DEIM Forum 2009 E7-3

Web構造分析を目的とした多次元データマイニング機構の効率化: To 型制約問い合わせの処理方法

張　洪鋒† 大森　匡† 星　守†

† 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: †{hongfeng,omori}@hol.is.uec.ac.jp

あらまし　著者らは、Web 空間のコミュニティ構造分析を多次元的な制約条件の下で実行するデータベースシステムを試作している。このシステムは、コミュニティ構造を、完全 2 部グラフ（コア）を単位として作り出す。本システムは、'From 型' 制約と' To 型' 制約と呼ぶ 2 種類の制約条件を用意しており、時間軸とあわせて 3 次元のデータキューブモデルに基づいて多角的な Web 空間分析を行う。From 型制約は、「どのドメインから見て重要なコアか」という基準であり、To 型制約は、「指定したドメインに関して周囲から見たときに重要なコアはどれか」という基準である。From 型制約の問い合わせ処理については、昨年の著者らの論文 [1] で述べた。本稿では、残っていた To 型制約条件の問い合わせ処理方法として、効率的な差分的計算法を提案する。

キーワード　データウェアハウス, Web マイニング, データキューブ

Efficient Query-Processing Algorithms in a Multi-Dimensional Web-Mining Database System: a Case of 'To'-Constraint Queries

Hongfeng ZHANG†, Tadashi OHMORI†, and Mamoru HOSHI†

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

Abstract　A hot topic in today's Web and database researches is to provide a new tool supporting web-structure mininig under multiple viewpoints. For this objective, the authors have been developing a database system for a multi-dimensional web-mining, on a basis of a new data cube model named an *itemset cube*. This database system provides two types of constraint queries, called 'From'-constraints and 'To'-constraints. The constraints are regarded as two dimensions of a data cube. Queries under these constraints choose appropriate cores (bipartite graphs) in a collected web-graph space and return a ranked list of web communities satisfying user-given intention. This paper describes efficient processing algorithms for the 'To'-constraint queries on this database system.

Key words　data warehouse, Web mining, data cube

1. はじめに

今日の Web 空間データを対象としたデータベースシステム研究において重要とされる課題の一つに、多様な分析視点から Web 空間内のコミュニティマイニングを行う課題がある。この課題は、Web 空間グラフからの何らかの組織（コミュニティ）を発見する問題として始まった [4]〜[7]。最近では、個人や状況に応じた Web 上の情報をパーソナライズして調べることが特に重視されるようになったことに伴い、コミュニティマイニングにおいても、アドホックに利用者から与えられた Web 空間分析用の問い合わせに答えて適切な Web 空間情報やポータルを生成しようとする試みがある [9] [10]。

著者らは、上記のような「多様な分析視点からの Web 空間データからのコミュニティマイニング」を支援することを目的として、多次元制約の下で Web 構造マイニングを行うデータベースシステムを提案してきた [1]〜[3]。このシステムは、「アイテムセットキューブ」と呼ぶデータキューブモデルに基づいている。アイテムセットキューブは、多次元制約条件を表す最小立方格子に、当該条件を満たすデータから計算した高頻度アイテムセットを格納したデータキューブモデルであり、ロールアップやスライス等の論理演算を使って多次元的な高頻度アイテムセット計算を行うモデルである [8]。

著者らは、'05 年から、このアイテムセットキューブに基づいた Web 空間構造分析用のデータベースシステムを試作している [2], [3]。本システムは、Web 空間構造分析でいうコア (完全 2 部グラフ) を高頻度アイテムセットとして求めることを基本として、多次元制約に応じたコアを求めて、それに基づいたコミュニティ構造を答えとして返すデータベースシステムである。

本システムは、コア分析のために用意する制約条件として、「From 型制約」と「To 型制約」と呼ぶ 2 種類を用意している。From 型制約は、「どのドメインから見て重要なコアか」という基準であり、To 型制約は、「指定したドメインに関して周囲から見たときに重要なコアはどれか」という基準である。そして、これらの制約と時間軸の合計 3 次元のデータキューブモデルに基づいて多角的な Web 空間分析を行う。

著者らは、既に、文献 [2], [3] で、特定組織（大学) のドメイン内の Web 空間を対象にして、本システムによるコミュニティ分析の有用性を調べてきた。一方、データキューブとしての問い合わせ処理の効率化については、文献 [1] において、i) 「対象とする Web 空間を多次元制約下でどの程度の細かさで捉え，実体化しておくべきか」 を決めた後、ii) 「多様な多次元制約の組合せに，ロールアップなどのデータキューブ演算でどう対応するか」を論じて、From 型制約の問い合わせ処理については効率的な差分計算方法を述べた。そこで、本稿では、残っていた To 型制約条件の問い合わせ処理方法として、効率的な差分的計算法を提案する。

以下、2 節で関連研究、3 節でアイテムセットキューブに基づいた Web 構造分析方法の概要を述べ、4 節で、From 型/To 型多次元制約問い合わせの処理戦略を述べる。5,6 節で To 型制約問い合わせの処理方法と評価を行い、最後に 7 節で本稿をまとめる。

