3A2-01

ラベル付き和グラフによるグラフ系列からの頻出パターンマイニングの高速化

Improvement of Graph Sequence Mining by Labeled Union Graphs

猪口 明博[*1] 鷲尾 隆[*2]
Akihiro Inokuchi Takashi Washio

[*1]大阪大学 産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University

[*2]科学技術振興機構 さきがけ
PRESTO, Japan Science and Technology Agency

The mining of a complete set of frequent subgraphs from labeled graph data has been studied extensively. Recently, much attention has been given to frequent pattern mining from graph sequences. In this paper, we propose a method to improve GTRACE which mines frequent patterns called FTSs (Frequent Transformation Subsequences) from graph sequences. Our performance study shows that the proposed method is efficient and scalable for mining both long and large graph sequence patterns, and is some orders of magnitude faster than the conventional method.

1. はじめに

膨大なデータから有用な，あるいは興味のあるパターンを知識として発掘するデータマイニングの研究が盛んに行われている．有用性は人それぞれ異なるので定義するのは難しいが，一般に多くの事例を説明できる知識は有用と考えられる [4]．複数のアイテム集合のデータから頻出アイテム集合を列挙する Apriori アルゴリズムが提案されて以来，様々なデータ構造に対して頻出パターン列挙手法が提案されている．近年では，頂点間連結関係と頂点や辺ラベルの情報からなるグラフ構造に頻出する部分グラフパターン [3] をマイニングする手法が提案されている．提案されているグラフマイニング手法は実用上，非常に効率的であるが，部分グラフ同型問題が NP 完全であるため，より大きな部分グラフをマイニングするのに多くの計算時間を要する．従って，既存手法をグラフ系列のような複数グラフからなる大きなグラフに対して適用することは困難である．

しかしながら，グラフの系列によるモデル化が適している実世界の対象は多く存在する．図 1(a) は 4 状態，5 頂点 ID からなるグラフ系列を示している．例えば，人間関係ネットワークは人が頂点，関係が辺であるグラフで表現でき，人がコミュニティ（ネットワーク）に参加，脱退することで頂点や辺が増減する．同様に，遺伝子が頂点，相互関係が辺である遺伝子ネットワークは，進化の過程で遺伝子が新規獲得されたり，欠落，突然変異するグラフの系列で表現できる．

このようなデータ解析上のニーズを背景として，我々は，グラフ系列をマイニングする手法 GTRACE (Graph TRAnsformation sequenCE mining) を提案し，エンロン社の電子メールデータへ適用した [2]．GTRACE はエンロンデータから生成された 7 状態，100 頂点 ID からなるグラフ系列の集合には適用できたが，それ以上の規模のデータをマイニングするには膨大な計算時間を要した．そこで本稿では，GTRACE の性能を改善するための手法を提案する．

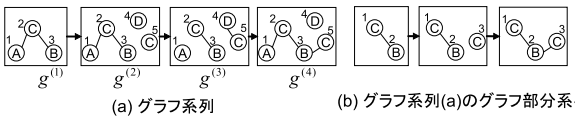

図 1: 観測グラフ系列とそのグラフ部分系列の例

連絡先: 猪口 明博，大阪大学 産業科学研究所，大阪府茨木市美穂ケ丘 8-1，inokuchi@ar.sanken.osaka-u.ac.jp

2. GTRACE

GTRACE は，図 1(a) に示すグラフ系列の集合から，それらに頻出する図 1(b) のような系列を列挙する手法である．GTRACE が対象とするグラフ系列は，以下を満たすグラフの系列である．

- 系列中でグラフの頂点数や辺数が増減する．
- 系列中で頂点ラベルや辺ラベルが変わる．
- 観測グラフ系列の中の連続する 2 つのグラフ $g^{(j)}$ と $g^{(j+1)}$ 間でその構造のごく一部のみが変化する．
- 各グラフは疎グラフである．

例えば，一度に大半の人間や遺伝子が入れ替わることはなく，更に各時点では個々の人間や遺伝子は他の一部としか関係を持たない人間関係ネットワークや遺伝子ネットワークのように，実世界の多くのグラフ変化は，これらの仮定を満たしている．

