1C3-3

Enumerating Maximal Clique Sets with K-plex Constraint by using Meta-cliques

Hongjie Zhai*1 Makoto Haraguchi*1 Yoshiaki Okubo*1 Etsuji Tomita*2

*1 Graduate School of Information Science and Technology, Hokkaido University
*2 The Advanced Algorithms Resaerch Laboratory, UEC Tokyo

Many variants of pseudo-cliques have been introduced as relaxation models of cliques to detect communities in real world networks. For most types of pseudo-cliques, enumeration algorithms can be designed just similar to maximal clique enumerator. However, the problem of enumerating pseudo-cliques are computationally hard, because the number of maximal pseudo-cliques is huge in general. Furthermore, because of the weak requirement of k-plex, sparse communities are also allowed depending on the parameter k. To obtain a class of more dense pseudo cliques and to improve the computational performance, we introdue a derived graph whose vertices are cliques in the original input graph. Then our target must be cliques or pseudo cliques of the derived graph under an additional constraint requiring high density in the original graph. An enumerator for this new class is designed and its computational efficientcy is experimentally verfied.

1. Introduction

For graph theory, cliques are subgraphs in which all vertices connect to each other. It is usually used to detect densely connected communities. However, for real-world datasets, there exist many noises to make dense communities not to be cliques. The strict definition of clique is not suited for real applications. To address this issue, many relaxation models of clique, also called pseudo-cliques, are introduced [3]. Every pseudo-clique model weaken some requirements of clique. k-clique and k-clan are defined by weakening reachability. k-core, k-plex are defined by relaxing the requirement of closeness. In cases of k-cores and k-plexes, some vertices may not be connected. In this paper, we will mainly discuss k-plex.

K-plex was introduced by Seidman [1]. K-plex is a subgraph in which each member is connected to at least $n-k$ other members. When $k=1$, k-plex becomes clique. For k-plex model with small k values as ($k=2,3$), k-plexes can detect the densely connected subgraphs. However, when k grows larger, sparse graph such as chain or circle will become k-plexes. The number of k-plexes also grows exponentially. Even though clique enumeration can be done efficiently, the task of finding all the maximal k-plexes for larger k is actually impractical, because k-plex allows much more combinations of vertices than clique.

In this paper, our target is to obtain densely connected k-plexes with cliques as their components, inluding overlapping cliques in paper [8]. We consider the k-plexes are only combined from maximal cliques whose size is larger than a given lower bound. In other words, the target of this paper is to detect the maximal clique sets under the k-plex constraint. By introducing the concept of k-Maximal-Clique-Set(k-MCS) and k-clique-graph, maximal clique sets can be found efficiently with a simple algorithm. Moreover, to exclude the k-plexes which have sparsely connected core, we also introduce bond measure as an extra constraint into our algorithm. The bond constraint can cut off most combination of sparsely connected k-plexes with small "cores". The rest of this paper is organized as follows. Basic definitions, notations are presented in Section 2.1. Section 2.2 presents the basic idea of k-MCS and k-clique-graph. Our algorithms are given in Section 2.3 and an additional constraint called bond measure is introduced in Section 2.4. We also show experiments of our algorithms in Section 3. Finally the paper is concluded with a summary and direction for future works.

Contact: Zhai Hongjie
Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido, 060-0814
Graduate School of Information Science and Technology, Hokkaido University
E-mail: zhaihj@kb.ist.hokudai.ac.jp

2. Proposed Method

2.1 Notations

Let $G=(V,E)$ be a simple undirected graph. Vertices $V=\{v_1,v_2,\ldots,v_n\}$ and edges $E=\{e_1,e_2,\ldots,e_m\}$. $|G|$ denotes the number of vertices in G. A subset of vertices $c\subseteq V$ is a clique if all the pairs of vertices are connected to each other. A clique is maximal if it is not a proper subset of another cliques. $C(G)=\{c_1,c_2,\ldots,c_j\}$ denotes all the maximal cilques of G, and all its members are sorted by size so that $|c_1|\geq|c_2|\geq\cdots\geq|c_j|$. The problem of enumerating all maximal cliques is "Maximal Clique Problem" and the base algorithm is called BK which was first introduced in 1973 [2].

