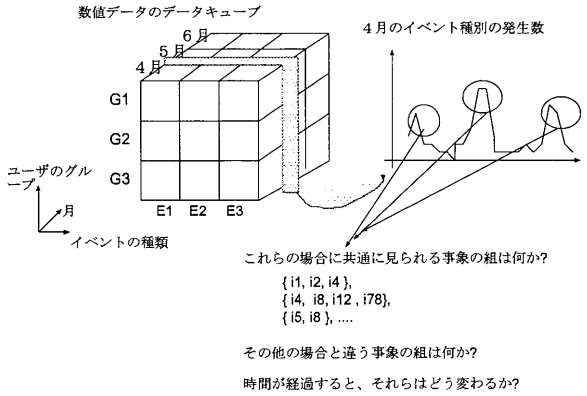

図 2: 度数分布から状況指定でアイテムセット計算を行う場合

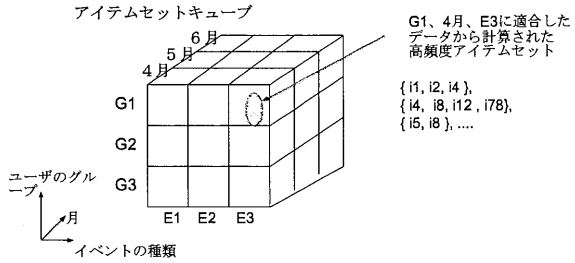

図 3: 3 次元のアイテムセットキューブ

このように、あらかじめ必要なスキーマでアイテムセットキューブを実体化しておくことにより、2.3 節の Q1 から Q3 といった多様なアイテムセット分析にも効率良く答えることができる。

2.5 技術的な問題

上記の枠組みが有効であるためには、計算コストとして次の点が必要である：

1. 実体化が速いこと
2. 概念階層がなくても、アドホックに与えられたロールアップ処理が速く計算できること。

データキューブが数値データを持つ場合には上記は満たされる。しかし、高頻度アイテムセットを有す場合には自明ではない。特に、アイテムセットキューブでは、ある次元の分割が互いに素でない場合があり、そのような属性があるときの実体化やロールアップは自明ではない。

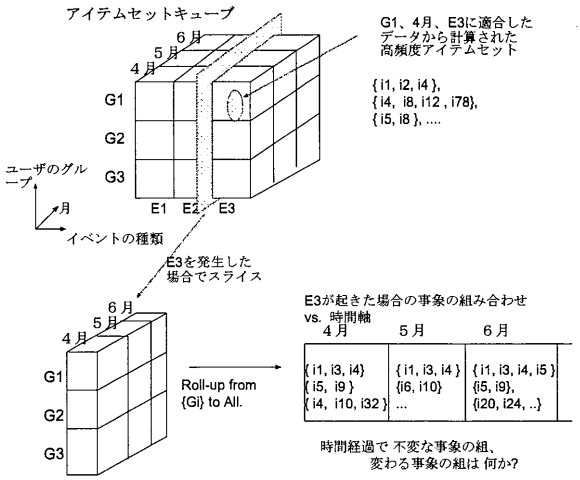

図 4: アイテムセットキューブとそのスライス、ロールアップ結果

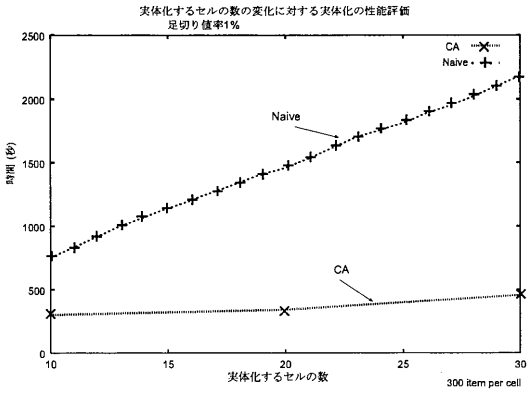

図 5: 実体化計算における 1 次元セル数 vs. 実行時間

3 演算の実行方法

3.1 実体化について

アイテムセットキューブの 1 属性が互いに素でない分割を持つ場合の実体化アルゴリズムは、文献 1 で Cubic Apriori 法 (CA 法) として著者らが提案している。ここでは、その概要を述べ、2 次元、3 次元化した際の結果も示す。

互いに素な分割が与えられた属性については、その分割に応じてレコード集合を分割して、各場合分けについて独自に計算を行う。従って、後は、1 次元の互いに素でない分割により規定されたセル $C_1, C_2, ..., C_n$ について支持率 s で各セルの高頻度アイテムセットを計算することである。