2. 関 連 研 究

Web 構造分析の研究は、文献 [4] が、あるトピックに関心を持ったページ集合をコミュニティと呼び、Web 空間データをグラフとおいたときの完全 2 部グラフをコミュニティのコアと呼んで、コアを効率的に求める手法を示したことに始まる．それ以後、コミュニティの検索を行う研究 [5] や，コミュニティ間の関連性を表す地図を作成する研究 [6]、イントラネットの場合における構造解析の研究 [7]、完全ではない 2 部グラフ構造をコアと見なす手法などがある．一方、Web 空間データを対象として、関係代数演算の拡張によって多様な問い合わせを記述、実行する研究として文献 [9] がスタンフォード大学で行われている。

3. アイテムセットキューブによる Web 空間構造分析

3.1 アイテムセットキューブによるコア計算

アイテムセットキューブ [8] とは、いつ、どこで、誰がといった多次元制約の下で一定頻度以上起きている事象の組（高頻度アイテムセット）を求めるためのデータキューブモデルであり、図 1 の様に、最小立方格子（セル）に、対応するレコード集合

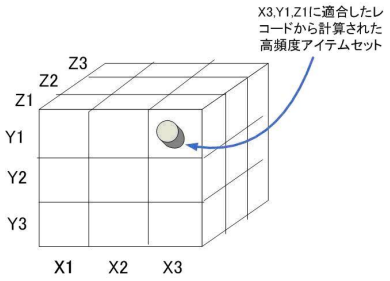

図 1　アイテムセットキューブ

から求めた高頻度アイテムセットを持つ。これを操作することで、多次元分析条件に応じた事象の組を効率よく調べることができる。

一方、Web 構造マイニングの分野では、完全 2 部グラフをコアと呼び、コアに基づいた Web コミュニティ分析が行われる。(有向) 完全 2 部グラフとは、グラフ G について頂点集合 $V(G)$ を 2 つの頂点集合 $V1, V2$ に分割したとき、$V1$ 同士・$V2$ 同士の頂点間には辺が存在しないが、$V1$ の任意点から $V2$ の任意点への有向辺が存在するグラフである。

そこで、我々は、アイテムセットキューブの考えに基づいて多次元制約の下でコア計算を行うことで、多角的な Web 空間の構造分析を行うデータベースシステムを提案してきた [1]〜[3]。

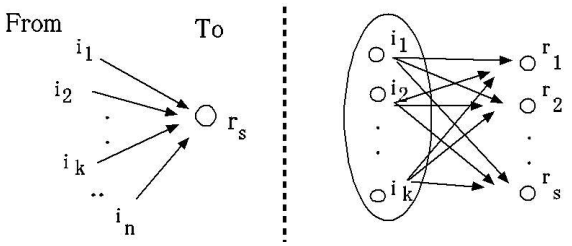

図 2　コアの計算方法

この体系では、始点数 i 終点数 j のコア ((i,j)-コアと表記する) を、高頻度アイテムセットとして求める。そのため、まず、Web リンク構造を表すデータとして、図 2 左側のように、各ページ r_s について、r_s へのリンク入力関係を表すレコード (リンクレコードと呼ぶ) を用意する。つまり、r_s への全ての入方向リンクの始点ページを $i_1, i_2, \ldots, i_k, \ldots, i_n$ として、これらをアイテムとしたレコード $[i_1, i_2, \ldots, i_k, \ldots, i_n]$ を作る。次に、このレコード集合から ($i_1, i_2, \ldots, i_n$ をアイテムとした) 高頻度アイテムセットを求める。すると、$\{i_1, i_2, \ldots i_k\}$ を $k-$ アイテムセットと考えているから、これらを含むリンクレコード数が当該 k-アイテムセットのサポート数になる。その結果、計算するサポート数の最小値 (最小サポート数 s) を決めた時、リンクレコード集合 D において高頻度 (サポート数が s 以上) となるアイテムセットを I とし、I を含む (= 支持する) 全てのレコードの終点ページの集合を A とすると、求まる高頻度アイテムセットは、I を始点集合、A を終点集合としたコアになっている。以下、コアの始点側ノードを、そのコアのハブ (hub) ノードと呼び、コアの終点側ノードを、オーソリティ (authority)

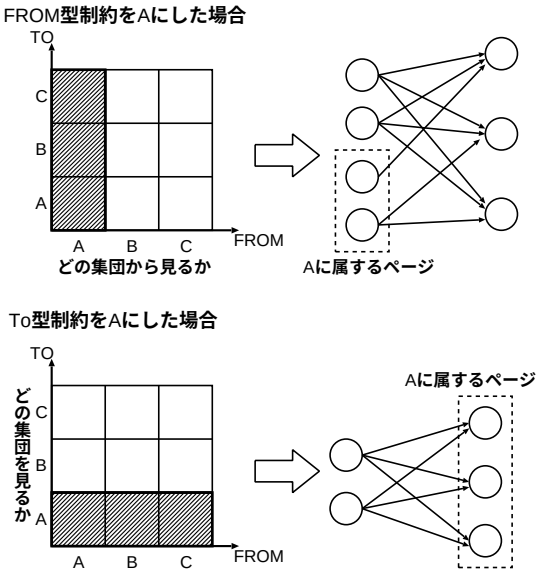

図 3　From 型制約と To 型制約

ノードと呼ぶ。

以上の考え方に基づくと、適当な分析用の多次元制約を与えれば、アイテムセットキューブによる多次元制約下のコア計算を行うことができる。

3.2　Web 構造分析のための多次元制約

ここでは、Web 構造分析用に我々が用意している多次元制約を説明する。

分析対象とする Web 空間は、組織別の階層構造を持つ。 例えば、電気通信大学 (UEC) のトップドメインの下に、情報分野のドメイン、電気系のドメイン、その他の学科系のドメインがある。そこで、我々の研究では、分析用の多次元制約として、これらのドメインにおける、誰にとって誰が重要であるかを分析するために、「From 型制約」と「To 型制約」を定義した [2], [3]。