Definition 1 *A subset of vertices $S\subseteq V$ is a k-plex if $deg_{G[S]}(v)\geq|S|-k$ for every vertex v in S.*

A k-plex is called maximal if it is not a proper subgraph of other k-plex. Because of the weak requirement of k-plexes, target vertices are not guranteed to be connected. k-plex is anti-monotonic, it means that if vertices set X is k-plex, for any subset $Y\subset X$, $\forall x\in X-Y$, $\{x\}\cup Y$ is also a k-plex.

Definition 2 *A subset of cliques $T \subseteq C(G)$ is a k-MCS if for $T = \{c_a, c_b, \ldots, c_u\}$ $(a \leq j, b \leq j, \ldots, u \leq j)$, vertices set $c_a \cup c_b \cup \cdots \cup c_u$ is a k-plex in G.*

A k-MCS is maximal if it is not a proper subgraph of any other k-MCS. But it is clear that maximal k-MCSes are not always maximal k-plexes in graph G. Different maximal k-MCS may represent the same vertices set in G. Eventhough deleting duplicated point can be easily done as post-process, here we still treat them as different k-MCS. According to the definition, k-MCS is also anti-monotonic. From now on, we will say a clique set $C = \{c_a, c_b, \ldots, c_u\}$ is a k-plex in G by meaning $c_a \cup c_b \cup \cdots \cup c_u$ is a k-plex in G.

2.2 Basic Idea

For a given graph G, integer k and size parameter L, we enumerate all the maximal cliques c_i in G with constraint $|c_i| \geq L$ and search for all the maximal k-MCSes based on this clique set. If $L = 1$, we are using all the maximal cliques. Size constraints on cliques is extremely useful for the graphs that contain large amount of triangles or solitude vertices. We should also point out that enumerating all maximal k-MCSes equals the process that searching for all maximal k-plexes on a graph in which all the vertices are cliques. To give a clear state, we have the following definition:

Definition 3 *A graph $C_G = (C_V, C_E)$ is called k-clique-graph if every vertex $c_i \in C_V$ is a clique in graph G and an edge exists between two vertices c_i and c_j if and only if $c_i \cup c_j$ is a k-MCS.*

According to Definition 3, only the pairs that can be k-plexes are connected to each other. From anti-monotonicity of k-MCS, we know that for a vertices set $C \subseteq C_V$, C is a clique on k-clique-graph C_G if C is a k-plex in G. When searching for maximal k-MCSes, this property allows us to limit candidates in a small range, which is all the maximal clique on k-clique-graph. We call the cliques on k-clique-graph "meta-clique" because they are the cliques of cliques for graph G. We have to point out that cliques on k-clique-graph are not always k-MCSes when the size of meta-cliques is larger than 2. It is necessary to validate all the meta-cliques with more than 2 members if they are k-MCSes or not.

2.3 Algorithms

Algorithm 1 gives a general steps for enumerating all maximal k-MCSes. In this algorithm, Build-K-Clique-Graph (Algorithm 2) will build a k-clique-graph from normal graph. Then it is able to list all meta-cliques by Enumerate-Maximal-Clique-K-plexes (Algorithm 3). In Algorithm 2, we first enumerate all the maximal cliques, then validate all the cliques pair-wisely if they can be a k-clique-graph or not. In Algorithm 3, we search for all maximal "meta-cliques" on k-clique-graph which is built in previous step. The algorithm is designed based on CLIQUES [6], an efficient algorithm for enumerating maximal cliques. A clique set X will be expanded into a larger clique set $X_x = X \cup \{x\}$ under k-plex constraints by adding a vertex $x \in Cand(X)$, where $Cand(X) = \{v \in \cap N(X) | X \cup \{v\}$ is a k-plex in $G\}$. Starting from the empty k-plex $X = \emptyset$ and $Cand(X) = C_v$, we recursively iterate this process until there is no clique set can be expanded.

