2003－DBS－131　(42)
2003／7／17

頻出グラフマイニング手法の一般化に関する研究

猪口明博[†,††]　鷲尾隆[††]　元田浩[††]

グラフデータに埋もれた特徴的なパターンを取り出すグラフマイニングは出力パターンの数の膨大さと NP 完全として知られる部分グラフ同型問題を含むので，非常に膨大な計算コストを要する．特徴的なパターンを取り出す手法が幾つか提案されているが，それらの手法はある特定のグラフパターンのみしか取り出すことができない．本稿では，大規模データベースから実用的な時間で様々な多頻度グラフパターンを全て抽出する手法を提案する．提案手法の効率の良さは，グラフのデータ表現と出力パターンを制限するバイアスに由来する．提案手法は実データを用いて，計算時間とデータ量の観点から評価される．

Generalization for Frequent Subgraph Mining

Akihiro Inokuchi,[†,††] Takashi Washio[††] and Hiroshi Motoda[††]

Data mining to derive frequent subgraphs from a dataset of general graphs has high computational complexity because it includes the explosively combinatorial search for candidate subgraphs and subgraph isomorphism matching. Although some approaches have been proposed to derive characteristic patters from graph structured data, they limit the graphs to be searched within a specific class. In this paper, we propose an approach to conduct a complete search of various classes of frequent subgraphs in a massive dataset of labeled graphs within practical time. The power of our approach comes from the algebraic representation of graphs, its associated operations and well-organized bias constraints to limit the search space efficiently. It performance has been evaluated through real world datasets, and the high scalability of our approach has been confirmed with respect to the amount of data and the computation time.

1. はじめに

グラフデータに埋もれた特徴的なパターンを取り出すグラフマイニングには多くの応用分野があり，近年注目され始めている．特徴的なパターンを取り出す問題は，出力パターンの数の膨大さと NP 完全として知られる部分グラフ同型問題を含むので，非常に膨大な計算コストを要する．例えば，WARMR は帰納論理プログラミングに基づいて，頻出する全てのパターンを出力する手法であるが，WARMR が実用時間内で取り出すことができたのは，述語数にして 6，頂点数にして 3 個程度のパターンであった[3]．計算コストの問題を対処するため，出力されるパターンにある種の緩和を導入した手法が幾つか提案されている．SUBDUE[2] や GBI[16),20)] は完全探索ではなく，Greedy 探索を用いて，高速に特徴的なパターンを取り出す．しかし，これらの手法は完全探索を必要とする場合には適用できず，さらに重要なパターンを取り出せない可能性を含んでいる．一方，MolFea は version space 法と呼ばれる，データに矛盾しない規則の上界と下界の集合を常に保持しながら解を探索する手法を用いて，グラフ構造データからパスパターンを取り出す．MolFea の原理自体はグラフパターンの抽出にも適用できるが，実用の面から MolFea はグラフのデータ集合から閉路や枝分かれがないパスの抽出についてのみ議論されている[4]．

近年，完全探索を採用し，グラフ，木の集合から全ての頻出パターンを取り出す手法が提案されている．それぞれの手法は対象とするパターンを効率良く取り出すことができるが，取り出されるパターンは特定の構造に限定される．TreeMiner[21] や FREQT[1] は順序木や一般の木の集合に含まれる多頻度 (順序) 木パターンを取り出すことができるが，それらはラベル付きグラフのような，より複雑な構造には適用できない．一方，AGM[7),9)]，FSG[13]，gSpan[19] はラベル付きグ

† 日本アイ・ビー・エム株式会社 東京基礎研究所
Tokyo Research Laboratory, IBM Japan
†† 大阪大学産業科学研究所
Institute of Scientific and Industrial Research, Osaka Univ.

ラフから多頻度部分グラフパターンを取り出すことができる．ラベル付きグラフは順序木，木，パスよりも一般的な構造であるので，それらの手法は木構造の抽出問題にもそのまま適用できるが，それらのパターンを取り出すために最適化されているわけではないので，パターンを効率良く取り出すことはできない．

本稿では，特定のグラフパターンではなく，様々なグラフパターンを抽出するための汎用で，効率良い手法を提案する．提案した手法に追加のバイアスを適用することで，連結グラフパターン，順序木パターンなど様々なグラフパターンを取り出すことが可能である．我々はこの手法を *B-AGM(Biased Apriori based Graph Mining)* と呼ぶ．本稿の後半では提案手法を実装し，他の特定のパターンを取り出すために最適化された手法と実データを用いて比較実験を行う．