From 型制約は、「どのドメインのページからリンクを張られているか」に着目した制約である。今、対象とする Web 空間を A, B, C の 3 つのサブドメインにわけて考える。このとき、「From 型制約を A に設定する」(From(A) 制約と表記) とは、「ドメイン A のページを少なくとも 1 つ始点に持つようなリンクレコードだけを対象にしてコアを求める」ことを意味する。こうして求まるコアを、「From(A) 制約を満たすコア」と呼ぶ。図 3 の上部に、その例を示す。(図中、コアは始点数 2、終点数 3 の (2,3)-コアであって、コアの始点が A ドメインとは限らないことに注意してほしい)。 From(A) 制約を満たすコアは、ドメイン A から見たときに重要なコアを表している。

一方、To 型制約は「どのドメインのページにリンクを張っているか」に着目したものである。形式的には、「To 型制約を A に設定する」(To(A) 制約と表記) とは、「ドメイン A のページを終点に持つようなリンクレコードだけを対象にしてコアを求める」ことを意味する。こうして求まるコアを、「To(A) 制約を満たすコア」と呼ぶ。このコアは、ドメイン A に属すページだけを終点として持つコアである (図 3 の下部)。 To(A) 制約を満たすコアは、Web 空間全体からドメイン A を見たときに重要なコアを表している。

結果、多次元制約として From 型/To 型制約を与えて、アイテムセットキューブによる多次元制約下のコア計算ができる。

3.3　コアコミュニティグラフについて

我々のシステムでは、利用者が多次元制約として From 型/To 型制約を与えると、アイテムセットキューブによる多次元制約下のコア計算を行った後に、そのコアを極大コアに直して、その集合から、「コアコミュニティグラフ」とよぶコミュニティ間の関連性を表すグラフを作成して、答えとして返す [2], [3]。(ここで、コア c が極大コアであるとは、c ではないコア c' で「c' のハブノード集合が c のハブ側ノード全てを含み、かつ、c' のオーソリティノード集合が c のオーソリティノード全てを含む」という条件を満たすものが存在しない場合を言う。)

コアコミュニティグラフは、与えられた From 型/To 型制約下で求まった極大コア集合において、強く関連する極大コア同士を併合して 1 ノード（コアコミュニティノード）として、そのノード間の関連性を有向辺で表したグラフである。コアコミュニティグラフにより、グラフ構造に基づいたランキングを行って、重要なコアコミュニティノードを判定できる。

極大コアの併合規則は、次の通り：「極大コア $c1, c2$ がオーソリティノードを 2 つ以上共有するときに、$c1$ と $c2$ を同値とみなす。 この基準で同値類となる極大コア集合を求め、それらを併合して 1 つのコアコミュニティノードとする」。

こうして作ったコアコミュニティノード間に、元のリンク構造を反映した有向辺を追加して、コアコミュニティグラフを作成している。有向辺の付け方やランクづけの計算方法については、文献 [3] で述べている。

図 4 は、2005 年の uec.ac.jp ドメイン下の情報部門でも電気部門でもない部門 Other について From(Other) 制約で求めたコアコミュニティグラフである。番号づけされたタグは、ランク順とコアコミュニティノードが表す組織の内容を示している。

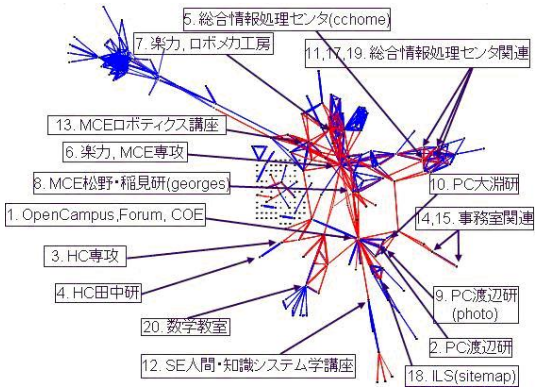

図 4　コアコミュニティグラフ

4.　From 型/To 型制約の組合せの問い合わせの処理問題について

アイテムセットキューブモデルによる多次元制約下のコア計算を効率的に実行する体系を考える。以下、 対象とする Web 空

間を A,B,C の 3 つのサブドメインとし、これらからなる From 型/To 型の 2 次元制約を考える。

まず、図 5 に示す 3 つのアイテムセットキューブを事前に計算する、この処理を、従来のデータキューブの用語に沿って、実体化と呼ぶ。そして、問い合わせに応じて、準備したアイテムセットキューブを使い、応答することを考える。これにより，効率的な応答をしたい。例えば，問い合わせ Q = To(A or B) なら, これを, To(A) と To(B) の各結果であるコア集合から効率的に計算したい。 この操作は，論理的にはデータキューブのロールアップ演算に対応する。

まず、事前に計算しておくアイテムセットキューブは、下記の 3 つである：

D1: From 型/To 型制約無しで実体化を行い，極大コアを格納したアイテムセットキューブ ($b = 8, s = 4$).