In all these algorithms, FindMaximalCliqeus can be any algorithm to enumerate all maximal cliques on undirected graph. Here we use the BronKerbosch algorithm which was introduced in [2].

Algorithm 1: Maximal-k-MCS

Input: An undirected graph $G = (V, E)$
Output: All maximal k-MCSes
C_G = BuildKCliqueGraph (G);
result = EnumerateMaximalCliqueKplexes (C_G);
return *result*;

Algorithm 2: Build-K-Clique-Graph

Input: Graph G, Integer K, L
Output: k-clique-graph
Initialize $C_V = \emptyset, C_E = \emptyset$;
cand = FindMaximalCliques (G);
while *cand* $\neq \emptyset$ **do**
  curCand = a clique in cand;
  cand = *cand* − *curCand*;
  if $|C_V| < L$ **then**
    continue;
  end
  for *all cliques* c_i *in* C_V **do**
    if $c_i \cup curCand$ *is k-plex* **then**
      $C_E = C_E + e(i, \|C_V\|)$;
    end
  end
  $C_V = C_V$ + curCand;
end
return $G(C_V, C_E)$;

2.4 Bond Constraint

In many cases, our target is to find dense parts of graphs. However, as we have already discussed, k-plex does not have constraint on connectivity. In the algorithm 2 (Build-K-Clique-Graph), clique sets can be k-MCSes regardless of whether they are connected or not. For this reason, we use bond [5], an extended Jaccard coefficient, to calculate the degree of overlappingness of two vertices sets. The definition of Bond is shown in Definition 4. By only considering the clique pairs whose bonds are above certain degree, it allows us to focus on the dense connected parts of graphs. Algorithm 4 shows the algorithm for building k-clique-graph with bond measure.

Definition 4 *For two vertices set A and B, the Bond Measure is defined as* $Bond(A, B) = \frac{|A \cap B|}{|A \cup B|}$

For dense graphs, there usually exist large amount of maximal cliques. In Algorithm 2, all cliques are checked

Algorithm 3: Meta-Clique(C_G, R, P, X)

Input: k-clique-graph C_G, compsub R, candidate set P, not set X
Output: Set of maximal k-MCSes
if *P and X are both empty* **then**
  output R;
  return;
end
choose the pivot vertex u in $P \cup X$ as the vertex with the highest number of neighbors in P;
$cand = P\ N(u)$;
while $cand \neq \emptyset$ **do**
  curCand = smallest clique in $cand$;
  $tmpP = P \cap N(v)$;
  $tmpX = X \cap N(X)$;
  newP = all clique in $tmpP$ that can form k-plex with R;
  newX = all clique in $tmpX$ that can form k-plex with R;
  Meta-Clique(C_G, $newP$, $newX$);
  $cand = cand - curCand$;
  $P = P - curCand$;
  $X = X + curCand$;
end
return;

Algorithm 4: Build-K-Clique-Graph-With-Bond

Input: Graph G, Integer K, L, Float B
Output: k-clique-graph
Initialize $C_V = \emptyset, C_E = \emptyset$;
cand = FindMaximalCliques (G);
while $cand \neq \emptyset$ **do**
  curCand = a clique in cand;
  $cand = cand - curCand$;
  if $|C_V| < L$ **then**
    continue;
  end
  for *all cliques c_i in C_V* **do**
    if $Bond(c_i, curCand) > B$ **then**
      if *$c_i \cup curCand$ is k-plex* **then**
        $C_E = C_E + e(i, \|C_V\|)$;
      end
    end
  end
  $C_V = C_V$ + curCand;
end
return $G(C_V, C_E)$;