2. 問題定義と AGM アルゴリズム

我々は，提案手法の中で AGM アルゴリズムの基本原理を利用する．AGM に，ある特定のバイアスを加えることで，B-AGM は様々な多頻度部分グラフパターンを抽出できる．本節では，我々が対象とする問題を定義し，AGM の基本的なアルゴリズムを紹介する．次節において，様々な種類のグラフのマイニングを可能にするバイアスを提案する．

2.1 グラフ構造データ

多頻度グラフマイニングの入力は頂点の集合 V，辺の集合 E，頂点ラベルの集合 L_V，辺ラベルの集合 L_E からなるグラフの集合である．V, E, L_V, L_E が $V = \{v_1, v_2, \ldots, v_k\}$，$E = \{e_h = (v_i, v_j) | v_i, v_j \in V\}$，$L_V = \{lb(v_i) | \forall v_i \in V\}$，$L_E = \{lb(e_h) | \forall e_h \in E\}$ と与えられたとき，グラフ G は $G = (V, E, L_V, L_E)$ と表される．ここでは，同一グラフ中の複数の頂点（辺）が同一のラベルを持つ場合がある．表記の利便のために，グラフ G の頂点，辺，頂点ラベル，辺ラベルの集合を $V(G), E(G), L_V(G), L_E(G)$ と表す．グラフの頂点数をグラフのサイズと呼ぶ．

グラフの構造は隣接行列を用いて表現できる．$num(lb(e_h)), num(lb(v_i))$ をそれぞれ辺ラベル，頂点ラベルに割り当てられた整数値をする．サイズが k のグラフ G が与えられたとき，その隣接行列 X_k の (i,j)-要素 $x_{i,j}$ は

$$x_{i,j} = \begin{cases} num(lb(e_h)) & \text{if } e_h = (v_i, v_j) \in E(G) \\ 0 & \text{if } (v_i, v_j) \notin E(G) \end{cases},$$

となる．ここで，$i, j \in \{1, \cdots, k\}$ である．隣接行列の i 行（i 列）に相当する頂点を第 i 頂点と呼ぶ．隣接行列 X_k のグラフ構造を $G(X_k)$ と表す．同一のグラフに対応する隣接行列は行（列）の割り当て方で複数存在する．そこで，同一のグラフを一意に表現するためにある特定の条件を満たす隣接行列を正準形 (canonical form) と呼ぶ．正準形を定義するために，隣接行列 X_k の $code$ を，その要素を用いて

$$code(X_k) = x_{1,2}x_{1,3}x_{2,3}x_{1,4}\cdots x_{k-2,k}x_{k-1,k},$$

と定義する．さらに，頂点ラベルを含めた $CODE$ として，

$$CODE(X_k) = num(lb(v_1))\cdots num(lb(v_k))code(X_k),$$

と定義する．ある同一のグラフと一対一に対応する正準形は最大（最小）の $CODE$ をもつ隣接行列と定義する．

グラフ G, G_s が以下の条件を満たすような関数 ϕ が存在するとき，G_s を G の部分グラフと呼ぶ．

$$\phi : V(G_s) \to V(G),\ i \in V(G_s) \quad lb(i) = lb(\phi(i)),$$
$$i, j \in V(G_s) \quad lb(i,j) = lb(\phi(i), \phi(j)).$$

さらに以下を満たすとき，G_s を G の誘導部分グラフと呼ぶ．

$$\forall u, v \in V(G_s), (u,v) \in E(G_s) \Rightarrow (u,v) \in E(G)).$$

グラフ G が G_s を含むとき，$G_s \sqsubseteq G$ と書く．

グラフデータの集合 GD が与えられたとき，グラフパターン G_s の支持度 $sup(G_s)$ は

$$sup(G_s) = \frac{|\{G_s | G \in GD, G_s \sqsubseteq G\}|}{|GD|}$$

と定義する．ユーザが指定した最小支持度以上の支持度を有するグラフパターンを多頻度グラフと呼ぶ．我々が対象とする問題は，グラフの集合と最小支持度が入力として与えられたとき，全ての多頻度グラフパターンを列挙する問題である[8]．

