2005－AL－100（4）
2005／3／17

核を考慮した疑似クリークの抽出

大久保 好章　　原口 誠
北海道大学大学院情報科学研究科コンピュータサイエンス専攻

概要：本稿では，疑似クリーク抽出問題について議論する．疑似クリークは，所与の閾値以上の度合で節点を共有する極大クリークを統合したものであり，共有される節点集合をその核と定めることで，統合された根拠を明確にする．疑似クリークに関する基本的な性質を明らかにし，それらに基づく枝刈り規則を利用した疑似クリーク探索について考察する．特にここでは，サイズが上位 N，すなわち，Top-N 疑似クリークを深さ優先探索により抽出するアルゴリズムを与える．予備実験により，クリーク探索によるクラスター抽出の観点から，疑似クリークを抽出することで，本質的に異なるより多くのクラスター抽出が可能となることを示す．

Extracting Pseudo-Cliques with Certain Degree of the Core

Yoshiaki OKUBO and Makoto HARAGUCHI
Division of Computer Science, Graduate School of Information Science and Technology
Hokkaido University

Abstract: In this paper, we are concerned with a problem of finding pseudo-cliques in a given graph G. A pseudo-clique is defined as the union of several maximal cliques in G with a required degree of overlap. Such a degree is determined by a user-defined parameter τ. We present a depth-first algorithm for finding pseudo-cliques whose sizes are in the top N. Based on some simple theoretical properties, effective pruning rules can be applied during our search. Our experimental result shows some advantage of considering Top-N pseudo-cliques. It is also verified that the prunings are invoked very frequently in the search.

1　はじめに

本稿では，所与の無向グラフから疑似クリークを抽出する問題について考察する．

様々な応用領域における重要なタスクが，無向グラフからの**最大クリーク**，あるいは，**極大クリーク**抽出問題として定式化できることが知られている．著者らはこれまで，固体間の類似関係をグラフ表現し，そこからサイズが上位 N の極大クリーク，すなわち，

連絡先　〒 060-0814　札幌市北区北 14 条西 9 丁目
北海道大学大学院情報科学研究科コンピュータサイエンス専攻
TEL: 011-706-7161 (FAX 兼用)
E-mail: { yoshiaki, mh }@ist.hokudai.ac.jp

Top-N 極大クリークを抽出することで，様々なクラスターを見つける枠組について考察してきた [8]．それにより，興味あるクラスターの抽出が可能であることが確認できた一方で，構成節点がわずかに異なる極大クラスターが多数抽出され，それらが Top-N のほとんどを埋め尽くしてしまう現象も度々観測された．これは，本質的に異なるクラスターの数が N に対して極僅かであることを意味し，様々なクラスターを見つける立場からは望ましくない．この様な場合，重複の度合が大きな極大クラスター同士は区別せず，ひとつのクラスターを形成していると考えることで，本質的に異なるクラスターをより多く見つけることが可能となろう．こうした背景のもと，本研究では，重複の度合が閾値以上の極大クリーク族をひとつの**疑似クリーク**と見做し，サイズが上位 N の疑似クリークを抽出する問題，すなわち，Top-N **疑似クリーク問題**を定義し，その計算アルゴリズムを与える．特にここでは，極大クリーク族の重複部分を，その疑似クリークの**核**と定め，それらが統合された根拠を明確にする．

2 準備

V を節点集合，$E \subseteq V \times V$ を枝の集合とする無向グラフ G を $G = (V, E)$ と表す．グラフ G において，節点 $v \in V$ と隣接する節点の集合を $N_G(v)$ で表し，その要素 (節点) 数，すなわち，$|N_G(v)|$ を G における v の**次数**と言う．これを $degree_G(v)$ で参照する場合もある．なお，文脈上明らかな場合は，単に $N(v)$ や $degree(v)$ と略記する．