[Cubic Apriori 法]

今、アイテムセット I に、各セル C_i $(i = 1, ..., n)$ での I の出現回数を c_i として、$v(I) = [c_1, c_2, ..., c_n]$ を与える。

手順 1. 各レコード r について長さ 1 のアイテムセット I_1 の数え上げを行い、r が満たすセル C_i についてのみ c_i をインクリメント. 全レコードスキャン後、セルあたりの頻度足切りを行い、長さ 1 の高頻度アイテムセットの集合 L_1 を確定。

手順 2. 長さ $k(\geq 2)$ の候補アイテムセット I_k を 長さ $k-1$ の高頻度アイテムセット L_{k-1} から生成する。I_k が候補となるのは、あるセルにおいて、I_k の長さ $k-1$ の真部分集合が全て高頻度の時に限られる。

手順 3. 各レコード r について、手順 1 と同様に候補 I について r が満たすセル C_i についてのみカウンタ c_i をインクリメント。全レコードスキャン後に、頻度足切りを行い、L_k を確定。L_k が空になるまで、手順 2 へ戻る。
[]

CA 法は、セルごとに個別にアイテムセット計算をしない。そのため、各セルにレコードを分類してから Apriori を行うような単純な手法 (Naive 法と呼ぶ) に比べ、長さ 2 から 3 のアイテムセット処理時の大きなハッシュ表の探索照合処理が 1 回で済む。

図 5 は、互いに素でない分割でセルが決められた場合における、1 次元のアイテムセットキューブを、Naive 法と CA 法で実体化した計算時間である。ただし、支持率 1%, 用いた人工データは、アイテム数 10^4, レコード 30 万件、レコードあたり平均アイテム数 20, 高頻度アイテムセット長設計平均 4、Pentium 4 (1.5GHz)+256MB メモリ である。セルの分け方は、第 i 番目のセグメント＝「アイテム $300i$ 番から 300 個のうちどれかのアイテムを含む」で定義した。セル数増加に対して CA 法が冗長計算を回避できることがわかる。

2 次元、3 次元の場合: 互いに素でない分割を持つ属性からなる 2 次元、3 次元の場合でも、実体化は、当該する多次元上のセルを列挙し 1 次元のセル列と考えることで、CA 法をそのまま適用できる。

上と同じデータとセグメントで次元あたり 10 セグメントあると見なし、第 1 次元の時の 4 セルに別の次元の 4 セルをかけて 2 次元 16 セルにした場合と、さらに第 3 次元の 4 セルをかけて 64 セルにした場合について、CA 法による実体化の計算時間を図 6 に示す (支持率 1%)。数え上げの対象となるレコード数は次元数によらず一定である。高頻度アイテムセット計算の長さ 2 における候補アイテムセット数は、1 次元で 42 万、2 次元で 65 万、3 次元で 125 万程度となっている。

CA 法のメモリ量の最大値は、候補アイテムセット数の最大値 とセル数の積になる。Apriori の戦略をそのまま使っているため、メモリ溢れ対策にも同じ手法が適用可能である。

3.2 実例

図 7 は、実際に分析を行った例である。用いたデータは、ある会員制団体の Web サーバのアクセスログ 2002 年分、セッションレコード数 29588 件である。

分析は、「イベント E_i に関するページのいずれかを見たレコードである」(E_i = 大会、講演会、会員申込、の 3 つ) の条件でレコードを分割し、各季節 (冬 1 月〜3 月、

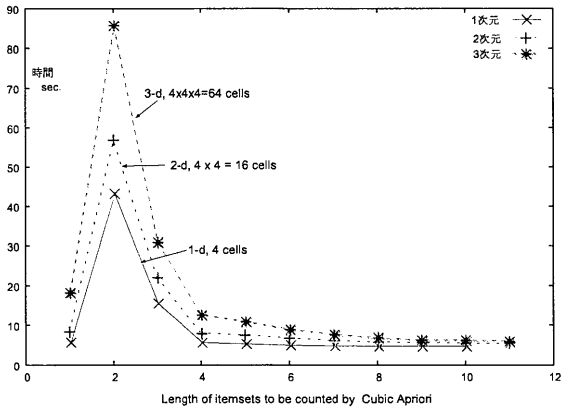

図 6: 次元数 1,2,3 の時の CA 法： アイテムセット長 vs. 実行時間. Pentium 4(2.5GHz) メモリ 1GB.