2.2 AGM アルゴリズム

我々は以前，AGM(Apriori-based Graph Mining) アルゴリズムを提案した[7),9)]．AGM は汎用なアルゴリズムであり，連結グラフパターンだけでなく，非連結グラフパターンも抽出することができる．AGM は Apriori 同様，支持度の逆単調性利用して，サイズが 1 の多頻度グラフパターンから，順にサイズの大きなグラフパターンを抽出して行く．図 1 は AGM アルゴリズムの概略である．はじめに 1×1 の隣接行列が頂点ラベルの種類だけ生成され，C_1 に代入される．次に，多頻度グラフパターンの候補の支持度を求め，多頻度グラフパターンを F_k に代入する．続いて，既に抽出されている大きさ k の多頻度グラフパターンを組み合わせて，大きさ $k+1$ の多頻度グラフの候補を生成する．この操作は C_k が空集合になるまで続けられ，最後に全ての多頻度グラフパターンが出力される．

図 1 の関数 Generate-Candidate は隣接行列結合，部分グラフチェック，正準化の 3 つの部分からなる．隣接行列結合部では，大きさ k の多頻度グラフパターンの隣接行列を結合して，大きさ $k+1$ の多頻度グラフの候補を生成する．2 つの隣接行列 X_k, Y_k が与えられ，以下の条件を全て満たすとき，2 つの隣接行列を結合し，大きさ $k+1$ の隣接行列 Z_{k+1} を生成する．

条件 1 X_k, Y_k が第 k 行，第 k 列の要素以外の要素が全て等しいとき，すなわち

$$X_k = \begin{pmatrix} X_{k-1} & \mathbf{x}_1 \\ \mathbf{x}_2^T & 0 \end{pmatrix}, Y_k = \begin{pmatrix} X_{k-1} & \mathbf{y}_1 \\ \mathbf{y}_2^T & 0 \end{pmatrix}$$

$lb(v_i \in V(G(X_k))) = lb(v_i \in V(G(Y_k)))$
$(i = 1, \cdots, k-1)$

条件 2 X_k が正準形であるとき．

条件 3 $CODE(X_k) \geq CODE(Y_k)$ が満たされるとき．☆．

もし，X_k, Y_k が条件を満たし結合可能な場合，2 つの隣接行列を結合して，Z_{k+1} を生成する．

$$Z_{k+1} = \begin{pmatrix} X_{k-1} & \mathbf{x}_1 & \mathbf{y}_1 \\ \mathbf{x}_2^T & 0 & z_{k,k+1} \\ \mathbf{y}_2^T & z_{k+1,k} & 0 \end{pmatrix},$$

$lb(v_i \in V(G(Z_{k+1}))) = lb(v_i \in V(G(X_k)))$
$(i = 1, \cdots, k)$,
$lb(v_{k+1} \in V(G(Z_{k+1}))) = lb(v_k \in V(G(Y_k)))$.

X_k, Y_k は Z_{k+1} の第 1 生成行列，第 2 生成行列と呼ばれる．Z_{k+1} の 2 つの要素 $z_{k,k+1}, z_{k+1,k}$ は X_k, Y_k の要素からは決めることができない．それらの可能な値として，$G(Z_{k+1})$ の第 k 頂点，第 $k+1$ 頂点の間に辺が無い場合と，あるラベルをもつ辺が存在する場合があるので，Z_{k+1} は $(|L_E(E)|+1)$ 個の隣接行列が生成される．以上のようにして作られた隣接行列を正規形 (normal form) と呼ぶ．

$G(Z_{k+1})$ が多頻度グラフであるための必要条件は，その全ての誘導部分グラフが全て多頻度グラフでなければならない．このチェックは部分グラフチェックで行われ，多頻度グラフパターンの候補数を大幅に減らすことができる[8]．候補となるグラフの隣接行列を生成した後，データベースにアクセスしてそれらの支持度を求める．しかし，正規形の中にも同一のグラフを表す隣接行列が複数個存在するため，正規形の隣接行列の中でどの行列が正準形であるか求める必要がある．この操作は正準化で行われる．部分グラフチェックと正準化の詳細は[7),9)] を参照されたい．

☆ 正準形が $CODE$ の最小値で定義される場合は，$CODE(X_k) \leq CODE(Y_k)$ が満たされるときとなる．

```
// GD : グラフの集合からなるデータベース
// F_k : 大きさ k の多頻度グラフパターンの隣接行列の集合
// C_k : 大きさ k の多頻度グラフパターン候補の
//                                  隣接行列の集合
// minsup : 最小支持度
 0)  Main(GD, minsup){
 1)    C_1 ←{ 大きさ 1 の多頻度グラフの候補の
 2)           隣接行列 };
 3)    k ← 1;
 4)    while(C_k ≠ ∅) {
 5)      Count(GD, C_k);
 6)      F_k ← {c_k ∈ C_k | sup(G(c_k)) ≥ minsup};
 7)      C_{k+1} ← Generate-Candidate(F_k);
 8)      k ← k + 1;
 9)    }
10)    return ⋃_k {f_k ∈ F_k | f_k is canonical};
11)  }
```