グラフ G の任意の (異なる) 節点間に辺が存在する時，G を**完全グラフ**と呼ぶ．

グラフ $G = (V, E)$ おいて，V の部分集合を V' とする．$G' = (V', E \cap V' \times V')$ で定義されるグラフを，G の**部分グラフ**と呼び，$G(V')$ と表記する．特に，G' が完全グラフである時，それは**クリーク**と呼ばれ，単に，その構成節点集合 V' で表すものとする．また，その**サイズ**を $|V'|$ で定める．G のクリーク Q と Q' が，$Q \subseteq Q'$ の関係にある時，Q' を Q の**拡張** (extension) と呼ぶ．G のクリークのうち，包含関係のもとで極大なものを，**極大クリーク**と呼ぶ．特に，サイズが最大である極大クリークは**最大クリーク**と呼ばれる．一般に，最大クリークは一意に決まらないことに注意する．

3 核を考慮した疑似クリーク

本節では，**核を考慮した疑似クリーク**の概念を導入し，その抽出問題を定義する．

まず，所与の無向グラフ G について，その極大クリーク族の**重複度**を定義する．

定義 3.1 (極大クリーク族の重複度)
$\mathcal{C} = \{C_1, \ldots, C_m\}$ を G の極大クリーク族とする．以下で定義される $overlap(\mathcal{C})$ を $\mathcal{C}$ の**重複度**と呼ぶ．

$$overlap(\mathcal{C}) = \min_{C_i \in \mathcal{C}} \left\{ \frac{\left| \bigcap_{C_j \in \mathcal{C}} C_j \right|}{|C_i|} \right\}.$$

■

定義より，極大クリーク族の重複度 α は，$0 \leq \alpha \leq 1$ の値をとり，値が大きくなるに従い重複の度合が大きくなる．

本研究で抽出対象となる疑似クリークは，重複度を用いて次の通り定義される．

定義 3.2 (疑似クリーク)
$\mathcal{C} = \{C_1, \ldots, C_m\}$ を G の極大クリーク族とする．節点集合

$$pseudo(\mathcal{C}) = \bigcup_{C_i \in \mathcal{C}} C_i$$

を，**重複度** $overlap(\mathcal{C})$ **の疑似クリーク** (psuedo-clique) と呼び，そのサイズを $|pseudo(\mathcal{C})|$ と定める．また，各 C_i の積，すなわち，$\bigcap_{C_i \in \mathcal{C}} C_i$ を疑似クリーク $pseudo(\mathcal{C})$ の**核** (core) と呼ぶ． ■

重複が少ない極大クリーク同士を統合して疑似クリークと見做しても，まとまりとしての説得力には欠けるであろう．その意味で，疑似クリークの重複度は，それを構成する極大クリークをひとつに見做すに至った根拠を表すと考えられる．よってここでは，重複度の閾値を設け，閾値以上の重複度を有する疑似クリークのみを抽出の対象とする．特に，サイズが上位 N の疑似クリークを求めることで，根拠が明白な様々なまとまりを抽出することを考える．

定義 3.3 (Top-N 疑似クリーク問題)
G を無向グラフ，τ を重複度閾値とする．G の極大クリークから構成される重複度 τ 以上の疑似クリーク (これを τ-疑似クリークと呼ぶ) のうち，サイズが上位 N であるものを求める問題を Top-N **疑似クリーク問題**と定める． ■

次節では，Top-N 疑似クリークの抽出アルゴリズムについて議論する．

4 Top-N 疑似クリーク抽出アルゴリズム

無向グラフ $G = (V, E)$ と重複度閾値 τ が与えられた時，G の Top-N 疑似クリークを求める計算手続きについて議論する．その詳細を述べる前に，疑似クリークの基本的な性質を明らかにしておく．

まず，クリークの拡張候補節点を定める．

定義 4.1 (クリークの拡張候補節点)
Q を G のクリークとする．Q の任意の節点に隣接する $v \in V$ を，Q の**拡張候補節点**と呼ぶ．Q のすべての拡張候補節点の集合を $cand(Q)$ と表記する． ■

定義より，任意の $v \in cand(Q)$ について，$Q \cup \{v\}$ がクリークを形成することは明らかである．また，以下が成立することも容易にわかる．

観察 4.1
Q と Q' を $Q \subseteq Q'$ なる G のクリークとする．この時，$cand(Q) \supseteq cand(Q')$ かつ $|Q| + |cand(Q)| \geq |Q'| + |cand(Q')|$ である． ■

