Finding Top-k Maximal Cliques in an Uncertain Graph

Zhaonian Zou, Jianzhong Li, Hong Gao, Shuo Zhang

Department of Computer Science and Technology, Harbin Institute of Technology, China
{znzou, lijzh, honggao, zhangshuocn}@hit.edu.cn

***Abstract*— Existing studies on graph mining focus on exact graphs that are precise and complete. However, graph data tends to be uncertain in practice due to noise, incompleteness and inaccuracy. This paper investigates the problem of finding top-k maximal cliques in an uncertain graph. A new model of uncertain graphs is presented, and an intuitive measure is introduced to evaluate the significance of vertex sets. An optimized branch-and-bound algorithm is developed to find top-k maximal cliques, which adopts efficient pruning rules, a new searching strategy and effective preprocessing methods. The extensive experimental results show that the proposed algorithm is very efficient on real uncertain graphs, and the top-k maximal cliques are very useful for real applications, e.g. protein complex prediction.**

I. Introduction

Graph mining has played an important role in numerous applications. Almost all existing studies on graph mining are only concerned with *exact graphs* that are precise and complete. However, graph data is generally subject to uncertainties in practice. We call such kind of graphs *uncertain graphs* [1].

Example 1: In bioinformatics, interactions between proteins are typically represented as a graph, called *protein-protein interaction (PPI) network*, where vertices represent proteins, and edges represent interactions between proteins. It has been noted that all existing PPI detection methods produce a significant amount of noisy interactions that don't really exist and miss a fraction of real interactions [2]. Thus, it is more realistic to represent a PPI network as an uncertain graph with the uncertainty of each edge representing the chance of the interaction existing in practice. The STRING database (http://string-db.org) is such a public data source that provides PPIs as well as their uncertainties explicitly.

By extending the model in our prior work [1], an uncertain graph is modeled as a weighted graph, where the weight of each vertex represents the probability that the vertex actually exists, and the weight of each edge represents the conditional probability that the edge really exists given its endpoints. In essence, an uncertain graph G represents a probability distribution over the set of all *implicated graphs* of G. Each implicated graph is an exact graph obtained by selecting some vertices from the vertex set of G, followed by selecting some edges, whose endpoints are among the selected vertices, from the edge set of G. Each implicated graph represents a possible structure in which the uncertain graph G may exist in practice.

Identification of densely connected vertices is a fundamental problem in graph mining. Specifically, a *clique* is a set of vertices with each pair of vertices connected by an edge, and a *maximal clique* is a clique that is not a subset of any other cliques. In practice, cliques or maximal cliques are often viewed as cores of dense structures in graphs. We take the following example to show that it is also of practical importance to discover cliques from an uncertain graph.

Example 2: A *protein complex* is a group of proteins that interact with each other at the same time and place in cells to complete some specific biological functions. Accurate protein complex prediction from a PPI network can serve as an inexpensive guide for biological experiments to discover new protein complexes. It has been shown that protein complexes generally correspond to densely connected vertices in PPI networks, and cliques typically serve as cores of protein complexes [3]. Since PPI networks are actually uncertain graphs, we should discover cliques occurring with high probability.

This paper investigates the problem of finding top-k maximal cliques in an uncertain graph. Due to the uncertain nature, a set of vertices may not form a maximal clique in all implicated graphs. Instead, each vertex set has a probability of being a maximal clique across all implicated graphs. This probability is called *the maximal-clique probability*. The top-k maximal cliques are then defined as a collection of k vertex sets with the largest maximal-clique probabilities. Moreover, only cliques containing at least s vertices are considered in this paper since small cliques are generally lack of practical significance. To the best of our knowledge, this paper is the first one to investigate this problem. The main contributions of this paper are summarized as follows.

1) The problem of finding top-k maximal cliques in an uncertain graph is motivated and formalized.
2) It is proved that the maximal-clique probability of a vertex set can be computed in polynomial time.
3) A branch-and-bound algorithm is proposed to find top-k maximal cliques in an uncertain graph, which adopts four efficient pruning rules, a new searching strategy and two preprocessing methods.
4) Experiments were performed on real uncertain graphs to evaluate the performance of the proposed algorithm.
5) The proposed algorithm was applied in protein complex prediction to verify its advantage in improving the accuracy and robustness of prediction.

II. Related Work

The prior work related to this paper is reviewed in the full version of this paper [4].

III. Problem Definition

Definition 1: An *uncertain graph* is a system $G = (V, E, P_V, P_E)$, where (V, E) is an undirected graph, $P_V : V \rightarrow [0, 1]$ is a function assigning existence probability values to the vertices in V, and $P_E : E \rightarrow [0, 1]$ is a function assigning existence probability values to the edges in E upon the condition that the endpoints of each edge exist.