2.1 グラフ系列の表現形式

グラフ系列中で連続する 2 つのグラフのごく一部が変化するという仮定より，各グラフ $g^{(j)}$ をその全頂点，及びその間の辺で直接表す方法は冗長である．部分系列を効率よく探索するためには，計算コストと空間コストを抑えるためのグラフ系列の簡潔な表現が必要となる．そこで本節では，GTRACE が用いるグラフ系列の表現形式を説明する．

ラベル付きグラフ g を $g = (V, E, L, f)$ で表す．ここで，$V = \{v_1, v_2, \cdots, v_z\}$ は頂点集合，$E = \{(v, v') \mid (v, v') \in V \times V\}$ は辺集合，L は頂点と辺のラベル集合であり，$f : V \rightarrow L$ である．本稿では，頂点のみがラベルをもつグラフを用いて議論するが，本稿で議論する手法は頂点と辺がラベルを持つグラフにも適用可能である．グラフ g の頂点集合，辺集合，ラベル集合を $V(g), E(g), L(g)$ と表す．また観測グラフ系列を $d = \langle g^{(1)}\ g^{(2)} \cdots g^{(n)} \rangle$ と表す．$g^{(j)}$ は j 番目に観測されたグラフである．$g^{(1)}$ を系列の先頭，$g^{(n)}$ を系列の末尾とする．グラフの各頂点 v は ID をもち，$id(v)$ と表す．頂点集合と辺集合に対する ID の集合を以下のように定義する．

$$ID_V(d) = \{id(v) | v \in V(g^{(j)}), g^{(j)} \in d\}$$
$$ID_E(d) = \{(id(v), id(v')) | (v, v') \in E(g^{(j)}), g^{(j)} \in d\}$$

グラフ系列を簡潔に表現するため，グラフ系列中の連続する 2 つのグラフ $g^{(j)}$ と $g^{(j+1)}$ の差異に着目する．

定義 1 観測グラフ系列 $d = \langle g^{(1)} g^{(2)} \cdots g^{(n)} \rangle$ の各グラフ $g^{(j)}$ を外部状態と呼ぶ．さらに，連続する 2 つのグラフ $g^{(j)}$ と $g^{(j+1)}$

表 1: グラフ系列データのための変換規則

頂点追加 $vi^{(j,k)}_{[u,l]}$	ラベルが l，ID が u である頂点を $g^{(j,k)}$ へ追加し，$g^{(j,k+1)}$ へ変換
頂点削除 $vd^{(j,k)}_{[u,l]}$	ラベルが l，ID が u である頂点を $g^{(j,k)}$ から削除し $g^{(j,k+1)}$ へ変換
頂点ラベル変更 $vr^{(j,k)}_{[u,l]}$	ID が u である頂点のラベルを l に変更し，$g^{(j,k)}$ を $g^{(j,k+1)}$ へ変換
辺追加 $ei^{(j,k)}_{[(u_1,u_2),\bullet]}$	ID が u_1 と u_2 である頂点間にラベル l の辺を追加し，$g^{(j,k)}$ を $g^{(j,k+1)}$ へ変換
辺削除 $ed^{(j,k)}_{[(u_1,u_2),\bullet]}$	ID が u_1 と u_2 である頂点間から辺を削除し，$g^{(j,k)}$ を $g^{(j,k+1)}$ へ変換

の間を補間するグラフ系列を $s^{(j)} = \langle g^{(j,1)} g^{(j,2)} \cdots g^{(j,m_j)} \rangle$ で表し，各 $g^{(j,k)}$ を内部状態と呼ぶ．ただし，$g^{(j,1)} = g^{(j)}$ かつ $g^{(j,m_j)} = g^{(j+1)}$ とする．観測グラフ系列 d は補間系列 $d = \langle s^{(1)} s^{(2)} \cdots s^{(n-1)} \rangle$ で表される． ■

外部状態の順序は観測グラフ系列中のグラフの順序であるが，内部状態の順序は人工的に補間されたグラフの順序であり，$g^{(j)}$ と $g^{(j+1)}$ の間に様々な補間系列が考えられる．GTRACE は，グラフ系列マイニングの計算コストと空間コストを抑えるために，グラフ編集距離に基づき最短の補間系列を選択する．