D2: 制約を From(A), From(B), From(C) 各々とした場合に実体化を行い、各場合の極大コアを格納したアイテムセットキューブ ($b = 12, s = 4$)。

D3: 制約を To(A), To(B), To(C) 各々とした場合に実体化を行い、各場合の極大コアを格納したアイテムセットキューブ ($b = 12, s = 4$)。

ここで、パラメタ b は、リンクレコードの入辺全てが同一サイト内のリンク (内部リンク) であったとき、その総数が b 以上であれば当該する内部リンク全てをレコードから削除して、その後のリンクレコードを使って極大コアを計算することを意味している。s は、極大コアを計算するときの最小サポート数である。アイテムセットキューブ D1, D2, D3 は、それぞれ、決められたパラメタの下で適当な極大コア集合を計算して保存する。 b パラメタ設定の理由は、昨年の著者らの研究 [1] で述べている。$b = 12$ の方が、$b = 8$ の場合よりも、より細かく内部リンクを考慮して極大コアを計算していることに注意してほしい。また、計算対象とするコアは、始点数 2 以上、終点数は 4 以上のもの ((2,4)-コア) に統一している。

以下、これらの事前に実体化したキューブからロールアップ演算をして、From 型/To 型制約に関する問い合わせに対応していく。

From 型制約の組合せに関する効率化の解決法は、去年 [1] で提案、実装されたので、これから To 型制約問い合わせの処理方法を述べる。

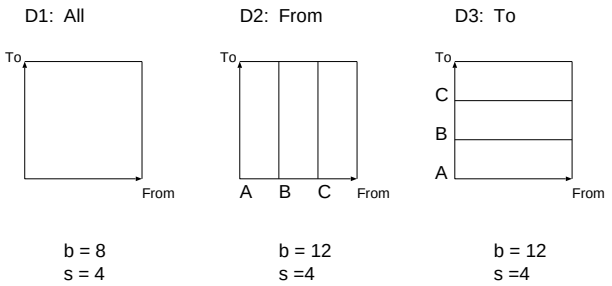

図 5　事前に準備するキューブとそのパラメータ

4.1　用語の定義

（1）「To(A) 制約を満たすレコード」

形式的には、ドメイン A に属すノードがレコード r の終点側ノードとして存在する時、r は To(A) を満たすと定義する。

（2）「To(A) 制約を満た極大コア」

To(A) 制約を満たすレコードのみから計算された極大コア。つまり、図 6 のように、コア自体の終点側ノードが必ずドメイン A のページとなっている極大コア。

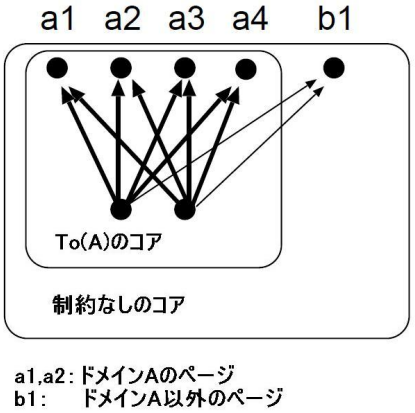

図 6　TO(A) 制約を満たすコア

4.2　問い合わせ処理の戦略

A ドメインまたは B ドメインのページにリンクを張っているコア集合、即ち To(A or B) 制約を満たすコア集合を求める方法を考えてみよう。まず、To(A or B) 制約を満たすリンクレコードのみから実体化して、コア集合を求める方法がある。これを「再実体化法」と呼ぶ。

また、図 5 の D1 のようなアイテムセットキューブに格納している制約なし全ての極大コアが既に計算されているので、これらの極大コアから To(A or B) 制約を満たす極大コアを選出方法もある。これを「フィルタリング法」と呼ぶ。（ただし、$b = 8$ である）。

より細かく内部リンクを考えて詳細に Web 構造を分析する場合（b の値を大きくする時)、リンクレコードの始点側のページ数が大幅に増えるので、図 5 の D1 の様な制約なし全ての極大コアを格納するアイテムセットキューブは、実体化の計算が不可能になる [1]。そのため、より細かく調べたい場合 ($b = 12$) は、D3（や D2）から必要なコア集合を求める。すなわち、問い合わせ Q=To(A or B) に対して、図 5 の D3 のアイテムセットキューブに To(A) 制約を満たす極大コアと To(B) 制約を満たす極大コアが既に算出されているので、これら既に存在している極大コア集合に、一部足りないコアを算出して加えれば、To(A or B) 制約を満たすコア集合を得ることができる。これを「マージ法」と呼ぶ。

以下、これら 2 つの方法を述べる。

5.　To 型制約のフィルタリング法

To 型制約のフィルタリング法とは、制約なしのコア全体から、制約条件を満たさないコアを排除するという考えに従って、

制約を満たすコアを求める方法である。

まず、From/To 制約なし、つまり全てのレコードから計算された極大コアの集合を C_0 とする。また、任意のノード（ページのこと）n に対して、このノードが A ドメインに属しているかどうかを表すフラグ $IsToA[n]$ を用意する。

