極大2部クリークの高速列挙法とデータマイニングへの応用

宇野 毅明* 有村 博紀† 浅井 達哉†

平成 15 年 7 月 15 日

あらまし: 本稿では，2 部グラフ $G = (V, E)$ の極大 2 部クリークの列挙を行うアルゴリズムの提案を行う．既存のアルゴリズムの計算量は，極大 2 部クリーク 1 つあたり $O(|V||E|)$ であるが，本研究のアルゴリズムは $O(\Delta^3)$ である．ここで，Δ は G の最大次数である．また計算機実験により，ランダムに作ったグラフに対する実験的な計算時間は $O(\Delta)$ となることを示す．さらに，このアルゴリズムを，データベースから頻出集合を発見する問題に適用し，ベンチマーク問題に対して計算機実験を行った．計算結果をから，これらの問題に対しては既存の手法よりも本研究の手法が高速であることを示す．

キーワード: 数え上げ，発生，計算量，アルゴリズム，疎グラフ，頻出集合発見，データマイニング

1 はじめに

グラフ $G = (V = V_1 \cup V_2, A)$ の任意の枝が V_1 と V_2 の頂点を結ぶ枝であるとき，G は**2 部グラフ**とよばれる．G の頂点部分集合 $H \subseteq V_1, K \subseteq V_2$ に対して，H の任意の頂点と K の任意の頂点の間に枝があるとき，H と K を合わせた頂点集合を**2 部クリーク**とよぶ．$K = \emptyset, H = V_1$ である場合，あるいはその逆である場合も 2 部クリークである．ある 2 部クリークが他の 2 部クリークに含まれないとき，その 2 部クリークを**極大 2 部クリーク**とよぶ．本稿では，入力したグラフの極大 2 部クリークを，重複することなく全て出力する問題を扱う．

グラフ G の極大 2 部クリークを列挙する問題は，V_1 の頂点間，および V_2 の頂点間に枝を張ったグラフの極大クリーク列挙問題と等価である．グラフの極大クリークを列挙問題に対しては，1977 に築山ら [8] によって，極大クリーク 1 つあたり $O(|V||E|)$ 時間のアルゴリズムが提案されている．よって，2 部グラフの極大 2 部クリークは，このアルゴリズムを用いて 1 つあたり $O(|V|^3)$ 時間で列挙できる．さらに，簡単な改良により計算時間は $O(|V||E|)$ になる．

クリークは，データベースのクラスタリング，ウェブネットワークのコミュニティー発見，データマイニング，学習理論など様々な分野の様々なモデルで用いられている．極大クリークは，グラフのクリーク全体を良く表現するため，これらモデルの研究において，極大クリークの列挙はしばしば有効な問題解決手段となる．しかし，これらのモデルで現れる問題は通常多くの頂点を持つ．そのため，築山らアルゴリズムで極大クリークを列挙したのでは，多大な時間がかかる．

これら現実の問題は疎であり，グラフの平均次数・最大次数は小さいことが多い．そこで，本研究では，極大 2 部クリーク専門の列挙アルゴリズムを開発し，疎であることを利用した高速化を行う．その結果，極大 2 部クリーク 1 つあたりの計算量は $O(\Delta^3)$ になる．ここで，Δ は G の最大次数である．さらに計算実験により，ランダムに作成したグラフでは，平均的な計算時間は $O(\Delta)$ となることを示す．

2 部クリークはデータマイニングの頻出集合発見問題に応用がある．極大 2 部クリーク列挙は，データベースから closed item set の極大元を全て見つけ出す問題と等価である事が知られている．よって，本研究で提案

*国立情報学研究所, 〒 101-8430 東京都千代田区一ツ橋 2-1-2 e-mail: uno@nii.jp
†九州大学システム情報科学研究院, 〒 812-0053 福岡県福岡市東区箱崎 6-10-1 e-mail: arim@i.kyushu-u.ac.jp, t-asai@i.kyushu-u.ac.jp

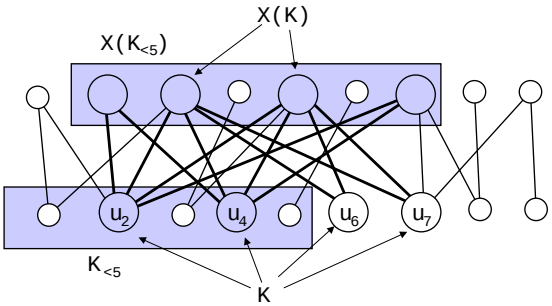


図 1: 極大 2 部クリーク K と $K_{\leq 5}, X(K), X(K_{\leq 5})$

するアルゴリズムを用いて，closed item set が列挙できる．計算機実験により，ベンチマーク問題では既存の closed item set 列挙アルゴリズムよりも本研究の手法が高速であることを示す．