定義 2 頂点や辺の追加，削除，ラベル変更を変換の最小単位とし，それらの変換を編集距離 *1* とする．内部状態系列 $s^{(j)} = \langle g^{(j,1)} \cdots g^{(j,m_j)} \rangle$ の連続する *2* つの内部状態の編集距離は *1* である．また，内部状態系列中の任意の *2* つの内部状態の編集距離は最小である． ■

本稿では，最小単位の変換を変換規則を用いて表す．

定義 3 $g^{(j,k)}$ を $g^{(j,k+1)}$ へ変換する変換規則を $tr^{(j,k)}_{[o_{jk},l_{jk}]}$ で表す．ただし，$o_{jk} \in ID(V) \cup ID(E)$，$l_{jk} \in L$ である．

- tr は頂点や辺の追加，削除，ラベル変更のいずれか．
- o_{jk} は変換される頂点や辺の *ID*.
- l_{jk} は変換される頂点や辺のラベル. ■

本稿では簡単化のため変換規則 $tr^{(j,k)}_{[o_{jk},l_{jk}]}$ を $tr^{(j,k)}_{[o,l]}$ と略記する．GTRACE が用いる 5 種の変換規則を表 1 に示す．例えば，j 番目の外部状態の k 番目と $k+1$ 番目の内部状態間で，ラベルが l で ID が u である頂点の追加を $vi^{(j,k)}_{[u,l]}$ で表す．本稿では頂点のみがラベルをもつグラフについて議論するので，辺に関する変換規則の引数 l はダミー辺数であり，‘•’ で表す．

以上より，変換系列を以下のように定義する．

定義 4 内部状態系列 $s^{(j)} = \langle g^{(j,1)} g^{(j,2)} \cdots g^{(j,m_j)} \rangle$ を変換規則を用いて $seq(s^{(j)}) = \langle tr^{(j,1)}_{[o,l]} tr^{(j,2)}_{[o,l]} \cdots tr^{(j,m_j-1)}_{[o,l]} \rangle$ と表し，内部状態変換系列と呼ぶ．さらに，外部状態系列 $d = \langle g^{(1)} \cdots g^{(n)} \rangle$ を内部状態変換系列の系列である外部状態変換系列 $seq(d) = \langle seq(s^{(1)}) seq(s^{(2)}) \cdots seq(s^{(n-1)}) \rangle$ で表す． ■

このような変換系列によるグラフ系列の表記は，グラフが徐々に変化するという仮定の下で，連続するグラフの差異のみに注目した表現形式であるので，グラフによる直接の系列表記に比べ簡潔である．また，如何なるグラフ系列も表 1 に示す 5 種の変換規則で表現可能である．

2.2 頻出変換部分系列のマイニング

本節ではグラフ系列の集合から頻出変換部分系列をマイニングする手法を示す．2.1 節で説明した外部状態の系列から頻出変換部分系列をマイニングするために，変換系列 $seq(d')$ が変換系列 $seq(d)$ の部分系列であるとき，$seq(d') \sqsubset seq(d)$ と書く．紙面の都合上，詳細な定義を省略するが，詳細は文献 [2] を参照されたい．

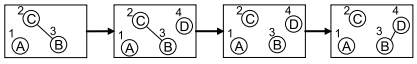

図 2: 関連性のない頂点を含む外部状態系列

GTRACE は，実用性の観点から出力される系列中の頂点と辺が互いに関連がある (relevant) 系列のみを列挙する．例えば，図 2 のグラフ系列では，ラベルが A で ID が 1 である頂点は，どの外部状態においても他の頂点と連結していないため，他の頂点と関連がないと考える．一方，頂点 2 と頂点 4 はどの外部状態においても直接は接続していないが，それらの頂点はラベル B をもつ頂点 3 と，1 番目の外部状態と 4 番目の外部状態でそれぞれ連結している．この場合，本稿では頂点 2 と 4 は頂点 3 を介して互いに関連があると考える．このように，図 2 における関連性のある系列の例として，頂点 2，3，4 を含み，頂点 1 を含まないものが考えられる．以上の外部状態系列の連結性の議論に基づいて，頂点と辺の ID の関連性を以下に定義する．