図 1 AGM アルゴリズムの概略

全ての正準形が求められた後，実際にデータベースにアクセスして多頻度グラフパターンの候補の支持度を求める．部分グラフ同型問題[5]，部分木同型問題は NP 完全であり，部分順序木同型問題は 2 つの木のサイズの積で求めることができる[11]．

3. 出力を特定するためのバイアス

AGM アルゴリズムは全ての頻出するグラフパターンを抽出する．しかし文献[7),9)] に掲載されている手法は，誘導部分グラフとして含まれるグラフパターンのみが抽出されるようなバイアスを含んでいる．このバイアスはデータベースにアクセスして支持度を求めるときに，グラフパターンがデータベース中のデータに誘導部分グラフとして含まれているかとチェックしている．

本節では，図 2 に示されるように，B-AGM に様々なバイアスを加えることで，連結グラフや順序木など，ある特定のグラフパターンのみが抽出できるように拡張する．抽出したいグラフパターンを指定するためのバイアスは正準形の定義，生成行列の結合条件からなる．

3.1 連結グラフ抽出バイアス

このバイアスを適用すると，連結した多頻度グラフパターンのみを抽出することができる．グラフパターンはデータに対し，部分グラフとして含まれる場合と，誘導部分グラフとして含まれる場合があり，その違いはデータベースにアクセスして，頻度を計算するときの部分グラフ同型マッチングの違いである．

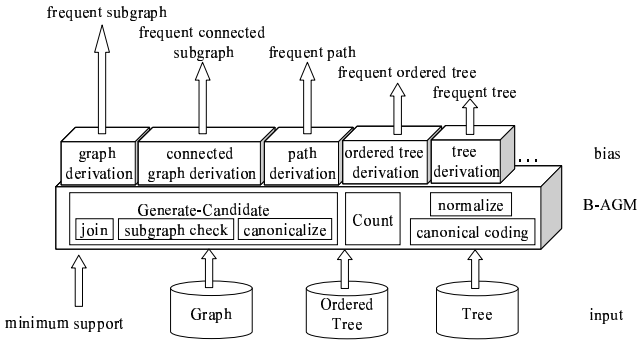

図 2 B-AGM Framework

正準形

隣接行列 X_k の左上側 $i \times i$ の部分行列を X_i $(1 \leq i \leq k)$ とする．同一のグラフ G を表す隣接行列の集合を $\Gamma_1(G)$ を

$$\Gamma_1(G) = \{X_k | G(X_i) \text{ is connected}, \forall i = 1, \cdots, k-1, G \equiv G(X_k)\}$$

と定義すると，グラフ G の正準形 C_k は $\Gamma_1(G)$ の中で $CODE$ が最大になる隣接行列である．

$$C_k \quad w.r.t. \quad CODE(C_k) = \max_{X_k \in \Gamma_1(G)} CODE(X_k)$$

結合の条件

条件 1，2 は従来の AGM と同様で，新たに条件 3，4 が連結グラフパターンを抽出するために追加される．

条件 3 グラフ $G(X_k), G(Y_k)$ の第 k 頂点ラベルが等しい場合は，$code(X_k) \geq code(Y_k)$ が満たされる．等しくない場合は，

$$num(lb(v_k \in V(G(X_k)))) > num(lb(v_k \in V(G(Y_k)))),$$

あるいは，$G(Y_k)$ が非連結であるとき．

条件 4 $G(X_k)$ が連結グラフであるとき．

3.2 パス抽出バイアス

このバイアスは閉路や枝分かれを持たないグラフパターンであるパスを抽出するためのバイアスである．正準形は連結グラフ抽出バイアスと同様に定義される．

結合の条件

条件 1，2 は従来の AGM と同様で，新たに以下の条件 3,4,5 が加えられる．

条件 3 $CODE(X_k) \geq CODE(Y_k)$，あるいは，$G(Y_k)$ が非連結グラフであるとき．

条件 4 $G(X_k)$ が連結グラフであるとき．

条件 5 $G(Y_k)$ が連結である場合は，Z_{k+1} の要素 $z_{k+1,k}, z_{k,k+1}$ は 0 となるとき．

3.3 順序木抽出バイアス

正準形

隣接行列の行 (列) の順番は，順序木に対し前順で頂点 ID を振った場合の順番と一致するものとする．このとき，こような隣接行列を正準形とする．この定義によって，生成される正規形は必ず正準形となり，冗長な隣接行列の生成を避けることで，計算の効率化を計る．