2 アルゴリズム

この節では，本稿で提案する極大 2 部クリーク列挙アルゴリズムの説明をする．本稿のアルゴリズムは，Avis と福田 [3] によって提案された逆探索法をもとに，築山らのアルゴリズムを改良したものである．まず説明の前にいくつかの記法を導入する．

2 部グラフ $G = (V_1 \cup V_2, A)$ に対して，$V_1 = \{v_1, ..., v_m\}, V_2 = \{u_1, ..., u_n\}$ とする．全ての V_1 の頂点と隣接する V_2 の頂点は存在しないとする．頂点集合 $H \subseteq V_1$ に対して，$X(H)$ を H の全ての頂点に隣接する V_2 の頂点の集合とする．$K \subseteq V_2$ に対しても同様に $X(K)$ を定める．$H \subseteq V_1$ と $K \subseteq V_2$ を合わせた頂点集合が極大 2 部クリークであれば，またそのときに限り，$X(H) = K$ かつ $X(K) = H$ が成り立つ．そこで，以後極大 2 部クリークを，その V_2 に含まれる頂点のみで定める．頂点 v に対して，$X(\{v\})$ は v に隣接する頂点の集合になる．頂点集合 K に対して，$K_{\leq i}$ を添え字が i 以下の K の頂点の集合とする．

極大 2 部クリーク $K \subseteq V_2$ に対して，その親を $X(X(K_{\leq i}))$ で定める．ただし i は $X(K_{\leq i-1}) \neq X(K)$ となる最大の i とし，この i を K の**親添え字**とよぶ．任意の極大 2 部クリーク $K \neq \emptyset$ に対して，その親が唯一的に定まる．また，$K_{\leq i} \subset K$ であることから，$X(K_{\leq i-1}) \neq X(K)$ であるなら，$X(K) \subset X(K_{\leq i})$ であり，よって

$$X(X(K_{\leq i})) \subset X(X(K)) = K$$

となる．ゆえに，この親子関係は非巡回的であり，極大 2 部クリーク $\emptyset$ を根とする木構造を導出する．この木構造を**列挙木**とよぶ．列挙木を根から出発して深さ優先探索することにより，全ての極大 2 部クリークを列挙できる．

列挙木を深さ優先探索するには，与えられた極大 2 部クリークの子供を列挙するアルゴリズムを構築し，根の子供を列挙した後に，再帰呼び出しによってそれら子供の子孫を列挙すれば良い．計算を簡略化するため，極大 2 部クリーク V_2 は列挙する対象から除外する．

極大 2 部クリーク K に対して，$K[i]$ を

$$K[i] = X(X(K_{\leq i} \cup \{u_i\}))$$

で定める．極大 2 部クリーク H が K の子供であり，H の親添え字が i であるとする．このとき，親の定義から $K = X(X(H_{\leq i-1}))$ であり，$u_i \in H, u_i \notin K$ であることから，

$$\begin{aligned} X(K) \cap X(\{u_i\}) &= X(K_{\leq i-1} \cup \{u_i\}) \\ &= X(H_{\leq i}) \end{aligned}$$

が成り立つ．よって，$H = K[i]$ が成り立つ．逆に，K の親添え字よりも大きな添え字 i に対して $K[i]_{\leq i-1} = K_{\leq i-1}$ が成り立つならば，$X(K[i]) = X(K[i]_{\leq j})$ が任意の $j \geq i$ について成り立つので，$K[i]$ は K の子供である．よって，K の子供は，$K[i]$ を各 i について生成し，$K[i]_{\leq i-1} = K_{\leq i-1}$ が成り立つかどうかを判定する

ことにより，全て見つけることができる．この判定は全ての i について行う必要は無い．(1) i は K の親添え字よりも大きい，(2) u_i は $X(K)$ のどれかの頂点と隣接する，という 2 つの条件を満たすものだけで十分である．

極大 2 部クリーク K が $K = \emptyset$ でなければ，この 2 つの条件を満たす u_i は高々 Δ^2 個しか存在しない (Δ はグラフの最大次数)．各 u_i について $K[i]_{\leq i-1} = K_{\leq i-1}$ の判定には $\mathrm{O}(|X(K[i])|\Delta)$ 時間かかる．$|X(K[i])|$ は u_i に隣接する $X(K)$ の頂点の数である．よって，各 u_i について $|X(K[i])|$ の合計を取ると，それは $|X(K)|\Delta$ を超えない．これにより，判定にかかる時間の合計は $\mathrm{O}(\Delta^3)$ となる．