定義 5 外部状態系列 $d = \langle g^{(1)} \cdots g^{(n)} \rangle$ に対し，ラベルを持たない d の和グラフ $g_u(d) = (V_u, E_u)$ を以下のように定義する．

$$V_u = \{id(v) | v \in V(g^{(j)}), g^{(j)} \in d\}$$
$$E_u = \{(id(v), id(v')) | (v, v') \in E(g^{(j)}), g^{(j)} \in d\}$$ ■

和グラフは変換系列に対しても同様に定義される．外部状態系列 d，あるいは変換系列 $seq(d)$ の和グラフが連結であるとき，d，あるいは $seq(d)$ の ID は互いに関連があると定義する．GTRACE は和グラフが連結である変換系列のみを列挙する．グラフ系列の集合 $DB = \{\langle tid_i, d_i \rangle | d_i = \langle g_i^{(1)} \cdots g_i^{(n_i)} \rangle\}$ に対し，変換部分系列 $seq(d')$ の支持度 $\sigma(seq(d'))$ を

$$\sigma(seq(d')) = |\{tid_i | \langle tid_i, d_i \rangle \in DB, seq(d') \sqsubseteq seq(d_i)\}|$$

と定義する．最小支持度 σ' 以上の支持度を有する部分系列を頻出変換部分系列 (Frequent Transformation Subsequence: FTS) と呼ぶ．関連研究同様，$seq(d'_1) \sqsubset seq(d'_2)$ ならば $\sigma(seq(d'_1)) \geq \sigma(seq(d'_2))$ である支持度の逆単調性が成り立つ．以上の定義により，グラフ系列マイニングを以下のように定義する．

問題 1 グラフ系列の集合 $DB = \{\langle tid_i, d_i \rangle \mid d_i = \langle g_i^{(1)} \cdots g_i^{(n_i)} \rangle\}$ と最小支持度 σ' が入力として与えられたとき，DB 中の *rFTS (relevant FTS)* を全て列挙する．

効率良く関連のある rFTS を全て列挙するため，はじめに，定義 5 に基づいて DB 中のグラフ系列の和グラフを計算する．次に，和グラフの集合から既存グラフマイニング手法 AcGM[3] を用いて，頻出連結部分グラフを取り出す．頻出連結部分グラフが取り出されるたびに，変換系列の包含関係に対応した PrefixSpan を呼び出す．ここで PrefixSpan の入力である変換系列の集合は以下の射影により生成される．

定義 6 グラフ系列 $\langle tid, d \rangle \in DB$ と連結グラフ g が与えられたとき，$seq(d)$ の部分系列に射影する $proj_1$ を以下のように定義する．

$$proj_1(\langle tid, d \rangle, g) = \{\langle tid', seq(d') \rangle \mid$$
$$tid = tid', seq(d') \sqsubseteq seq(d),\ g_u(d') = g,\ \nexists seq(d'')$$
$$s.t.\ (seq(d') \sqsubset sed(d'') \sqsubseteq seq(d) \wedge g_u(d'') = g)\}$$ ■

1) **GTRACE**(DB, σ')
2) $G_u = \{g_u(d) \mid \langle tid, d\rangle \in DB\}$
3) for g =AcGM(G_u, σ'); untill $g \neq null${
4) $proj_1(DB, g) = \bigcup_{\langle tid,d\rangle \in DB} proj_1(\langle tid, d\rangle, g)$
5) $F' =$ PrefixSpan($proj_1(DB, g), \sigma'$)
6) $F = F \cup \{\alpha \mid \alpha \in F' \wedge g_u(\alpha) = g\}$
7) }

図 3: GTRACE の概略

式に示されるように，射影によって得られる部分系列の和グラフは g と同型である．また，この射影により，1 つのグラフ系列 $\langle tid, d\rangle$ から複数の変換規則が出力されることに注意されたい．rFTS の和グラフは $seq(DB)$ の全系列から生成される和グラフ集合 G_u において頻出連結部分グラフとなるので，もし和グラフの集合 G_u から連結な頻出部分グラフ g が得られれば，定義 6 により生成された射影系列から，和グラフが g である rFTS を全て列挙することができる．