The existence probability, $P_V(v)$, of a vertex v is the probability of v existing in practice. The conditional existence probability, $P_E(e|u, v)$, of an edge $e = (u, v)$ is the probability of e existing between vertices u and v upon the condition that both u and v exist in practice. Thus, a graph in classical graph theory, which is called *exact graph* in this paper, is a special uncertain graph with existence probabilities of 1 on all vertices and conditional existence probabilities of 1 on all edges. Essentially, an uncertain graph *implicates* a set of exact graphs, each of which represents a possible structure in which the uncertain graph may exist in practice.

Definition 2: An exact graph $G' = (V', E')$ is *implicated* by an uncertain graph $G = (V, E, P_V, P_E)$, denoted by $G \Rightarrow G'$, if and only if $V' \subseteq V$ and $E' \subseteq E \cap (V' \times V')$, i.e. the endpoints of each edge in E' are contained in V'.

In this paper, we assume that both the existence probabilities of vertices and the conditional existence probabilities of edges of an uncertain graph are *mutually independent*. Based on this assumption, the probability of an uncertain graph $G = (V, E, P_V, P_E)$ implicating an exact graph $G' = (V', E')$ is

$$\Pr(G \Rightarrow G') = \prod_{v \in V'} P_V(v) \prod_{v \in V \setminus V'} (1 - P_V(v)) \prod_{e=(u,v) \in E'} P_E(e|u, v) \prod_{e=(u,v) \in E \cap (V' \times V') \setminus E'} (1 - P_E(e|u, v)), \quad (1)$$

where $E \cap (V' \times V')$ is the set of edges with both endpoints in V'. The fuction $\Pr(G \Rightarrow G')$ given in Equation 1 defines a probability distribution over all implicated graphs of G.

Example 3: Figure 1(a) shows an uncertain graph, in which the real number on each vertex is the existence probability of the vertex, and the real number on each edge is the conditional existence probability of the edge given its endpoints. The probability distribution over all 18 implicated graphs of the uncertain graph is illustrated in Figure 1(b).

Definition 3: A *clique* in an undirected exact graph is a set C of vertices such that for every two vertices $u, v \in C$, there exists an edge connecting u and v. The *size* of a clique is the number of vertices it contains. A clique is called *maximal* if it is not a subset of any other cliques.

Let $G = (V, E, P_V, P_E)$ be an uncertain graph with Ω as its set of implicated graphs. The probability of a vertex set $C \subseteq V$ being a maximal clique across all implicated graphs of G is given by

$$\sum_{G' \in \Omega, \text{ and } C \text{ is a maximal clique in } G'} \Pr(G \Rightarrow G'), \quad (2)$$

which is called *the maximal-clique probability* of C. Thus, the problem to be solved in this paper can be defined as follows.

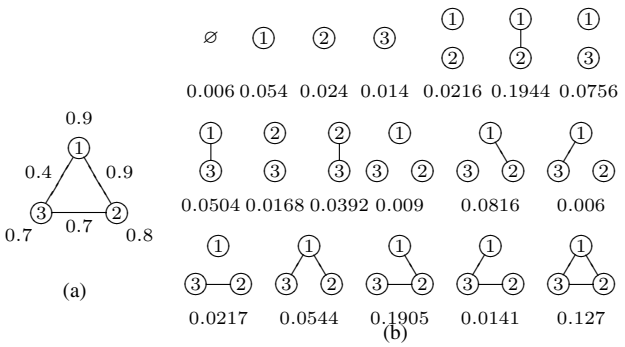

Fig. 1. An uncertain graph and the probability distribution over all its implicated graphs.

The Top-k Maximal Cliques Problem
Input: an uncertain graph G and positive integers k and s.
Output: a collection $\mathcal{F}$ of k vertex sets of G such that

1) each vertex set in $\mathcal{F}$ has size at least s, and
2) for any vertex set $C \in \mathcal{F}$ and any vertex set $C' \notin \mathcal{F}$, the maximal-clique probability of C is not less than the maximal-clique probability of C'.

The top-k maximal cliques problem is NP-hard since it has *the maximal cliques enumeration problem* [5] as a special case when $k = \infty$, $s = 1$, and G is an exact graph.

IV. Computing Maximal-Clique Probability

This section shows that the maximal-clique probability of a vertex set can be computed in polynomial time. Let $\widehat{G} = (V, E)$ denote the exact graph obtained by removing uncertainties from an uncertain graph $G = (V, E, P_V, P_E)$. It is easy to derive the following lemma from Definition 2.

Lemma 1: For an uncertain graph $G = (V, E, P_V, P_E)$, a set of vertices $C \subseteq V$ is a clique in some implicated graphs of G if and only if C is a clique in $\widehat{G}$.