$K = \emptyset$ の場合，K の子供は高々 n 個である．各添え字 i に対して $K[i]$ が子供であるかどうかの判定は，$\mathrm{O}(\Delta X(\{u_i\}))$ 時間かかる．全ての i について，$|X(\{u_i\})|$ の合計を取ると，グラフの枝数と等しくなる．よって，計算時間は $\mathrm{O}(|A|\Delta)$ となる．

子供を見つけるアルゴリズムを使えば，列挙木を深さ優先探索できる．深さ優先探索は，各頂点からその子供に移動し，全ての子供の探索が終了したら親に戻る，というアルゴリズムである．このアルゴリズムで必要となる作業は，各頂点の子供を順に見つけることだけである．よって，列挙木の根，つまり極大 2 部クリーク $\emptyset$ から出発し，現在訪れている頂点の子供を列挙し，各子供に対してその子孫を再帰呼び出しで列挙することにより，列挙木の全ての頂点を探索することができる．列挙木の各頂点は極大 2 部クリークに対応するので，列挙木の深さ優先探索により，全ての極大 2 部クリークを列挙することができる．

探索中，各反復では必ず 1 つ極大 2 部クリークが出力される．先の議論により，各反復の計算時間は，列挙木の根以外では $\mathrm{O}(\Delta^3)$ である．よって，このアルゴリズムの極大 2 部クリーク 1 つあたりの計算量は，$\mathrm{O}(\Delta^3)$ となる．この他に，根の反復でのみ $\mathrm{O}(|A|\Delta)$ 時間がかかる．メモリの使用量は $\mathrm{O}(|V| + |A|)$ である．

さらに，このアルゴリズムに以下の実装上の工夫を加えた．計算量に変化はないが，無駄な計算を削減できると期待される．

(1) 頂点の添え字を，次数の小さいものから昇順でつける
(2) グラフの隣接行列の，次数が大きな頂点に対応する行をメモリに持つ
(3) 各反復で，各 $X(K[i])$ を $\mathrm{O}(|X(K)|\Delta)$ 時間で求める
(4) $K[i]_{\leq i-1}$　の頂点を添え字順で求め，逐次 $K_{\leq i-1}$ と比較することにより，$K[i]_{\leq i-1} = K_{\leq i-1}$ の判定を行う

(1) により，極大 2 部クリーク K の親添え字よりも添え字が大きく，かつ $X(K)$ に隣接するような頂点が少なくなり，各反復で子供の判定をする回数が減少する．

(2) により，次数が大きな頂点に対しては，他の頂点と隣接するかどうかの判定が定数時間でできるようになる．次数の小さな頂点に対しては，隣接行列を用いずとも，隣接の判定は短時間でできる．

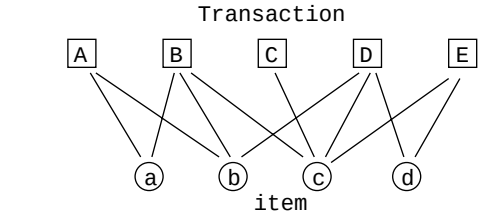
(3) 上記の通り，各 u_i について $|X(K[i])|$ の合計を取ったものは，$|X(K)|\Delta$ を超えない．これを利用して，以下のように全ての $X(K[i])$ を $\mathrm{O}(|X(K)|\Delta)$ 時間で作成する．

最初，各頂点 u_i について $S_i = \emptyset$ とする．次に，$X(K)$ の各頂点 v について，v が隣接する頂点 u_j に対して $S_i = S_i \cup \{v\}$ とする．この作業の終了後，各 u_i について，$S_i = X(K[i])$ が成り立つ．この方法により，判定が必要な $X(K[i])$ 全てを生成する時間は $\mathrm{O}(\Delta^2)$ となる．

(4) により，$K[i]_{\leq i-1} = K_{\leq i-1}$ が成り立たない場合には，早期に判定の手続きを終了することができ，結果として $K[i]$ の構築時間を短縮することができる．

3　頻出集合発見問題

E を**アイテム**の集合，$\mathcal{F}$ を E の部分集合族とし，$\mathcal{F}$ の要素を**トランザクション**とよぶ．E の部分集合 K に対して，$X(K) = \{F | K \subseteq F, F \in \mathcal{F}\}$ とする．与えられた定数 α に対して，$|X(K)| \geq \alpha$ を満たす K は**頻出集合**とよばれる．K が他の頻出集合の部分集合で無いなら，K は**極大頻出集合**とよばれる．あるトランザクションの集合 $B \subseteq \mathcal{F}$ に対して，$\{K | X(K) = B\}$ を closed item set とよぶ．任意の closed item set に対して，その極大元は 1 つしか存在しない．任意の $K, K', K \neq K', K, K' \subseteq E$ に対して，もし K と K' が同




```
Transactions
A = {a,b}
B = {a,b,c}
C = {c}
D = {b,c,d}
E = {c,d}
```

```
Maximal bipartite cliques with
 two or more transactions
{a,b,A,B}   {b,A,B,D}
{b,c,B,D}   {c,B,C,D,E}
{c,d,D,E}
```