図 3 は DB から rFTS を全て列挙するアルゴリズムを示している．はじめに 2 行目で外部状態系列の変換系列集合 $seq(DB)$ の和グラフ集合 G_u を計算する．3 行目の AcGM は G_u から頻出連結部分グラフ g を 1 つずつ出力する関数であり，4 行目において g を用いて射影データを生成し，5 行目で射影データに含まれる頻出変換部分系列を列挙する．最後に，PrefixSpan により列挙された FTS の和グラフが g_u と同型ならば，それを rFTS として出力する．この処理は AcGM が g を出力する限り続けられる．

3. 提案手法: GTRACE2

GTRACE の計算時間のほとんどはその内部で呼ばれる PrefixSpan の計算時間である．その理由は以下の通りである．G_u を $\langle tid, d\rangle \in DB$ から生成される全和グラフ，g を G_u の頻出連結部分グラフとする．和グラフは変換系列の外部状態を重ねて生成するため，各外部状態が疎グラフであっても，和グラフは密グラフになりやすい．G_u はラベルなし密グラフの集合であるので，g は $g_u \in G_u$ の一箇所だけでなく，複数箇所に部分グラフとして出現しやすい．従って，頻出連結部分グラフ g と和グラフが g を含む 1 つグラフ系列 $\langle tid, d\rangle$ から，射影によって多くの変換系列が出力される．PrefixSpan の計算時間は入力の系列数に比例するので，GTRACE の内部で呼び出される PrefixSpan は非常に多くの計算時間を要する．

射影によって出力される変換系列の数を減らすために，和グラフを以下のように再定義する．

定義 7 グラフ系列 d のラベル付き和グラフを $g_u(d) = (V_u, E_u, L \cup \{l_+\}, f_u)$ と定義する．ここで，

$$V_u = \{id(v) \mid v \in V(g^{(j)}), g^{(j)} \in d\}$$
$$E_u = \{(id(v), id(v')) \mid (v, v') \in E(g^{(j)}), g^{(j)} \in d\}$$
$$f_u(o)_{|o\in V_u} = \begin{cases} l & \textit{if グラフ系列}d\textit{中で, }ID\textit{ が}o\textit{である頂点のラベルが常に}l \\ l_+ & \textit{otherwise} \end{cases}$$

L は d の頂点ラベル集合，l_+ は L に含まれない新たなラベルとする． ■

f_u は，グラフ系列 d 中で ID が o である頂点のラベルが変わらず l である場合，和グラフの対応する頂点に l を割り当て，それ以外の頂点に l^+ を割り当てることを示している．

例 1 図 *4* は，同じグラフ系列に対して，定義 *5* により生成される和グラフと定義 *7* によるラベル付き和グラフの例を示している． グラフ系列 d の *ID* が *1* である頂点のラベルは変更されているので，ラベル付き和グラフの *ID* が *1* である頂点はラベル l_+ をもつ．

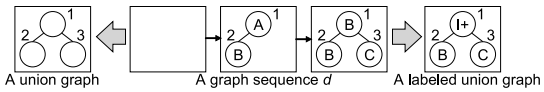

図 4: 和グラフとラベル付き和グラフ

ラベル付き和グラフは変換系列 $seq(d)$ に対しても以下のように定義される．

定義 8 変換系列 $seq(d)$ のラベル付き和グラフを $g_u(seq(d)) = (V_u, E_u, L \cup \{l_+, l_-\}, f_u)$ と定義する．ここで，

$$V_u = \{v \mid tr^{(j,k)}_{[v,l]} \in seq(d), tr \in \{vi, vd, vr\}\} \cup \{v, v' \mid tr^{(j,k)}_{[(v,v'),l]} \in seq(d), tr \in \{ei, ed\}\},$$
$$E_u = \{(v, v') \mid tr^{(j,k)}_{[(v,v'),\bullet]} \in seq(d), tr \in \{ei, ed\}\}.$$
$$f_u(o')_{|o'\in V_u} = \begin{cases} l & \textit{if seq(d)中で，変換規則}tr^{(j,k)}_{[o,l]}\textit{ の }o\textit{ の値が}o'\textit{である変換規則の}l\textit{の値が全て同じ} \\ l_+ & \textit{else if 上記の}l\textit{の値が異なる} \\ l_- & \textit{otherwise} \end{cases}$$ ■