G のクリーク Q について，$Q \subseteq C_{max}$ なる G の任意の極大クリーク C_{max} は，$cand(Q)$ 中のいくつかの節点で Q を拡張することにより得られる．ここで，疑似クリークの核はクリークであることに注意すると，Q を核とする疑似クリークのサイズは高々 $|Q|+|cand(Q)|$ であることがわかる．このことから，以下の性質が明らかとなる．

観察 4.2
Q を G のクリークとする．今，Top-N の疑似クリークが暫定的に見つかっているとし，その最小サイズを k と仮定する．$|Q| + |cand(Q)| < k$ である時，Q の任意の拡張 Q' を核とする疑似クリークは Top-N に成り得えない． ■

τ を重複度閾値とする．今，τ-疑似クリーク $\tilde{C}$ の核を Q とすると，$\tilde{C}$ は，$Q \subseteq C$ かつ $|Q|/|C| \geq \tau$ なる任意の極大クリーク C の和集合として得られる．この様な C について，$Q \cup D = C$ となる，部分グラフ $G(cand(Q))$ の極大クリーク D が存在することに注意しよう．つまり，疑似クリーク $\tilde{C}$ を得るためには，$|Q|/(|Q| + |D|) \geq \tau$ なる $G(cand(Q))$ の任意の極大クリーク D を見つければ十分である．

こうした極大クリークの抽出は，一般に計算コストの高いタスクであるが，ここでは以下の理由により，計算負荷がある程度抑えられるものと期待できる．

- 疑似クリークの核が小さい場合，それを構成する極大クリークのサイズも，重複度閾値に制約されて必然的に小さくなる．サイズの小さな極大クリークは，比較的容易に見つけられることから，その計算コストは現実的な範囲に収まるものと期待できる．

- 一方，疑似クリークの核 Q が大きくなるにつれて，$cand(Q)$ のサイズは単調に減少し，それに伴い部分グラフ $G(cand(Q)$ の節点数も少なくなる．小さなグラフから極大クリークを抽出する負荷はそれほど大きくないことが期待できるため，この場合も現実的なコストで計算が可能となろう．

前者は，具体的には次の性質によって支持される．

観察 4.3
G のクリーク Q について，Q を核とする τ-疑似クリーク $\tilde{C}$ の抽出を考える．部分グラフ $G(cand(Q)$ のクリーク D について，$|D| > (\frac{1}{\tau} - 1) \cdot |Q|$ であるならば，$\tilde{C}$ の抽出過程において D の任意の拡張を考慮する必要はない． ■

核を Q とする疑似クリーク $\tilde{C}$ の抽出にあたっては，一般に，$G(cand(Q))$ の 極大クリークを計算する必要があることを上に述べたが，次の場合には，その計算をせずに $\tilde{C}$ を同定することができる．

観察 4.4
G のクリークを Q，重複度閾値を τ とする．以下が成り立つ時，$Q \cup cand(Q)$ は Q を核とする疑似クリークとなる．

- $(\frac{1}{\tau} - 1) \cdot |Q| \geq k$，ここで k は $G(cand(Q))$ における極大クリークの上限値である．

- 任意の $v \in cand(Q)$ について，$G(cand(Q))$ における v の次数は $|cand(Q)| - 1$ よりも小さい． ■

前者により，$Q \cup cand(Q)$ が τ-疑似クリークとなることが保証されるが，一般には，その核は Q の (真の) 拡張となる．核が Q であることは，後者により保証される．

極大クリークの上限値は，多くの最大クリーク抽出アルゴリズムにおいて利用されている [2, 3, 4, 5, 6]．特に，文献 [3] により，グラフの**染色数**がタイトな上限値を与えることが示されているが，それを正確に求めるのは極めて困難である．そのため多くのアルゴリズムでは，逐次彩色を行なって得た近似値を用いることが多い．

以上の議論から，無向グラフ G における Top-N τ-疑似クリークを深さ優先探索によって抽出するものとする．その基本戦略は次の通りである．G のクリーク Q について，$G(cand(Q))$ の極大クリークを抽出することで，Q を核とする τ-疑似クリーク $\tilde{C}$ を求める．もし，$\tilde{C}$ のサイズが，既に暫定的に見つかっている Top-N 疑似クリークの最小サイズよりも大きい場合は，$\tilde{C}$ を Top-N 暫定リストに登録した後，Q の最小の真の拡張 Q' について同様の処理を行なうことで，Q' を核とする τ-疑似クリークの抽出を試みる．もし $|Q| + |cand(Q)|$ が暫定 Top-N における最小サイズよりも小さい場合は，観察 4.2 より，Q の任意の拡張を調べる必要はないことから，Q の拡張処理を直ちに枝刈る．初期の核 Q を空集合に設定し，考慮すべき核がなくなるまでこうした処理を深さ優先で繰り返し，最終的に得られた Top-N (暫定) リストを解として出力する．