図 2: 頻出集合問題を 2 部グラフで表現: トランザクションを 2 つ以上含む極大 2 部クリークは $\{a, b, A, B\}$, $\{b, A, B, D\}, \{b, c, B, D\}, \{c, B, C, D, E\}, \{c, d, D, E\}$ の 5 個で，それぞれの V_2 に含まれる頂点 (例えば a, b) が closed item set の極大元に対応

じ closed item set に属するならば，$X(K) = X(K')$ であり，任意の $F \in X(K)$ は K と K' を含む．よって，$X'' = K' \cup K$ も，$X(K'') = X(K)$ を満たし，その closed item set に含まれるので，K と K' の少なくとも片方は極大元でなくなるからである．

頻出集合は，データマイニングにおけるアソシエーションルール発見問題に応用があり，知識発見の分野では様々なモデルに用いられている．ただし，頻出集合の数は通常巨大であるので，極大頻出集合のみを列挙して使用するというモデルが主流であり，そのための列挙アルゴリズムの研究も盛んに行われている．しかし，極大頻出集合の列挙はさほど効率良く行えないこと (極大頻出集合 1 つあたり多項式時間のアルゴリズムはまだ提案されていない)，および極大頻出集合は頻出集合に比べて数が少ないため，頻出集合の持つ情報を完全には表現できていないと考えられることから，近年は closed item set に注目が集まっている．closed item set は,「同じトランザクション集合に含まれるアイテム集合」の集合であるので，ある意味で同じ特徴を持つアイテム集合の集合と考えられる．また，closed item set の極大元の集合は極大頻出集合の集合を含むため，closed item set は極大頻出集合の情報を完全に含む．つまり，closed item set は頻出集合と極大頻出集合の間にあるモデルであり，頻出集合に比べて，それほど情報を失っていないと考えられるからである．

近年，closed item set の極大元の中で頻出であるものを列挙する問題が研究されている．アルゴリズムも，Pei, J. Han, R. Mao による closet[7]，M. J. Zaki, C. Hsiao による CHARM[11] などが提案されている．しかし，これらは出力線形時間ではない．つまり，出力する closed item set 1 つあたりの計算時間が入力の多項式時間で押さえられていない．その一方で，closed item set の極大元と，2 部グラフの極大 2 部クリークは等価であることが知られている [4]．与えられたアイテム集合 E とトランザクションの集合 $\mathcal{F}$ に対して，以下のように 2 部グラフを構築すると，トランザクション集合 $\mathcal{F}$ に対する closed item set の極大元は，このグラフの極大 2 部クリークと 1 対 1 対応する．

・頂点集合 $V_1 = \mathcal{F}$ とする (V_1 の各頂点は各トランザクションと対応)
・頂点集合 $V_2 = E$ とする (V_2 の各頂点は各アイテムと対応)
・アイテム $e \in E$ がトランザクション F に含まれるとき，またそのときに限り $e \in V_2$ と $F \in V_1$ を枝で結ぶ

図 2 に一例を示した．このグラフの極大 2 部クリークが closed item set と対応することを以下で示そう．集合 K に対して，$X(K)$ は V_1 の頂点集合に対応する．グラフの構築方法から，K の任意の頂点と，$X(K)$ の任意の頂点の間には枝が張られている．そのため，K と $X(K)$ はグラフの 2 部クリークになる．X がある closed item set の極大元であるならば，任意のアイテム $e \notin K$ に対して，$X(K \cup \{e\}) \neq X(K)$ である．よって，あるトランザクション $F \in X(K)$ が存在して，e と $X(K)$ の間には枝が存在しない．また，$X(K)$ の定義より，任意のトランザクション $F \notin X(K)$ に対して，あるアイテム $e \in K$ が存在して，F と e の間には枝が存在しない．これは，K と $X(K)$ を合わせた頂点集合が極大 2 部クリークであることを意味する．

逆に，頂点集合 $H \subseteq V_1$ と $K \subseteq V_2$ が極大 2 部クリークであれば，$H = X(K)$ が成り立つ．さらに，$X(K) = H$ となるような V_1 の部分集合の中で，K が極大であることから，K は closed item set の極大元となる．

通常，closed item set 列挙問題の入力は疎なデータである．つまり，各トランザクションは，E の要素のごく一部しか含まないことが多い．このような疎な問題は，最大次数・平均次数の小さな 2 部グラフの極大 2 部クリーク列挙問題に変換される．よって，本研究で提案する極大 2 部クリーク列挙アルゴリズムを用いることによって，理論的にも，実用上も，高速な closed item set の極大元の列挙ができると考えられる．