C_0 の全ての元 c_0 に対して以下の処理 (1)(2) を行うことによって、最小サポート数 s の To(A) 制約を満たす極大コアの集合 C_1 を求めることができる：

（1） c_0 の終点側のノードがドメイン A であるかどうかをチェックする。ドメイン A に属してない終点ノードを全て c_0 から削除する。

（2） 削除した後、終点側のページ数が s 以上なら、このコアを C_1 の元にする。

6. To 型制約のマージ法

6.1 問　　題

複数のドメイン A または B について、To(A) の極大コア集合と To(B) の極大コア集合が既に計算済みと仮定する。このとき、組合せの制約 To(A or B) に対応する極大コア集合を求めたい。

図 7 に示すように、To(A) 制約を満たす極大コア集合を X とおき、To(B) 制約を満たす極大コア集合を Y とおいて、「差分になる極大コア」集合を Z とすると、To(A or B) 制約に対応するコア集合 V は $V = X \cup Y \cup Z$ である。ここで、「差分になる極大コア」というのは、To(A or B) 制約を満たす極大コアであるが、To(A) 制約を満たす極大コアでもなく、To(B) 制約を満たす極大コアでもないコアである。

つまり、ここでの問題は、To(A) 制約を満たす極大コア集合 X と To(B) 制約を満たす極大コア集合 Y が事前に算出されているときに、To(A or B) に応答する時に差分になる極大コア集合 Z だけを計算することである。

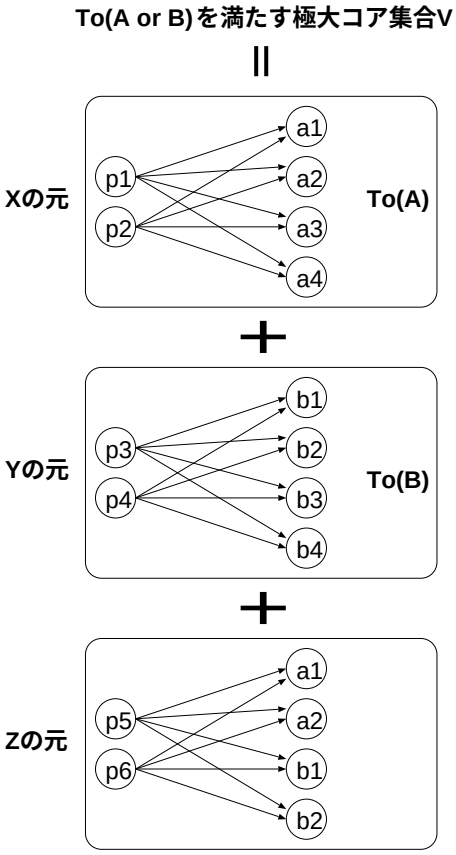

図 7　マージ法の考え方

上述の差分になる極大コア集合 Z の元になる極大コア c は以下の条件を満たすコアである：

「authority 側に A ドメインと B ドメインのページをおのおの 1 つ以上もち、かつ、authority ページとして A ドメインと B ドメインのページしか存在していない、かつ、authority 側のページ数が最小サポート数 s 以上である。」

6.2 Pruning を使った Z の計算法

上述の差分になる極大コア c の authority 側のページが A ドメインと B ドメインのページしか存在していないから、c を算出するため、To(A) を満たす入り辺型リンクレコード集合と To(B) を満たす入り辺型リンクレコード集合の和となるリンクレコード集合を作り、それだけを計算対象に考える。

また、極大コア c は「authority にドメイン A とドメイン B のノードを各々 1 つ以上もち、かつ、A ドメインと B ドメインのノードからなる authority 数が最小サポート数 s 以上ある」コアであるから、c の各 hub ノード h と authority ノード d は、必ず以下の必要条件を全て満たす (図 8 はその条件の例を示す)：

- 条件 1　hub ノード h がさしている終点ノードの集合には、A ドメインと B ドメインのノードを各々 1 つ以上もち、かつ、A ドメインまたは B ドメインとなるノード数は s 以上である。
- 条件 2　authority ノード d をさしている始点ノードの集合には、条件 1 を満たす hub ノード h が少なくとも 2 個以上存在する。

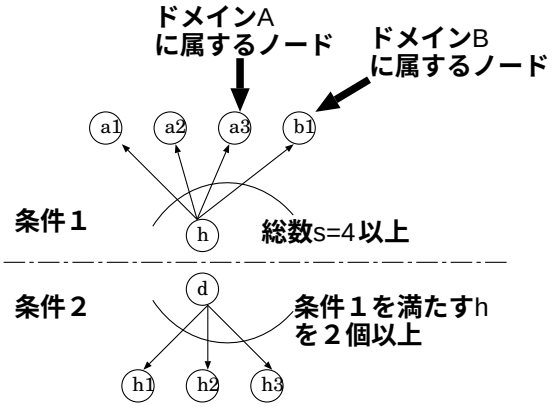

図 8　マージ法の考え方

従って、全体のリンクレコードから、以上の条件を満すコア計

算対象にならないリンクレコードの除去をすれば良い。すなわち、大きさ $(2, s)$ 以上のコアを求めるときに使った「Pruning」戦略を用いる。Pruning 処理でページ間のリンク関係を多く削除できる。Pruning 処理を一回した後に、リンクレコードが変わってしまうので、Pruning 処理を何度もすれば安定になる。