Join Operation

条件 1,2 は従来の AGM と同様で，新たに条件 3,4 が加えられる．

条件 3 $G(Y_k)$ が連結グラフの場合，$code(X_k) \leq code(Y_k)$ が満たされるとき．

条件 4 $G(X_k)$ が連結であるとき．

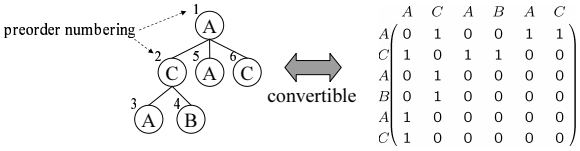

図 3 木構造の $CODE$

3.4 木構造抽出バイアス

図 4 の 2 つの木構造を順序木と考えると 2 つの木は異なるが，一般の木と考えると 2 つのデータは同型である．順序木のバイアスで定義したように順序木の場合は，全順序で頂点 ID を割り振ることで正準形を定義することができるが，一般の木の場合は，前順に頂点 ID を振っても同型な木に対し複数の隣接行列が存在する．従って，正準形を以下のように定義する．

正準形

隣接行列の行 (列) の順番は，順序木に対し前順で頂点 ID を振った場合の順番と一致するものとする．このとき，ある同一の木に対応する隣接行列の集合を $\Gamma_2(G)$ とすると，正準形を

$$C_k \quad w.r.t. \quad CODE(C_k) = \min_{X_k \in \Gamma_2(G)} CODE(X_k)$$

と定義する．ここでは他のバイアスとは違い，正準形は $CODE$ の最小値で定義され，最大値のものは正準形にはなりえない．

例えば，図 4 の 2 つ木構造は同型である．左の木の隣接行列の $CODE$ が最小となるので，左の隣接行列が正準形となる．このバイアスの結合の条件は順序木の同様である．

4. 実　　験

提案手法を C++で実装し．IBM PC 300PL (Windows2000, Pentium III 667MHz, メモリ 192MB)

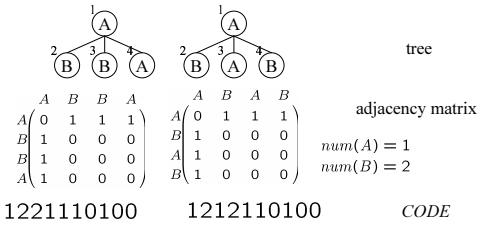

図 4 木のデータ表現

を使い，実データを用いて，他の手法と比較実験を行った．

4.1 連結グラフ抽出

はじめに，発がん性の有無が分かっている 340 個の分子構造を用いた．このデータは Predictive Toxicology Evaluation[18] から入手したデータである．分子の原子，結合，原子の種類，結合の種類をそれぞれグラフの頂点，辺，頂点ラベル，辺ラベルとして扱った．原子の種類は 24 種であったが，同種の原子でも異なる原子タイプのものは，別の頂点ラベルとして扱ったので，頂点ラベルは 66 種となった．頂点ラベルは，単結合，2 重結合，3 重結合，芳香族結合の 4 種，グラフの平均サイズは 27，最大グラフは 214 であった．図 5 は最小支持度を変化させたときの計算時間である．B-AGM は，誘導部分グラフとして含まれる連結グラフパターンの抽出，部分グラフとして含まれる連結グラフパターンの抽出の 2 つの結果がプロットされている．FSG[14]☆，gSpan[19]☆☆は，部分グラフの抽出である．図 5 より，B-AGM は FSG，gSpan と同等の実行速度である．

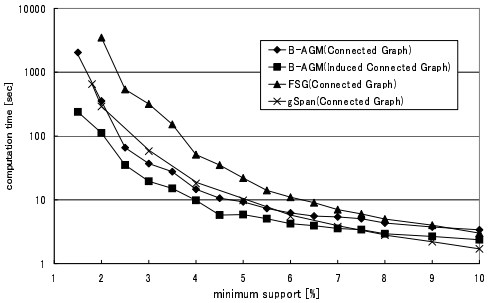

図 5 発ガン性データの計算時間

☆ 実験環境:Linux PC(dual AMD Athlon MP 1800+，メモリ 2GB)