極大頻出集合は closed item set の極大元であるため，closed item set を列挙し，その中で極大なものだけを出力することにより，極大頻出集合を列挙することができる．極大頻出集合の列挙が困難とされる，頻出集合数が巨大な問題に対しても，closed item set の数はそれほど大きくない場合が多く，効率の良い列挙が行えると考えられる．closeed item set の極大元が極大頻出集合であるかどうかの判定は以下のように行う．

ある closed item set の極大元 K が極大頻出集合でないならば，ある closed item set の極大元 K' が存在して，$K \subset K'$ が成り立つ．このとき，ある i に対して，

$$K \subset K[i] \subset K', \quad \text{および} \quad X(K') \subset X(K[i]) \subset X(K)$$

が成り立つ．よって，全ての $K[i]$ が頻出集合でなければ，またそのときに限り，K は極大頻出集合である．この判定は，全ての $X(K[i])$ を生成することにより行えるので，$O(\Delta^2)$ 時間でできる．

4 計算実験

本研究では，これらの工夫を加えたアルゴリズムを実装し，計算機実験を行った．使用機器は PentiumIII 500MHz と 256MB のメモリを搭載した PC，プログラム言語は C，OS は Linux である．

まず，ランダムに作成したグラフに対する実験結果を示す．実験で用いたグラフは，通常のランダムグラフとは異なる．これは，疎なランダムグラフでは極大クリークの大きさが極端に小さくなり，ほとんどの極大 2 部クリークが 1 頂点に隣接する枝の集合と等しくなってしまう．これでは実験の意味がないので，一部が密で大部分が疎であるようなグラフを作成するべく，以下の方法を用いた．V_1 の頂点の添え字は 0 から $|V_1|-1$，V_2 の頂点の添え字は $|V_1|$ から $|V_1|+|V_2|-1$ であるとする．

・各頂点 $i \in V_1$ に対して，$((i-r)+|V_2| \bmod |V_2|)+|V_1|$ から 頂点 $(i+r \bmod |V_2|)+|V_1|$ に対して，1/2 の確率で枝を張る．

r がほぼ平均次数になるので，r を調節することにより，任意の濃度のグラフを生成することができる．このグラフは局所的に密となるため，極大 2 部クリークのサイズはそれなりに大きくなり，また，頂点数が増加しても局所的な状況は変化しないため，頂点数の増加に対してほぼ線形に極大クリーク数が増加する．

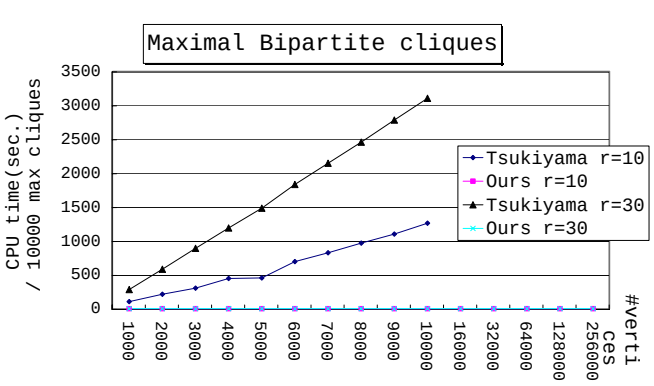
以下の表に，今回提案するアルゴリズムと，築山らのアルゴリズムの計算時間を示す．表中，築山は築山らのアルゴリズム，提案は本研究で提案するアルゴリズムを示し，計算時間は極大 2 部クリーク 10000 個あたりの秒数である．

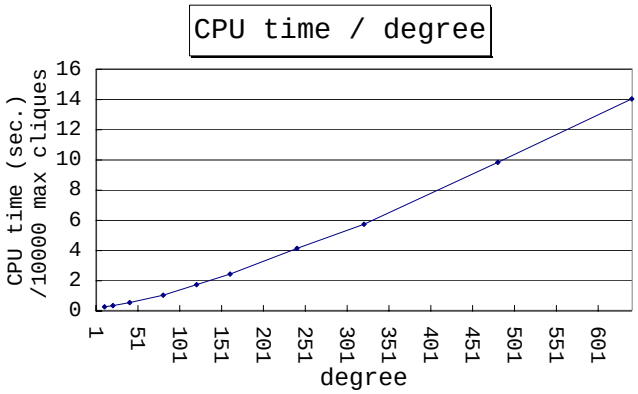
実験 1 $r = 10, 30$

頂点数	1000	2000	4000	8000	10000	16000	32000	64000	128000	256000
築山 $r=10$	104	214	446	966	1260					
築山 $r=30$	282	582	1190	2455	3100					
提案 $r=10$	0.33	0.32	0.3	0.3	0.27	0.3	0.3	0.34	0.34	0.35
提案 $r=30$	1.08	1.08	1.09	1.1	1.09	1.11	1.12	1.22	1.22	1.26
解数 $r=10$	2085	4126	8316	16609	20862	33586	67143	134911	270770	541035
解数 $r=30$	9136	18488	40427	68597	101697	165561	322149	625385	1233989	8351277