具体的に Pruning 処理は以下の処理に従って行う。(図 9 は各ステップの例を示す)

（1） To(A) または To(B) を満たすレコード集合を S とする。

（2） S をスキャンして、各ノード n に対して、ドメイン A のページを指している出辺の数を $OutToA[n]$ とし、ドメイン B のページを指している出辺の数を $OutToB[n]$ とする

（3） 各ノード n に対して、

- $OutToA[n] \geqq 1$
- かつ　$OutToB[n] \geqq 1$
- かつ　$OutToA[n] + OutToB[n] \geqq 4$

という条件を満たせば $test1[n] = 1$、満たさなければ $test1[n] = 0$ とする。

（4） S をスキャンして、

（a） 各終点ノード n について、その始点ノードを $m_k (k = 1, 2, ...)$ とした時

- $test1[m_k] = 1$ となる要素 m_k が 2 個以上あれば、$test2[n] = 1$ とする
- なければ $test2[n] = 0$ とする。

（b） これらのフラグを付けたら、以下の操作を行う

- $test2[n] = 0$ なら、n の始点ノード m_k を全て削除して、n を孤立点にする。
- $test2[n] = 1$ なら、$test1[m_k] = 0$ となる始点ノード m_k を削除する。

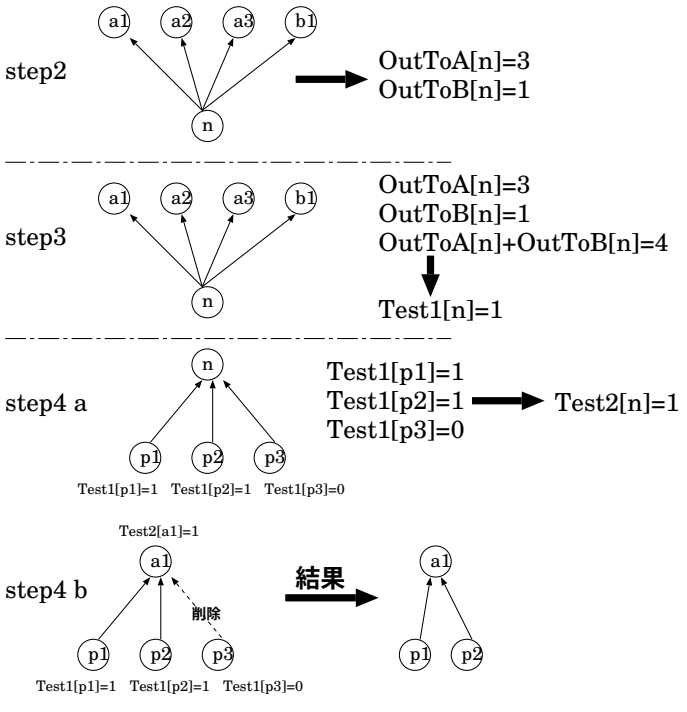

図 9　pruning 過程

これで、Pruning 処理一回が終わった。Pruning を繰り返し、その結果となる S を使って再実体化して求めた極大コア集合は、Z を含んでいる。

6.3　マージ法で To(A or B) 制約に応答する手順

To(A) の極大コア集合 X と To(B) の極大コア集合を Y が事前に算出されているので、マージ法で To(A or B) 制約に応答する流れは以下のようである：

（1） Pruning 処理を行う。

（2） Pruning 処理の出力結果レコードにおいて実体化を行って、Z を求める。

（3） $X \cup Y \cup Z$ を入力として後処理を行って、コアコミュニティノードを求める。

上述の流れで To(A or B) 制約を満たすコアコミュニティノード集合を算出できる。ただし、上の手続きの手順 2 で得られた To(A or B) 制約を満たす差分の極大コア集合 Z と、事前に算出された To(A) 制約を満たす極大コア集合 X 或いは To(B) 制約を満たす Y の間に、「冗長性」が出る可能性がある。例えば図 10 のように、コア c_2 が差分コア集合 Z の元として算出されたとき、To(A) 制約を満たす極大コア集合 X に図 10 下部のようなコア c_1 が存在するため、c_1 は c_2 の一部として含まれていることになる。

後処理でコアコミュニティノードを作成するときに、図 10 の c_1 と c_2 のように冗長になるコアはコアコミュニティノード 1 つにされるので、冗長性は正しく排除される。従って、ここで何もしないで、そのままでも正しい結果を算出することができる。

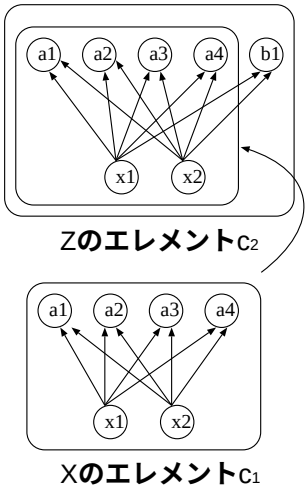

図 10　マージ法の冗長性

6.4　実　　験

UEC05 年の Web 空間データ (ページ数=108096、リンク数=518558) を対象に、枝きり数下限 $b = 8$ 及び $b = 12$ で作成した入辺型リンクレコードにおいて、最小サポート数 $s = 4$ のパ