☆☆ 実験環境：Linux PC（Pentium III 500 MHz，メモリ 448MB）

4.2 パス抽出問題

続いて，米国 National Institutes of Health (NIH) から入手できる HIV データを用いた[6]．NIH の DTP AIDS Antiviral Screen プログラムでは数万もの化合物について，HIV 患者への効果を調べている．本実験で利用したデータは，42,687 個の化合物の分子構造に関するデータと，HIV に対する活性の度合いについて Active(以下，CA)，Moderately Active(CM)，Inactive(CI) という情報を含むデータである．CA，CM，CI のデータ数は，422 個，1,081 個，41,184 個である．Kramer らは 41,768 個の化合物データについて，MolFre と呼ばれる手法を適用し，特徴的なパスパターンを取り出している☆☆☆．MolFea は CA に，ある最小支持度以上含まれるかつ，CI（あるいは CM）に最大支持度以下含まれるパターンを取り出した．最小支持度，最大支持度は統計尺度に基づいて決定される．具体的には CA に 13 回（最小支持度 3%）以上，CI に 516 回 (1.282%) 以下含まれるパターンを取り出している．全体の計算時間は約 5 時間 20 分であった．さらに CA に 13 回以上，CM に 8 回 (1.282%) 以下含まれるパターンについても取り出している．全体の計算時間は約 34 分であり，680 個以上のパターンが抽出されている．

同様にして B-AGM を HIV データに適用した．我々の手法は最小支持度しか設定できないので，処理は 2 段階に分けて実行される．はじめに設定された最小支持度を満たす全てのパターンが取り出される．続いて，取り出されたパターンの中で CI（あるいは CM）についても支持度を求め，最大支持度を満たすパターンのみを出力する．データ数が若干異なるため，最小支持度，最大支持度の値は異なるものの，これらの 2 つの閾値は同じ統計尺度のもとで設定された．はじめに CA に 13 個 (最小支持度 3%) 以上含まれ，かつ 516 個 (最大支持度 1.282%) 以下しか含まれないパターンを抽出した．計算時間は 12 分でほぼ同等のパターンを出力できた．さらに CA に 13 個以上含まれ，かつ 516 個 (最大支持度 1.282%) 以下のパターンを取り出した場合での計算時間は約 1 分であった．我々の手法は，最小支持度，最大支持度を設定し特徴的パターンを取り出す問題で，MolFea より遥かに効率よくパターンを取り出すことができるといえる．

4.3 順序木抽出

4.3.1 Web 巡回データ

続いて，順序木パターン抽出について比較実験を

☆☆☆ 実験環境：Linux PC（Pentium III 600 MHz）

行った．用いたデータはZakiによって提供された，あるWebサイトの1ヶ月間のログデータで，各個人の巡回ログは順序木にとして与えられる[21]．前処理後，13,361個のWebページと595,691個の巡回ログからなるデータが得られた．Webページが頂点ラベル，1人のログが1つのデータに相当する．我々は同じデータを用いて，同一計算機でTreeMinerとB-AGMの比較実験を行った．TreeMinerは順序木のデータ集合に含まれる多頻度順序木パターンを高速に発見する手法である．順序木構造に含まれるパターンの定義は，部分グラフの定義とは異なり，パターンPの根から葉へのパスがデータDの同一の道上に同じ順序で存在すれば，PはDに含まれていると定義する．例えば，図6において，左図は順序木データで，右図は順序木パターンPである．PはDの部分グラフではないが，ここではPはDに含まれるものとして考える．3節で提案した順序木抽出バイアスはそのままで，支持度計算（図1の関数Count）のみを変更することで，同じ条件下で比較ができる．図7は最小支持度を変化させたときの計算時間である．図7が示すように，B-AGMはTreeMinerと同等であると言える．

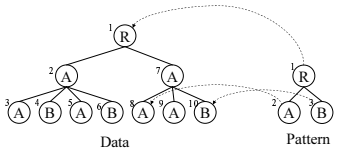

図 6 順序木データとパターンの例 (1)