実験 2: 頂点数 1 万で平均次数を変化させた場合

r	10	20	40	80	120	160	240	320	480	640
Ours	0.23	0.31	0.51	1	1.7	2.4	4.1	5.7	9.8	14





実験 1 の結果から，築山らのアルゴリズムの 1 反復の計算時間は枝数・頂点数の線形となるのに対して，本研究のアルゴリズムは枝数・頂点数が増えても平均次数が増えなければ計算時間は増大しないことが分かる．また，実験 2 から，平均計算時間は次数にほぼ比例することが分かる．

次に頻出集合発見問題のベンチマーク問題に対する実験結果を示す．用いたベンチマーク問題は，KDD-cup2000 [6] で用いられた，商品売り上げデータ (BMS-POS)，Web ネットワークから抽出したデータ (BMS-WebView1, 2)，人工的に作られたデータ (IBM-artificial) の 4 つである．これらは，広く世界的に使用されている問題である．既存のアルゴリズムとの比較は，Z. Zheng，R. Kohavi，L. Mason [12] による比較実験の結果を用いた．この論文では以下の 4 つの頻出集合／ closed item set 列挙アルゴリズムの比較を行っている．

・R. Agrawal, H. Mannila, R. Srikant, H. Toivonen,
　A. I. Verkamo による apriori[2] (頻出集合)
・J. Han, J. Pei, Y. Yin による FP-growth[5] (頻出集合)
・J. Pei, J. Han, R. Mao による closet[7] (closed item set)
・M. J. Zaki, C. Hsiao による CHARM[11] (closed item set)

[12] の実験で用いられた機器は，CPU クロックが 550MHz の Duron と 1GB のメモリを搭載した PC であり，本研究で用いた機器より多少性能の良いマシンである．

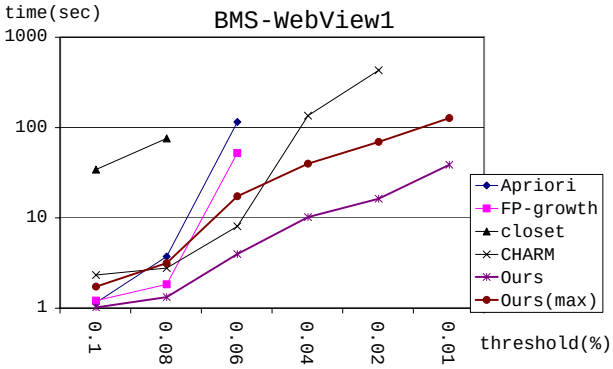
本研究の計算実験では，上記 4 つの問題に対するこれら 4 つのアルゴリズムの計算時間と，本研究で提案する closed item set の列挙アルゴリズム (Ours)，および極大頻出集合列挙アルゴリズム (Ours(max)) の計算時間を比較した．それぞれの問題で，閾値を 0.1, 0.08, 0.06, 0.04, 0.02, 0.01 に設定した場合について，計算を行った．結果は，以下の表にまとめた．表中，「-」が書かれている欄は，[12] に結果が記載されていなかった項目である．本研究で提案するアルゴリズム以外の 4 つのアルゴリズムの計算時間は，[12] の論文に掲載されているグラフから定規を当てて読み出した数値であるので，1 割程度の誤差があることを注意しておく．また，IBM-Artificial は問題生成プログラムを用いて生成したものであり，使用する乱数の違いから，同じ問題は作成できない．そのため，[12] で解かれた問題と本研究で解いた問題は異なる問題である．しかし，これらのグラフの頂点数・枝数は等しく，また同じ方法を使って作成されたことから，closed item set 数もほぼ同じであると考えられる．

また，本研究のアルゴリズムでは，閾値を用いずに全ての closed item set を列挙することができた．この結果も合わせて報告する．なお，表中の CIS は closed item set の略である．

BMS-Web-View1: アイテム数約 500，トランザクション数約 6 万，トランザクションの平均の大きさ約 2.5

閾値 (%)	0.1	0.08	0.06	0.04	0.02	0.01
Apriori	1.1	3.6	113	-	-	-
FP-growth	1.2	1.8	51	-	-	-
Closet	33	74	-	-	-	-
Charm	2.2	2.7	7.9	133	422	-
Ours	1.0	1.3	3.9	10	16	38
Ours(max)	1.7	3.1	17	39	63	125
CIS 数	3974	9391	64762	155651	422692	1240700
頻出集合数	3992	10287	461522	-	-	-
極大頻出集合数	2067	4028	15179	12956	84833	129754