表 1　Pruning 処理によるレコード数の変化 ($b = 8, s = 4$)

制約	回数	入力	削除	残る
To(ISJC or EE)	1	29185	29090	95
	2	95	7	88
	3	88	0	88
	4	88	0	88
	5	88	0	88
To(ISJC or Other)	1	30118	29774	344
	2	344	56	288
	3	288	0	288
	4	288	0	288
	5	288	0	288
To(EE or Other)	1	31253	30819	434
	2	434	38	396
	3	396	14	382
	4	382	1	381
	5	381	1	380

表 2　Pruning 処理によるレコード数の変化 ($b = 12, s = 4$)

制約	回数	入力	削除	残る
To(ISJC or EE)	1	44690	44583	107
	2	107	7	100
	3	100	0	100
	4	100	0	100
	5	100	0	100
To(ISJC or Other)	1	42042	41688	354
	2	354	65	289
	3	289	0	289
	4	289	0	289
	5	289	0	289
To(EE or Other)	1	39692	39224	468
	2	468	38	430
	3	430	17	413
	4	413	1	412
	5	412	1	411

ラメタで、To(ISJC or Other)、To(ISJC or EE)、To(EE or Other)(注1) 制約を満たす極大コア集合をマージ法でそれぞれ求めて見た。 表 1 は、$b = 8$ の場合 Pruning 処理によってマージ法の実体化の計算対象になるレコード数の変化を示す。表 2 は、$b = 12$ の場合 Pruning 処理によってマージ法の実体化の計算対象になるレコード数の変化を示す。「入力」は入力レコード数を表し、「削除」は Pruning 処理によってなくなるレコード数を表し、「残る」は Pruning 処理の結果として出力されたレコードの数を表す。マージ法では差分コア集合しか再計算しないので、表 1 に示すように、5 回の Pruning 処理をした後に、$b = 8$ の場合、マージ法で実体化対象になるレコード数はそれぞれ 88 件、288 件、380 件になった、$b = 12$ の場合、マージ法で実体化対象になるレコード数はそれぞれ 100 件、289 件、411 件になった。

再実体化法の場合、Pruning 処理をしないので、実体化の計算対象になるレコードは、求めた To 型制約を満たす全てのレコードである。即ち、再実体化法での実体化対象になるレコードの数はマージ法での一回目の Pruning 処理前のレコード数と同じである。表 1 と表 2 によって、マージ法の実体化対象になるレコード数は再実体化法のそれより大幅に減少した事が分かった。

(注1)：ISJC:情報分野ドメイン　EE:電気系ドメイン　Other:その他の学科ドメイン

表 3　マージ法の計算時間 ($b = 8, s = 4$)(単位：秒)

制約	Pruning 処理	実体化	後処理	合計
To(ISJC or EE)	1	6	8	15
To(ISJC or Other)	1	3	22	26
To(EE or Other)	1	2	16	19

表 4　マージ法の計算時間 ($b = 12, s = 4$)(単位：秒)

制約	Pruning 処理	実体化	後処理	合計
To(ISJC or EE)	6	8	615	629
To(ISJC or Other)	5	5	615	625
To(EE or Other)	5	2	292	299

表 5　再実体化法の計算時間 ($b = 8, s = 4$)(単位：秒)

制約	実体化	後処理	合計
To(ISJC or EE)	24	30	54
To(ISJC or Other)	126	48	174
To(EE or Other)	167	39	206

表 3 と表 4 に、$b = 8$ と $b = 12$ の場合のマージ法の計算時間をそれぞれ示す。各表の「後処理」欄は、極大コア集合を入力としてのコアコミュニティグラフを作成する処理である。比較として、表 5 に、$b = 8$ の場合再実体化法の計算時間を示す。($b = 12$ の場合、再実体化法は計算不可能のため省略) 表 3 と表 5 によると、マージ法での計算時間は再実体化法の時間より大幅に減少した事が分かる。これは、マージ法で Pruning 処理を行っているので、実体化の計算対象になるレコードの数と大きさが大幅に減少しているからである。また、マージ法で算出したコアコミュニティノード集合は再実体化法で算出した結果と一致することを確認している。

6.5　From 型/To 型制約の組合せの問い合わせの応答法の試行

最後に、上で述べた To 型制約の処理方法と文献 [1] で述べた From 型制約の効率的な処理方法とを合わせて、図 5 の様に事前に算出されているアイテムセットキューブを使って、From 型/To 型制約両方の組合せの問い合わせに対する効率的な応答法を試した。例えば問い合わせ Q = From(A or B) And To(A or B) を求める時、応答方法として次の方法にした (図 11)：

[提案方法] まず、既に D3 で算出されている To(A) 制約を満たす極大コア集合に、From(A or B) 制約のフィルタリング法 [1] を行い、From(A or B) And To(A) 制約を満たす極大コア集合を求める。次に、To(B) 制約を満たす極大コア集合に、From(A or B) 制約のフィルタリング法を行い、From(A or B) And To(B) 制約を満たす極大コア集合を求める。最後