以上をまとめたアルゴリズムを図 1 に示す．

5 予備実験

提案したアルゴリズムを C 言語で実装し，予備実験を行なった．その主目的は，クラスター抽出の観点から，Top-N の疑似クリークを抽出することで，様々なバリエーションのクラスターが獲得可能なことを示すことにある．

疑似クリークの抽出対象となるグラフは，2340 の節点と 9391 の辺で構成されている*．いくつかの重複度閾値のもとで，Top-20 の疑似クリーク抽出を試みた．なお，厳密な極大クリークを Top-20 抽出した場合，それらはふたつに大別される．ひとつは，8 節点を共有する極大クリーク族であり，それらの中で最大な極大クリークサイズは 11 であった．もうひとつは，これとはまったく異なる 8 節点を共有する極大クリーク族であり，これらの中で最大サイズは 12 であった．すなわち，上位 20 の極大クリークを抽出したが，本質的に異なるものは 2 種類であったと考えられる．

重複度閾値 $\tau = 0.8$ のもとで Top-20 の疑似クリークを抽出した場合，大別して 5 種類のクラスターを得ることができた．特に，上位 9 番目の疑似クリークのサイズは 12 であるにも拘らず，その核のサイズは 4，構成極大クリークの最大サイズは 5 であった．これは比較的小さな極大クリークが，ある程度数統合されたことを意味している．厳密な Top-N 極大クリーク抽出によって，この様な小さな極大クリークを得るためには N を十分大きく設定すればよいが，その場合，ユーザは得られた多数の極大クリークの中からそ

*これは，ある生物の遺伝子発現時系列データを，その発現パターンの類似性に基づいてグラフ表現したものである．

```
procedure main() :
  V ← the set of vertices in a graph ;
  E ← the set of edges in the graph ;
  N ← an integer for Top-N ;
  τ ← a threshold for overlap degree ;
  PC ← φ;
  size_num ← 0 ;
  min_size ← 0 ;
  FindPseudoCliques(φ, V) ;
  return PC ;
```

```
procedure FindPseudoCliques(Q, R) :
  if size_num = N and |Q| + |R| < min_size then
    return ;   /* 観察 4.2 に基づく枝刈り */
  endif
  for each v ∈ R in predetermined order
    begin
      MC ← φ ;
      α ← (1/τ − 1) · (|Q| + 1) ;
      k ← an upper bound of the maximum clique in R ∩ N(v) ;
      if k ≤ α then
        if ∀w ∈ R ∩ N(v), degree_{G(R∩N(v))}(w) < |R ∩ N(v)| − 1 then
          MC ← {R ∩ N(v), φ} ;   /* 観察 4.4 に基づく枝刈り */
        else
          FindMaxCliques(φ, R ∩ N(v)) ;
        endif
      else
        FindMaxCliques(φ, R ∩ N(v)) ;
      endif
      if ⋂_{C_i∈MC} C_i = φ then
        if size_num < N or |⋃_{C_i∈MC} C_i ∪ Q ∪ {v}| ≥ min_size then
          PC ← PC ∪ {⋃_{C_i∈MC} C_i ∪ Q ∪ {v}} ;
          size_num ← |{|PC| | PC ∈ PC}| ;
          min_size ← min{|PC| | PC ∈ PC} ;
        endif
      endif
      FindPseudoCliques(Q ∪ {v}, R ∩ N(v)) ;
    end
```

```
procedure FindMaxCliques(Q, R) :
  if |Q| > α then
    return ; /* 観察 4.3 に基づく枝刈り */
  endif
  if R = φ then
    MC ← MC ∪ {Q} ;
    return ;
  endif
  for each v ∈ R in predetermined order
    FindMaxCliques(Q ∪ {v}, R ∩ N(v)) ;
```