例えば，変換系列 $\langle ei^{(1,1)}_{[(1,2),\bullet]}\rangle$ のラベル付き和グラフは，2 頂点とそれらを結ぶ辺からなり，2 頂点はラベル l_- をもつ．すなわち，l_+ は系列中で対応する頂点のラベルが変更されることを表わし，l_- は系列から頂点ラベルを確定できないことを表わしている．

2.2 節で述べたように，GTRACE ははじめに DB から和グラフの集合 G_u を生成し，G_u から頻出連結部分グラフ g を列挙する．頻出連結部分グラフを列挙する際に，g が $g_u \in G_u$ の部分グラフであるかをチェックする．ラベル付き和グラフで導入された新たな頂点ラベル l_+ をもつ頂点は g のどの頂点にも対応すべきであり，頂点ラベル l_- をもつ頂点は g_u のどの頂点にも対応すべきであるので，AcGM の中の部分グラフ同型性の判定は以下により判定される．2 つのグラフ $g(V, E, L, f)$ と $g'(V', E', L', f')$ が与えられたとき，$v, v_1, v_2 \in V'$ に対して以下を満たす単射 $\phi : V' \to V$ が存在するとき，g' は g の部分グラフである．

1. $(\phi(v_1), \phi(v_2)) \in E$, if $(v_1, v_2) \in E'$
2. $f(\phi(v)) = f'(v)$, $f(\phi(v)) = l_+$, or $f'(v) = l_-$

以上のラベル付き和グラフと部分グラフ同型判定を GTRACE に取り込んだ手法を GTRACE2 と呼ぶ．GTRACE2 は以下に示す補題によりグラフ系列の集合から全ての rFTS を GTRACE よりも効率良く列挙することができる．

補題 1 ラベル付きグラフ g_2 からラベルを除いたグラフを g_1 とするとき，以下が成り立つ．

$$|\cup_{\langle tid,d\rangle \in DB} proj_2(\langle tid, d\rangle, g_2)| \leq |\cup_{\langle tid,d\rangle \in DB} proj_1(\langle tid, d\rangle, g_1)|$$

ただし，$proj_1$ と $proj_2$ はそれぞれ *GTRACE* と *GTRACE2* の射影関数である．さらに，$\langle tid_2, seq(d_2)\rangle$ が $proj_2(\langle tid, d\rangle, g_2)$ に存在するなら，$tid_1 = tid_2$ かつ $seq(d_2) \sqsubseteq seq(d_1)$ を満たす $\langle tid_1, seq(d_1)\rangle$ が $proj_1(\langle tid, d\rangle, g_1)$ の中に必ず存在する．従って，$\cup_{\langle tid,d\rangle \in DB} proj_2(\langle tid, d\rangle, g_2)$ に含まれる変換系列の平均変換規則数は $\cup_{\langle tid,d\rangle \in DB} proj_1(\langle tid, d\rangle, g_1)$ に含まれる変換系列の平均変換規則数以下になる． ■

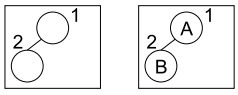

図 5: GTRACE と GTRACE2 の射影の入力グラフ

紙面の都合上，補題 1 の証明は省略するが，具体例を例 2 に示す．文献 [5] で述べられているように，PrefixSpan の計算時間は入力の系列数の増加に対して比例し，系列中の平均アイテム数（本稿の変換系列中の平均変換規則数に相当）の増加に対して指数関数的に増加する．補題 1 より，GTRACE2 の射影により生成される変換系列数と平均変換規則数はそれぞれ GTRACE の変換系列数と平均変換規則数以下であるので，GTRACE2 の中で呼び出される PrefixSpan の計算時間は GTRACE の PrefixSpan の計算時間に比べ減少する．