表 6　From(ISJC or Other) And To(ISJC or Other) の実行時間 (b=8 s=4) (秒)

	極大コア計算	後処理	合計
提案方法	7	11	18
再実体化法	34	11	45

表 7　From(ISJC or Other) And To(ISJC or Other) の実行時間 (b=12 s=4) (秒)

	極大コア計算	後処理	合計
提案方法	10	617	627
再実体化法	計算不可能	—	—

に、From(A or B) And To(A) と From(A or B) And To(B) に To 型制約のマージ法を行って、From(A or B) And To(A or B) を求める。

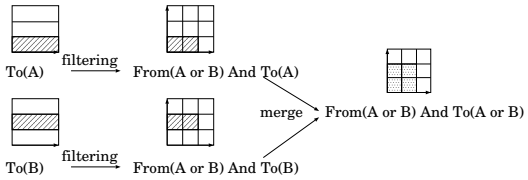

図 11　多次元的な問い合わせの応答方法

上記の方法に従って、UEC05 年の Web 空間データを対象に、枝きり数下限 $b = 8$ のパラメータで Q = From(ISJC or Other) And To(ISJC or Other) 制約の問い合わせ処理を行った。表 6 に、$b = 8$ の時の From(ISJC or Other) And To(ISJC or Other) の応答時間を示す。提案方法によって計算されたコアコミュニティグラフが、再実体化法の計算結果と一致する事は確認されている。表の「極大コア計算」時間は、上記の方法によって計算する時に、To(ISJC) から From(ISJC or Other) And To(ISJC) へのフィルタリング時間と、To(Other) から From(ISJC or Other) And To(Other) へのフィルタリング時間と最後のマージ処理時間を全て含んでいる。表 6 によると、図 11 に示す方法は再実体化法より、計算時間は効率的である。表 7 に、$b = 12$ の時の From(ISJC or Other) And To(ISJC or Other) の応答時間を示す。$b = 12$ の時でも極大コア計算は 10 秒以下であり、図 11 の考え方が妥当であると言える。もちろん、予め実体化しておくアイテムセットキューブの範囲によって異なる方法があり得るので、これらの評価が今後の課題である。

7. おわりに

本稿では、著者らが提案してきた Web 構造分析を目的した多次元データマイニング機構「アイテムセットキューブ」において、To 型制約に対する「フィルタリング法」と「マージ法」の 2 種類のロールアップ演算実行手法を提案し、電気通信大学の Web 空間データを使って評価した。その結果、To 型制約の組み合わせの問い合わせに対して、提案された手法は直接再実体化より効率的に応答することが示された。

本稿で提案された To 型制約の「フィルタリング法」と「マージ法」により、様々な To 型制約の組み合わせの問い合わせに効率的に応答できる。既に、文献 [1] で From 型制約の組合せの問い合わせに効率的に応答する手法も提案され、実装されている。従って、これらの手法を組合わせて、From(A or B) And To(A or B) の様な From 型制約と To 型制約両方の組合せの問い合わせにも効率的に答えることができるはずである。$b = 12$ で実体化したキューブ上のテストも含めて、多様な制約条件の下での性能評価が現在の課題である。

文　献

[1] 栗原 大輔, 大森 匡, 星 守, “Web 構造分析を目的とした多次元データマイニング機構の効率化,” DBSJ Letters Vol.7, No.1, pp.25-30, (from DEWS2008 D1-6), 2008.

[2] 山下 由展, 大森 匡, 星 守, “多次元データマイニングを用いた Web 空間の構造解析,” 電子情報通信学会 DEWS2006, 3B-o3, 2006.

[3] 林 和宏, 大森 匡, 山下 由展, 星 守, “多次元データマイニングによる Web 空間の構造解析の評価,” DBSJ Letters Vol.6, No.1, (from DEWS2007 B8-5) 2007.

[4] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, “Trawling the Web for emerging cyber-communities,” WWW8/Computer Networks, Vol.31(11-16), pp.1481–1493, 1999.

[5] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, “Extracting large-scale knowledge bases from the web,” In Proc. of the 25th VLDB Conference, pp.639–650, 1999.

[6] 豊田 正史, 吉田 聡, 喜連川 優, “ウェブコミュニティチャート: 膨大なウェブページを関連する話題を通して閲覧可能にするツール,” 電子情報通信学会論文誌, D-1 Vol. J87-D-1 No.2, pp.256–265, 2004.

[7] 大塚 浩司, 大町 真一郎, 阿曽 弘具, “ウェブコミュニティ内のページ・リンクから成る階層構造の抽出,” 電子情報通信学会, WI2-2006-33, pp.123–128, 2006.

[8] 成瀬 正英, 大森 匡, 星 守, “多次元的なログデータマイニングを実現するデータキューブ機構の提案と評価,” 電子情報通信学会, DEWS2005, 3C-i10, 2005.

[9] S.Raghavan, H.Garcia-Molina, “Complex Queries over Web Repositories,” VLDB 2003, pp.33-44, 2003.

[10] P.DeRose, et al., “Building Community Wikipedias: A Machine-Human Partnership Approach,” in ICDE 2008, pp.646-655, 2008.