図 1: Top-N τ-疑似クリーク抽出アルゴリズム

れを探し出すことを強いられる．しかし，疑似クリークを考えることで，そうした手間なしに，その存在を容易に認識することが可能となる．これは，疑似クリークを考えることによる極めて重要かつ有用な効果であることを強調しておく．

本予備実験により，探索の枝刈りも有効に機能していることが確認できた．$\tau = 0.8$ の設定において，実際に訪れた探索ノード数は 22235 であったが，その約半分の 10161 のノードにおいて枝刈りが適用された．このことから，本稿での枝刈り規則は，探索過程において十分な頻度で適用されるものと考える．

6 おわりに

本稿では，疑似クリークの概念を導入し，深さ優先探索により Top-N 疑似クリークを抽出するアルゴリズムについて議論した．疑似クリークは，閾値以上の重複度を有する極大クリークの和集合で定義され，本質的には同じと見做せる極大クリークをひとつに統合したものに相当する．特にここでは，重複部分を疑似クリークの核と定めることで，構成する極大クリークを統合するに至った根拠を明確化した．提案したアルゴリズムは，疑似クリークに関する基本的性質に基づく枝刈り規則を利用して疑似クリークを抽出する．予備実験により，探索中に本枝刈り規則が十分な頻度で適用されることを確認した．また，クラスター抽出の観点からも，疑似クリークを考えることで，これまで見つけることが困難であった様々なバリエーションのクラスター抽出が可能となることを確かめた．

今後は，より大規模なグラフを扱える様，アルゴリズムのさらなる改良を行ないたい．現在のアルゴリズムでは，核 Q に対して $G(cand(Q))$ の極大クリークを抽出する際，単純な深さ優先探索を用いている．大規模グラフにおいては，その計算負荷の増加が見込まれることから，極大クリークの高速列挙アルゴリズム [7] 等を利用した効率化が不可欠となろう．また，完全性を多少犠牲にした上で，経験則に基づく枝刈り規則を探索に導入することも現実的な改良策のひとつと考えられる．今後はこうした点を中心に考察を進めたい．

謝辞

本研究に関して大変有益な議論をして頂いた，北海道大学創成科学研究機構・安住薫助手の研究グループに感謝の意を表します．なお，本研究は一部，国立情報学研究所・共同研究『説得力をもった推論を可能ならしめるメタファー・類推』の補助を受けている．

参考文献

[1] I. M. Bomze, M. Budinish, P. M. Pardalos and M. Pelillo, “The Maximum Clique Problem”, Handbook of Combinatorial Optimization, Kluwer Academic Publishers, Supplement Vol. A, pp. 1 - 74, 1999.

[2] E. Tomita and T. Seki, “An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique”, Proceedings of the 4th International Conference on Discrete

Mathematics and Theoretical Computer Science - DMTCS'03, Springer-LNCS 2731, pp. 278 - 289, 2003.

[3] T. Fahle, "Simple and Fast: Improving a Branch-and-Bound Algorithm for Maximum Clique", Proceedings of the 10th European Symposium on Algorithms - ESA'02, Springer-LNCS 2461, pp. 485 - 498, 2002.

[4] D. R. Wood, "An Algorithm for Finding a Maximum Clique in a Graph", Operations Research Letters, vol. 21, pp. 211 - 217, 1997.

[5] P. R. J. Östergård, "A Fast Algorithm for the Maximum Clique Problem", Discrete Applied Mathematics, vol. 120, pp. 197 - 207, 2002.

[6] R. Carraghan and P. M. Pardalos, "An Exact Algorithm for the Maximum Clique Problem", Operations Research Letters, vol. 9, pp. 375 - 382, 1990.

[7] T. Uno, "Fast Algorithms for Enumerating Cliques in Huge Graphs", Technical Report of IEICE, Vol. 103, No. 31 (COMP2003 1-8), pp. 55 - 62, 2003. (in Japanese)

[8] Y. Okubo and M. Haraguchi, "Creating Abstract Concepts for Classification by Finding Top-N Maximal Weighted Cliques", Proceedings of the 6th International Conference on Discovery Science - DS'03, Springer-LNAI 2843, pp. 418 - 425, 2003.