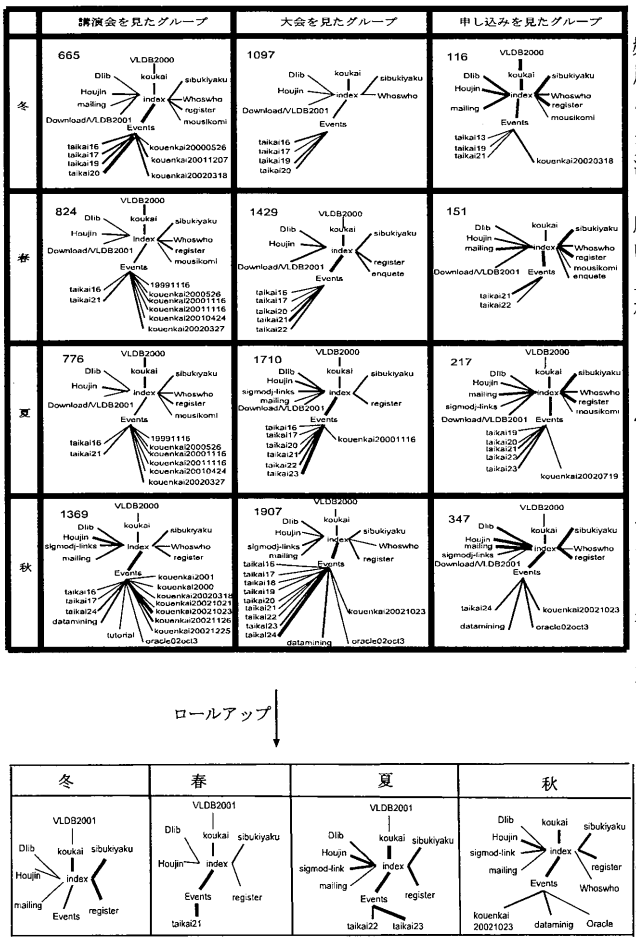

図 7: アイテムセットキューブの実例

春 4 月～6 月、夏 7 月～9 月、秋 10 月～12 月) との 2 次元で実体化した (支持率 1%)。その結果が、図 7 の上にある表である。この表は、セルあたりに計算した高頻度アイテムセットから、Web サイトの構造において高頻度にアクセスされる部分構造を抜きだし描画してある。セル左上スミの数値は、ヒット件数である。

これを、「どのイベントに関するページを見たか」という次元を「All」へとロールアップしたものが図 7 の下部になる。

3.3 ロールアップについて

互いに素な分割が与えられた属性については、相異なるセルに属すレコード集合は共通集合を持たない。このような場合には、各セルの高頻度アイテムセットの集合をマージして作ったアイテムセット集合のみが、ロールアップ後の候補アイテムセットになる。なぜなら、共通集合を持たないレコード集合 $D_1, D_2, ..., D_n$ の和 $D_1 \cup D_2 \cup ... \cup D_n$ において支持率 $s\%$ を越える頻度を有すアイテムセット I は、必ず、D_i $(i = 1, 2, \ldots, n)$ のいずれかで $s\%$ を越える頻度を有すからである。

この考えは、文献 4 で発表され、追加データに伴う高頻度アイテムセット計算に用いられている。従って、「月」属性の分割「1 月から 12 月」を四半期にロールアップする処理は、各月の高頻度アイテムセットの和集合をデータベーススキャン 1 回により数え直すだけで原理的には済む。

一方、互いに素でない分割の属性については、上記が成立しない。著者等は、概念階層が与えられた場合について、下位階層のデータキューブを計算する時に同時に上位階層のキューブを計算し、CA 法の特性により計算負荷を削減する手法を文献 1 で述べている。

4 おわりに

本稿では、アイテムセット分析に基づいた多次元的なログデータ分析を行うためのデータキューブ機構「アイテムセットキューブ」の概要を述べ、従来のデータキューブで成功してきた実体化、スライス、ロールアップによるオンライン的な多次元分析手法が、アイテムセットに基づいたログデータ処理においても効果的に行えることを述べた。ロールアップを含めた実装によるいくつかの実ログデータのテストについては稿を改めて報告したい。

参考文献

1. 助川 貴信, 大森 匡, 星 守, 蔦谷 雄一, 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, DEWS2003, 1-A-01, (2003)

2. R.Agrawal, et al., "Fast Algorithms for Mining Association Rules", 20th VLDB, pp.487-499, 1994.

3. S.Chaundhuri, U.Dayal, "An Overviewof Data Warehousing and OLAP technology", SIGMOD Record, Vol.26 (No.1), pp.65-73, 1997.

4. J.S.Park et al., "An Effective Hash based Algorithm for Mining Association Rules", Proc. ACM SIGMOD'95, pp.175-186, 1995.