Name	# of N.	# of E.	Density
GEOM-0	7343	11898	0.00044
CA-GrQc	5242	14484	0.00105

Table 1: Statistics of datasets

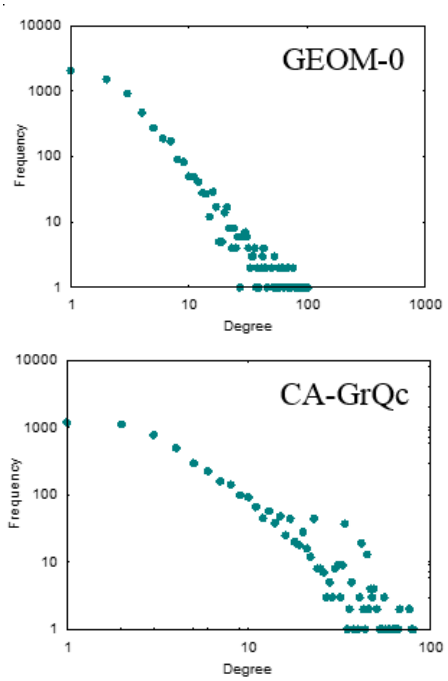

Figure 1: Degree Distribution

pair-wisely if they are k-MCSes or not. This will take a long time. By introducing bond measure, we are able to cut most useless candidates to improve the performance of enumerating algorithms.

3. Experiment

In order to evaluate the efficiency and scalability of our algorithm, we perform experiments on several benchmark graph datasets. The basic statistics of graph datasets are shown in Table 3.. Test instances are mainly real-world social networks. ca-GrQc is provided by Stanford Large Network Dataset Collection [7]. ca-GrQc is a collaboration network of Arxiv General Relativity. Geom0 is a part of Pajek [4]. This dataset is a collaboration network in computational geometry. The degree distribution of these dataset is plotted in Figure. 3.

The whole program is impremented in C with libGC and compiled with Clang. Our experiments are performed on Linux with single core 2.4GHz CPU and 8 Gbytes memory. The program is terminated if it runs for more than 180 minutes (three hours). As a comparison, we also performance the same experiments with the stardard k-plex enumerator, which is referred as MS-MaxKPlex [9]. The standard enumerator contains the following constraints on target maximal k-plexes:

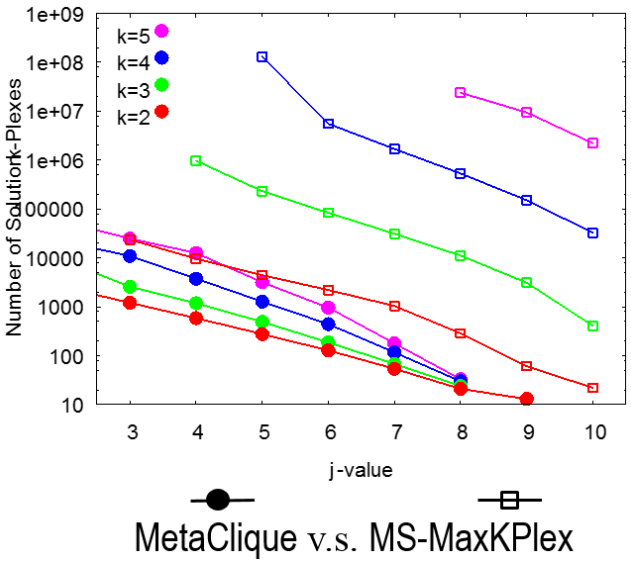

Figure 2: Dataset: GEOM0

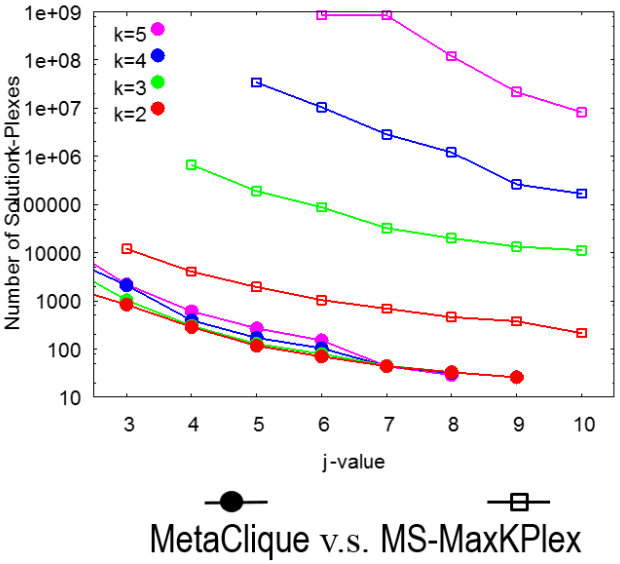

Figure 3: Dataset: ca-GrQc

- k-plex is connected.
- k-plex is not a path.
- k-plex contains at least $k+1$ vertices.

Furthermore, to compare with Meta-Clique, we added a minimum size constraint for MS-MaxKPlex. The constraint is that for given input integer L, only the k-plexes contain more than L vertices will be outputed. Figure 2 and Figure 3 shows the comparison of the number of k-plexes enumerated by meta-clique and MS-MaxKPlex. Here j represents the size constraint of cliques in meta-clique and size of k-plex in MS-MaxKPlex. Bond constraint of meta-cliques is set to 0. This means only connected k-MCSes will be enumerated. We can see that comparing with maximal k-plex enumerating, the meta-clique algorithm can restrict target into a small part of dense subgraphs. We also find that the number of meta-cliques is not so affected by the change of k. This means that the meta-clique may be able to be used as as index to characterize graphs.

4. Conclusion and Future Work

By restricting target on the combination of cliques, k-plexes can be found efficiently for large k. We introduce the concept of k-MCS and k-clique-graph, and propose an efficient maximal k-MCS enumeration algorithm. Proposed algorithm is shown to be powerful even for large-scale dataset. We also introduce constraints on size of clique and bond measure to analyze large network. The future work mainly focuses on improving the performance of k-MCS enumerator. We are also interested in combination pseudo-clique models other than k-plex with k-MCS framework.

References

[1] Stephen B. Seidman and Brian L. Foster. A graph - theoretic generalization of the clique concept. *Journal of Mathematical sociology* 6.1 (1978): 139-154.

[2] Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM* 16.9 (1973): 575-577.

[3] Jeffrey Pattillo, Nataly Youssef, and Sergiy Butenko. Clique relaxations in social network analysis. In M. T. Thai and P. M. Pardalos (Eds.), *Handbook of Optimization in Complex Networks: Communication and Social Networks*, pages 143-162. Spinger Science + Business Media, 2012.

[4] Vladimir Batagelj and Andrej Mrvar. Pajek datasets. *http://vlado.fmf.uni-lj.si/pub/networks/data/* (2006).

[5] Edward R. Omiecinski Alternative interest measures for mining associations in databases. *Knowledge and Data Engineering*, IEEE Transactions on 15.1 (2003): 57-69.

[6] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science* 363.1 (2006): 28-42.

[7] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data, jun, 2014.

[8] Makoto Haraguchi and Yoshiaki Okubo. A method for pinpoint clustering of web pages with pseudo-clique search. *Federation over the Web*. Springer Berlin Heidelberg, 2006. 59-78.

[9] Yoshiaki Okubo, Masanobu Matsudaira, and Makoto Haraguchi. Detecting Maximum k-Plex with Iterative Proper l-Plex Search. *Discovery Science: 17th International Conference, DS 2014*, Bled, Slovenia, October 8-10, 2014.