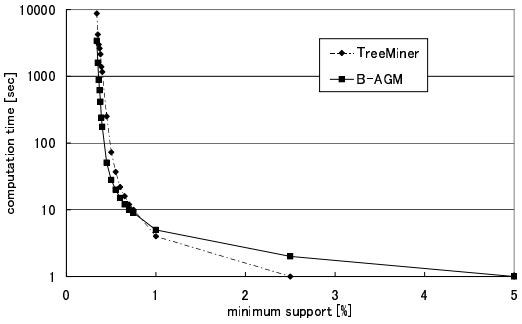

図 7 Web 巡回データの計算時間

4.3.2 半構造データ

浅井らはWebページデータから多頻度部分木を取り出す実験を行った[1]．データはオンライン文献データベースCiteseerから集められたデータで，構文解析されDOMツリーとなっている．前処理後，7,125個の頂点ラベルと196,247個の頂点からなる順序木となる．浅井らによって提供された同じデータとFREQTのJavaプログラムを用いて，比較実験を行った．これまでの問題設定とは異なり，データベース中のデータは1つの順序木である．従って，支持度の定義も若干異なる．[1]で順序木パターンPは支持度は$sup(P)$は全頂点数に対する，パターンPのルート出現回数で定義される．ルール出現とは，例えば，図8，パターンPの根である第1頂点は，データDの2，7で出現しているので，支持度は$sup(P) = 2/10$となる．本実験では比較のために，FREQTの支持度の定義を用いる．前実験度同様に，関数Countのみの変更で対応できる．図9は最小支持度を変化させたときの計算時間と取り出された多頻度順序木パターンの数である．FREQTはJavaで，B-AGMはC++で実装されているので，直接の比較をすることができないが，実行時間のみを考慮すると3倍程度B-AGMの方が高速である．

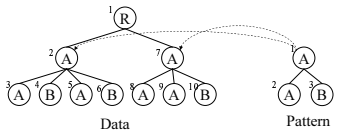

図 8 順序木データとパターンの例 (2)

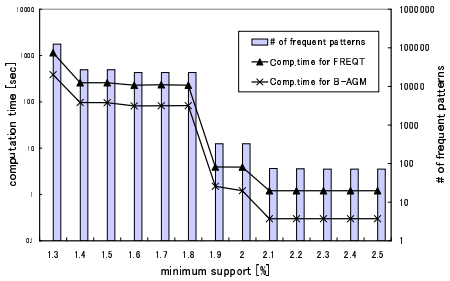

図 9 半構造データの計算時間と多頻度部分グラフの数

5. 関連研究と考察

グラフ構造データから特徴的なパターンを取り出す問題は，膨大な数の候補パターンを考慮しなくてはならないため計算コストが問題となる．SUBDUE[2]やGBI[16),20)]などの手法はある指標に基づいてGreedyにパターンを探索していく．SUBDUEはパターンの

表 1 Summary of Graph Mining Methods

データ	抽出パターン	Approach
グラフ	部分グラフ	AGM, B-AGM
グラフ	連結グラフ	FSG, gSpan, B-AGM
グラフ	パス	MolFea, B-AGM
順序木	順序木	FREQT, TreeMiner,B-AGM
木	木	TreeMiner, B-AGM

最小記述長に基づいて特徴的なパターンを取り出す. GBI の最新の研究では高いスコアをもつ，連結した頂点対を逐次的にチャンキングすることでパターンを取り出す．これらの手法は，ある指標のもとで高速にパターンを取り出すことが可能であるが，Greedy 探索を用いているので重要なパターンを取り出せるという保障はない．

一方, FSG[13], gSpan[19], MolFea[4], TreeMiner[21], FREQT[1] は AGM と同様に完全探索を用いている．FREQT と TreeMiner は頂点数 n の木構造のデータを保持するために $O(n)$ を必要とする．AGM は同じデータを保持するのに $O(n^2)$ を必要とするが，FREQT や TreeMiner はグラフパターンを抽出には適用できない．我々の手法は抽出するパターンに最適化されていないが，他の最適化された手法と同等に動作するので，汎用性を持ちながらも効率的であると言える．表 1 はデータベース中のデータ形式と抽出対象となるパターンのデータ形式をまとめたものである．表が示すように B-AGM は様々な問題に対して適用でき，さらに新たなバイアスを定義することで容易に別の問題も適用できる．

WARMR[3] や FARMR[17] もまた完全探索を用いる手法であるが，それらはグラフよりも汎用なクラスに属する．それらは帰納論理プログラミングで用いられる述語論理を用いてグラフデータを表現し，それらに含まれるパターンを取り出す手法で，述語論理のデータ表現と AGM でも用いられているような levelwise 探索を組み合わせた手法である．データ表現が述語表現であるので，変数を含むようなパターンを取り出せるので，我々の手法より汎用である．今後，これらの手法と我々の手法との関連についても研究していく予定である．