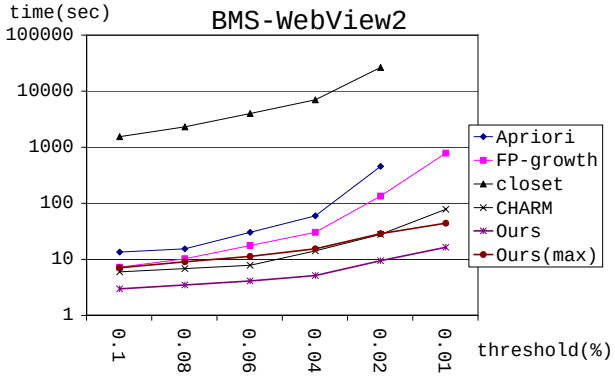
closed item set の総数：1427216 個，計算時間 46 秒



BMS-Web-View2: アイテム数約 3300，トランザクション数約 8 万，トランザクションの平均の大きさ約 5

閾値 (%)	0.1	0.08	0.06	0.04	0.02	0.01
Apriori	13.1	15	29.6	58.2	444	-
Fp-growth	7.03	10	17.2	29.6	131	763
Closet	1500	2250	3890	6840	25800	-
Charm	5.82	6.66	7.63	13.8	27.2	76
Ours	2.9	3.4	4.0	5.0	9.2	16
Ours(max)	6.9	8.8	11	15	28	43
CIS 数	22976	37099	60352	116540	343818	754924
頻出集合数	24143	42762	84334	180386	1599210	9897303
極大頻出集合数	3901	5230	7841	16298	43837	118022

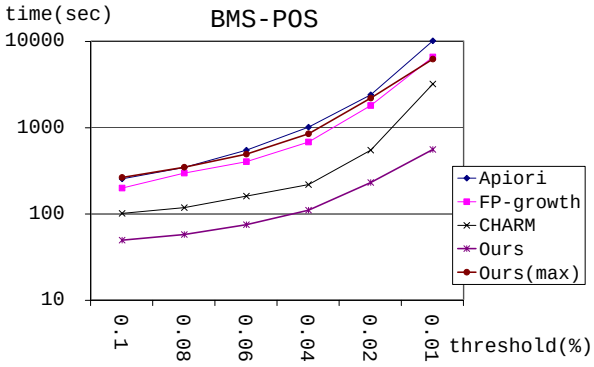
closed item set の総数: 1691051 個，計算時間 38 秒



BMS-POS: アイテム数約 1650，トランザクション数約 51 万，トランザクションの平均の大きさ約 6

閾値 (%)	0.1	0.08	0.06	0.04	0.02	0.01
Apriori	251	341	541	1000	2371	10000
Fp-growth	196	293	398	671	1778	6494
Closet	-	-	-	-	-	-
Charm	100	117	158	215	541	3162
Ours	49	57	74	109	228	551
Ours(max)	261	343	486	837	2171	6147
CIS 数	121879	200030	378217	840544	1742055	21885050
頻出集合数	121956	200595	382663	984531	5301939	33399782
極大頻出集合数	30564	48015	86175	201306	891763	4280416

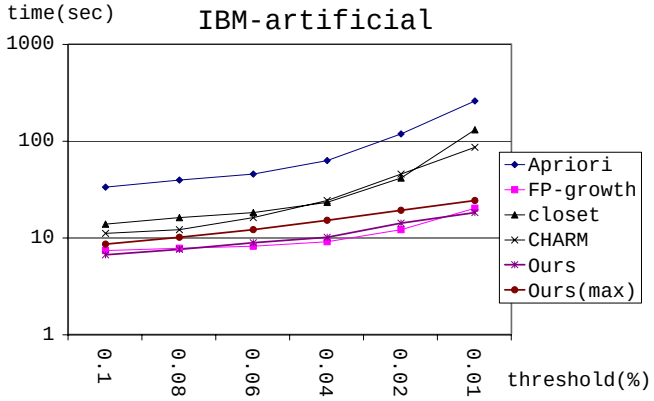
closed item set の総数：1787673748 個，計算時間 31259 秒