例 2 図 *4* の中央のグラフ系列は $\langle 1, d\rangle = \langle 1, \langle vi^{(1,1)}_{[1,A]} vi^{(1,2)}_{[2,B]} ei^{(1,3)}_{[(1,2),\bullet]} vi^{(2,1)}_{[3,C]} ei^{(2,2)}_{[(1,3),\bullet]} vr^{(2,3)}_{[1,B]}\rangle\rangle$ で表わされ，その和グラフとラベル付き和グラフはそれぞれ図 *4* の *(a)* と *(b)* になる．図 *5(a)* のグラフ g_1 を *GTRACE* の射影の入力とするとき，g_1 の *2* つの頂点は和グラフ $g_u(d)$ の *ID* が *1* と *2* である頂点に対応するか，*ID* が *1* と *3* である頂点の対応する．従って，$proj_1(\langle 1, d\rangle, g_1)\{\langle 1, \langle vi^{(1,1)}_{[1,A]} vi^{(1,2)}_{[2,B]} ei^{(1,3)}_{[(1,2),\bullet]} vr^{(2,3)}_{[1,B]}\rangle\rangle, \langle 1, \langle vi^{(1,1)}_{[1,A]} vi^{(2,1)}_{[3,C]} ei^{(2,2)}_{[(1,3),\bullet]} vr^{(2,3)}_{[1,B]}\rangle\rangle\}$ である．一方，図 *5 (b)* の g_2 が *GTRACE2* の射影の入力であるとき，g_2 の *ID* が *1* と *2* である *2* つの頂点はラベル付き和グラフ $g_u(d)$ の *ID* が *1* と *2* の頂点に対応するので，$proj_2(\langle 1, d\rangle, g_2) = \{\langle 1, \langle vi^{(1,1)}_{[1,A]} vi^{(1,2)}_{[2,B]} ei^{(1,3)}_{[(1,2),\bullet]}\rangle\rangle\}$ である．定義 *8* の $g_u(d) = g$ を満たすために，この射影された変換系列は $vr^{(2,3)}_{[1,B]}$ を含まない．

4. 評価実験

前節までに示した手法を統合した提案手法を C++で実装し，Intel Core 2 6700 2.66GHz の CPU，2GB のメインメモリを搭載した HP xw4400 を用いて評価実験を行った．

4.1 人工データ

表 2 のパラメータを用いて比較実験のための人工データを作成した．はじめに，2 頂点間の辺存在確率 p_e で外部状態 $g^{(0)}$ を生成する．続いて，確率 p_i で頂点や辺の追加，確率 p_i で頂点や辺の削除，確率 $p_d = 1 - p_i - p_d$ で頂点ラベル変更の変換規則を適用して，グラフを変換した．この変換ではグラフ系列中の連続する 2 グラフの編集距離は平均 d_e となるようにし，グラフ系列の ID 数が $|V_{avg}|$ になるまで続け，1 つのグラフ系列を生成した．このようにして $|DB|$ 個のグラフ系列を生成した．これと並行して $|V_{avg}|$ を $|V'_{avg}|$ に置き換えて N 個のグラフ系列を生成し，各系列 $d \in DB$ に N 個のグラフ系列の 1 つを上書きした．外部状態のグラフは $|L_v|$ 種の頂点ラベルを持つよう人工データを生成した．

表 2: 人工データ生成のパラメータと最小支持度のデフォルト値

パラメータ	デフォルト値
$vi^{(j,k)}_{[o,l]}, ei^{(j,k)}_{[o,l]}$ が変換規則の末尾に付加される確率	$p_i = 80\%$
$vd^{(j,k)}_{[o,l]}, ed^{(j,k)}_{[o,l]}$ が変換規則の末尾に付加される確率	$p_d = 10\%$
FTS 内の平均 ID 数	$\|V_{avg}\| = 6$
rFTS として埋め込まれる FTS 内の平均 ID 数	$\|V'_{avg}\| = 3$
頂点ラベル数	$\|L_v\| = 5$
rFTS として埋め込まれる FTS の数	$N = 10$
変換系列数	$\|DB\| = 1{,}000$
2 頂点間の辺存在確率	p_e =15%
連続する外部状態の平均編集距離	d_e =2
最小支持度	$\sigma' = 300$