6. むすび

本稿ではグラフ構造データから多頻度パターンをマイニングするための汎用な枠組みを提案した．提案した手法に追加のバイアスを付け加えることで，容易に様々な条件を満たす構造を取り出すことが可能である．提案手法を実装し，特定の出力パターンに最適化された他の手法を比較実験したところ，我々の汎用手法は同等の計算時間で実行することができた．

謝　辞

本稿の実験の遂行にあたり，情報の提供，助言を頂いた九州大学大学院の有村 博紀 助教授，浅井 達哉 氏に感謝いたします．

参 考 文 献

1) Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H. & Arikawa, S. Efficient Substructure Discovery from Large Semi-Structured Data. *Proc. of SIAM Int'l. Conf. on Data Mining*, pp. 158–174, (2002).
2) Cook, D. & Holder, L. Substructure Discovery Using Minimum Description Length and Background Knowledge. *Journal of Artificial Intelligence Research*, Vol.1, pp. 231–255, (1994).
3) Dehaspe, L., Toivonen, H. & King, R. Finding frequent substructures in chemical compounds. *Proc. of Int'l Conf. on Knowledge Discovery and Data Mining*, pp. 30–36, (1998).
4) De Raedt, L. & Kramer, S. The Levelwise Version Space Algorithm and its Application to Molecular Fragment Finding. *Proc. of Int'l. Joint Conf. on Artificial Intelligence*, pp. 853–859, (2001).
5) Garey, M. & Johnson, D. Computers and intractability: a guide to the theory of NP-completeness W.H. Freeman, San Francisco. (1979).
6) AIDS Antiviral Screen, http://dtp.nci.nih.gov/docs/aids/aids_data.html
7) Inokuchi, A., Washio, T. & Motoda, H. An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. *Proc. of European Conf. on Principles and Practice of Knowledge Discovery in Databases*, pp. 13–23, (2000).
8) Inokuchi, A., Washio, T., Nishimura, Y. & Motoda, H. A Fast Algorithm for Mining Frequent Connected Graph. *IBM Research Report* RT0448, (2002), February.
9) Inokuchi, A., Washio, T. & Motoda, H. Complete Mining of Frequent Patterns from Graphs: Mining Graph Data. *Machine Learning*, Vol.50, pp. 321-354, (2003).
10) Inokuchi, A., Washio, T., & Motoda, H. A General Framework for Mining Frequent Patterns in Structures. *IBM Research Report* RT0513, (2003), February.
11) P.Kilpelainen, P. and Mannila, H. Ordered &

Unordered Tree Inclusion. *SIAM Journal on Computing* Vol.24, pp. 340–356, (1995).
12) Kramer, S., De Raedt, L. & Helma, C. Molecular Feature Mining in HIV data. *Proc. of Int'l. Conf. on Knowledge Discovery and Data Mining*, pp. 136–143, (2001).
13) Kuramochi, M. & Karypis, G. Frequent Subgraph Discovery. *Proc. of Int'l. Conf. on Data Mining* pp. 313–320.
14) Kuramochi, M. & Karypis, G. An Efficient Algorithm for Discovering Frequent Subgraphs. *Technical Report* 02-026
15) Matsuda, T., Horiuchi, T., Motoda, H. & Washio, T. Extension of Graph-Based Induction for General Graph Structured Data. *Proc. of Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pp. 420–431, (2000).
16) Motoda, H. & Yoshida, K. Machine Learning Techniques to Make Computers Easier to Use. *Proc. of Int'l. Joint Conf. on Artificial Intelligence*, Vol. 2, pp. 1622–1631, (1997).
17) Nijiissen,S. & Kok,J. Faster Association Rules for Multiple Relations. *Proc. of Int'l. Joint Conf. on Artificial Intelligence*, (2001).
18) PTE, http://oldwww.comlab.ox.ac.uk/oucl/groups/machlearn/PTE
19) Yan, X. & Han, J. gSpan: Graph-Based Substructure Pattern Mining. *Proc. of Int'l. Conf. on Data Mining*, pp. 721–724, (2002).
20) Yoshida, K. & Motoda, H. CLIP: Concept Learning from Inference Patterns. *Artificial Intelligence*, Vol. 75, No. 1 pp. 63–92, (1995).
21) Zaki, M. Efficiently Mining Frequent Trees in a Forest. *Proc. of Int'l. Conf. on Knowledge Discovery and Data Mining*,(2002).