IBM-Artificial: アイテム数 1000，トランザクション数 10 万，トランザクションの平均の大きさ 10

閾値 (%)	0.1	0.08	0.06	0.04	0.02	0.01
Apriori	33	39	45	62	117	256
Fp-growth	7.3	7.7	8.1	9.0	12	20
Closet	13	16	18	23	41	130
Charm	11	13	16	24	45	85
Ours	6.6	7.5	8.8	10	14	18
Ours(max)	8.5	10	12	15	19	24
CIS 数	13773	22943	38436	67536	131341	229029
頻出集合数	15010	28059	46646	84669	187679	335183
極大頻出集合数	7853	11311	16848	25937	50232	114114

closed item set の総数：3160745 個，計算時間 49 秒



ほとんど全ての問題において，本研究で提案した極大 2 部クリーク列挙アルゴリズム (Ours) が短時間で問題を解いた．特に closed item set を列挙する closet，CHARM より，どの問題でも高速であり，計算時間が増大するほどその差は開いている．極大頻出集合列挙アルゴリズム (Ours(極大)) は，極大頻出集合であるかどうかの判定に多大な時間がかかっているため，極大 2 部クリーク列挙アルゴリズムより時間がかかっている．しかし，いくつかの問題では，他のアルゴリズムよりも短時間で計算が終了している．

今回実験に用いたベンチマーク問題では，closed item set の数と頻出集合の数に 2 倍以内の差しかない場合が多い．このような問題では，closed item set であるかどうかの判定を行っている分，closed item set 列挙アルゴリズムのほうが，頻出集合列挙アルゴリズムより計算時間が余計にかかると考えられる．しかし，本研究のアルゴリズムは，極大 2 部クリークの子供の生成方法を工夫するなどして 1 反復の計算時間を短縮したため，これらの問題に対しても頻出集合列挙アルゴリズムと大差ない時間で計算が終了している．BMS-WebView で閾値を 0.04% 以下に設定した問題など，closed item set 数が頻出集合数よりも 5 倍以上大きくなる問題では，明らかに極大 2 部クリーク列挙アルゴリズムが高速である．

5 まとめ

本稿では，極大 2 部クリークを列挙するアルゴリズムを提案した．アルゴリズムの計算量は，極大 2 部クリーク 1 つあたりグラフの最大次数の 3 乗のオーダーである．また，このアルゴリズムを用いて，データベースから closed item set および極大頻出集合を高速に列挙する方法を提案した．計算実験により，ランダムなグラフに対する計算時間は，極大 2 部クリーク 1 つあたりグラフの最大次数の線形オーダーであること，ベンチマーク問題では，既存の closed item set/ 極大頻出集合列挙アルゴリズムよりも高速であることを示した．

謝辞

この研究は，国立情報学研究所共同研究費の補助を受けた．

参考文献

[1] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules in Large Databases,” *In Proceedings of VLDB ’94,* pp. 487–499, 1994.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. I. Verkamo, “Fast Discovery of Association Rules,” *In Advances in Knowledge Discovery and Data Mining,* MIT Press, pp. 307–328, 1996.

[3] D. Avis and K. Fukuda, “Reverse Search for Enumeration,” *Discrete Applied Mathematics,* Vol. 65, pp. 21–46, 1996.

[4] E. Boros, V. Gurvich, L. Khachiyan, and K. Makino, “On the Complexity of Generating Maximal Frequent and Minimal Infrequent Sets,” *STACS 2002*, pp. 133-141, 2002.

[5] J. Han, J. Pei, Y. Yin, “Mining Frequent Patterns without Candidate Generation,” *SIGMOD Conference 2000*, pp. 1-12, 2000

[6] R. Kohavi, C. E. Brodley, B. Frasca, L. Mason and Z. Zheng, “KDD-Cup 2000 Organizers’ Report: Peeling the Onion,” *SIGKDD Explorations,* 2(2), pp. 86-98, 2000.

[7] J. Pei, J. Han, R. Mao, “CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets,” *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery 2000*, pp. 21-30, 2000.

[8] S. Tsukiyama, M. Ide, H. Ariyoshi and I. Shirakawa, “A New Algorithm for Generating All the Maximum Independent Sets,” *SIAM Journal on Computing,* Vol. 6, pp. 505–517, 1977.

[9] 宇野 毅明, “大規模グラフに対する高速クリーク列挙アルゴリズム,” 電気通信学会コンピュテーション研究会, 京都大学, 2003.

[10] 宇野 毅明, “大規模 2 部グラフに対する極大クリーク列挙アルゴリズムの改良と実装,” 情報処理学会アルゴリズム研究会 89, pp. 1-8, 2003.

[11] M. J. Zaki, C. Hsiao, “CHARM: An Efficient Algorithm for Closed Itemset Mining,” *2nd SIAM International Conference on Data Mining (SDM’02)*, pp. 457-473, 2002.

[12] Z. Zheng, R. Kohavi and L. Mason, “Real World Performance of Association Rule Algorithms,” *KDD 2001*, pp. 401-406, 2000.