表 3: $|DB|, |V_{avg}|, \sigma', |L_v|$ を変化させたときの実験結果

$\|DB\|, \|V_{avg}\|$	1000	5000	10000	8	9	10
GTRACE 計算時間	59.7	460.1	1067.7	9892.3	–	–
PS 計算時間	50.3	412.0	967.7	9792.3	–	–
頻出部分グラフ数	8	8	8	11	–	–
系列数	34721	172386	347768	179623	–	–
平均長	22.3	22.4	22.6	33.8	–	–
GTRACE2 計算時間	3.6	22.9	54.4	13.1	19.3	41.1
PS 計算時間	0.64	5.3	13.3	3.8	6.7	17.2
頻出部分グラフ数	22	21	20	34	35	35
系列数	13470	70039	148803	24653	32059	45867
平均長	5.7	5.7	5.7	7.0	7.2	7.7
$\sigma', \|L_v\|$	50	30	10	1	3	5
GTRACE 計算時間	3524.4	7664.7	–	4716.6	136.1	59.7
PS 計算時間	3508.7	7647.2	–	4691.7	124.1	50.3
頻出部分グラフ数	15	21	–	11	9	8
系列数	25482	18947	–	56643	37575	34721
平均長	25.9	26.6	–	26.0	23.4	22.3
GTRACE2 計算時間	25.7	54.1	276.8	1036	6.1	3.6
PS 計算時間	17.5	44.6	266.4	1045.0	1.4	0.64
頻出部分グラフ数	62	78	99	24	20	22
系列数	10044	8950	8027	63806	21492	13470
平均長	8.0	8.5	9.1	12.6	7.1	5.7

PS 計算時間：PrefixSpan の計算時間，頻出部分グラフ数：AcGM により G_u から列挙される頻出連結部分グラフの数, 系列数: PrefixSpan の入力となる変換系列の数の平均，系列長：PrefixSpan の入力となる変換系列の平均変換規則数

表 3 は $|DB|, |V_{avg}|, \sigma', |L_v|$ のいずれかを変化されたときの GTRACE と GTRACE2 の計算時間 [秒]，それらの内部で呼ばれる PrefixSpan の計算時間 [秒]，AcGM が列挙した頻出連結部分グラフの数，射影により出力された平均変換系列数と平均変換規則数を表わしている．変化させないパラメータはデフォルト値とした．3 時間で計算が終了しない場合を “–” で示す．

表 3 は，他の頻出パターンマイニング手法と同様に，GTRACE と GTRACE2 の計算時間の計算時間がグラフ系列 DB の増加に対して比例することを示している．さらに $|V_{avg}|$ の増加，σ' の減少，$|L_v|$ の増加に対して，GTRACE と GTRACE2 の計算時間が指数関数的に増加していることを表わしている．これは，$|V_{avg}|$ の増加などにより rFTS の数が増加するためである．しかし，GTRACE2 の計算時間は GTRACE に比べて遥かに短いことが分かる．表 3 の中で，特に興味深い結果は，頂点ラベル数が 1 の場合にも，GTRACE2 の計算時間は GTRACE の計算時間の約 4 分の 1 となったことである．頂点ラベル数が 1 の場合，グラフ系列のラベル付き和グラフは一般の和グラフと同型になるが，変換系列の和グラフは必ずしも同型になるとは限らない．例えば，変換系列 $\langle vi^{(j,1)}_{[1,l]} ei^{(j,2)}_{[(1,2),\bullet]}\rangle$ の和グラフにおいて，ID が 1 の頂点はラベル l をもち，ID が 2 の頂点はラベル l_- をもつ．従って，グラフ系列を 1 種類のラベルではなく，2 種類のラベルのグラフ系列データとして扱うことができるため，GTRACE2 の射影により出力される変換系列の数と系列中の変換規則数が減り，GTRACE2 は GTRACE に比べ効率良く rFTS を列挙することができる．

5. まとめ

本稿では，ラベル付き和グラフとそれに対応した部分グラフ同型判定を GTRACE に取り込んだ手法 GTRACE2 を提案した．これにより，GTRACE2 を系列の変換規則が多く，各状態の頂点数が多いグラフ系列にも適用することが可能となった．

参考文献

[1] Enron Dataset, `http://www.cs.cmu.edu/~enron/`

[2] A. Inokuchi and T. Washio. A Fast Method to Mine Frequent Subsequences from Graph Sequence Data. *Proc. of Int'l Conf. on Data Mining*, pp. 303–312, 2008.

[3] A. Inokuchi, et. al. A Fast Algorithm for Mining Frequent Connected Subgraphs. *IBM Research Report*, RT0448, 2002.

[4] 元田浩. 明示的理解に魅せられて. 人工知能学会学会誌 pp.615-625,1999.

[5] J. Pei, et. al. PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth, *Proc. of Int'l Conf. on Data Eng.*, pp. 2–6, 2001.