Lemma 1 shows that the maximal-clique probability of a vertex set C is zero if C is not a clique in $\widehat{G}$. In what follows, let $cliq(C)$ denote that a vertex set C is a clique across all implicated graphs of G, and $mcliq(C)$ denote that C is a maximal clique across all implicated graphs of G.

Lemma 2: For a set C of vertices in an uncertain graph $G = (V, E, P_V, P_E)$, if C is a clique in $\widehat{G}$, then the probability that C is a clique across all implicated graphs of G is

$$\Pr(cliq(C)) = \prod_{v \in C} P_V(v) \prod_{u,v \in C, u \neq v} P_E((u, v)|u, v). \quad (3)$$

Proof: The proof can be found in the full version [4]. ■

Lemma 2 gives us a method to compute $\Pr(cliq(C))$. For clarity, we call $\Pr(cliq(C))$ *the clique probability* of C. From Lemma 2, we can easily prove the following corollary that is the key to the design of the pruning rules in Section V.

Corollary 1: Let $G = (V, E, P_V, P_E)$ be an uncertain graph. For any two cliques C and C' in $\widehat{G}$ such that $C \subseteq C'$,

1) $\Pr(cliq(C)) \geq \Pr(cliq(C'))$, and
2) for any vertex $v \in V \setminus C'$,

$$\frac{\Pr(cliq(C \cup \{v\}))}{\Pr(cliq(C))} \geq \frac{\Pr(cliq(C' \cup \{v\}))}{\Pr(cliq(C'))}.$$

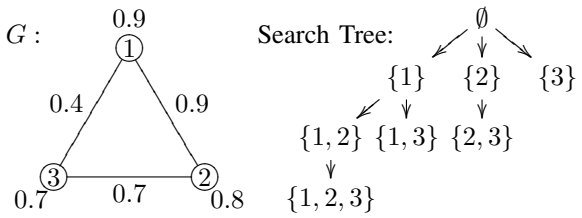

Fig. 2. Search tree for uncertain graph in Figure 1(a).

We also establish the following crucial lemma for computing the maximal-clique probability of a vertex set.

Lemma 3: For an uncertain graph G, let C be a clique in $\widehat{G}$ and $\{C_1, C_2, \ldots, C_t\}$ be the set of all cliques in $\widehat{G}$ such that each C_i is a superset of C and has one more vertex than C. Then, the maximal-clique probability of C is

$$\Pr(mcliq(C)) = \Pr(cliq(C)) \prod_{i=1}^{t} \left(1 - \frac{\Pr(cliq(C_i))}{\Pr(cliq(C))}\right). \quad (4)$$

Proof: The proof can be found in the full version [4]. ■

We proceed to analyze the time complexity of computing Equation 4. Let c be the number of vertices in C and d be the maximum degree of vertices in $\widehat{G}$. All cliques $C_1, C_2, \ldots, C_t$ can be obtained by intersecting the neighbors of all vertices in C, which can be done in $O(cd \log d)$ time. Moreover, we have $t \leq d$. The term $\Pr(cliq(C))$ in Equation 4 can be computed using Equation 3 in $O(c^2)$ time, and each term $\Pr(cliq(C_i))/\Pr(cliq(C))$ can be computed in $O(c)$ time. Thus, Equation 4 can be computed in $O(cd \log d + c^2 + tc) = O(cd \log d + c^2)$ time.

V. Branch-and-Bound Algorithm

In this section, a branch-and-bound algorithm is proposed to find top-k maximal cliques in an uncertain graph. The basic algorithm is presented first, followed by some optimization techniques and preprocessing methods.

A. Basic Algorithm

Given an uncertain graph G with $\widehat{G}$ denoting the exact graph obtained by removing uncertainties from G, let $\prec$ be an arbitrary total order on the vertices of G, e.g. the ascending order of indices of vertices. All cliques in $\widehat{G}$ can be organized into a search tree, where each node represents a distinct clique in $\widehat{G}$, the root represents a trivial clique, i.e. a clique with no vertices, and the parent of each non-root node C represents another clique C' such that $C' \subset C$, $|C| = |C'| + 1$, and the only vertex $v \in C \setminus C'$ satisfies $u \prec v$ for every vertex $u \in C'$. Figure 2 illustrates the search tree for the uncertain graph shown in Figure 1(a). Thus, the problem of finding top-k maximal cliques in G can be transformed into a tree searching problem, that is, seeking for k nodes in the search tree that have maximal-clique probability no less than that of any other node. In the following discussion, we also use a node in the search tree to refer to the vertex set it represents.

The basic algorithm maintains a min-heap $\mathcal{H}_{topk}$ of bounded size k to store the temporary top-k nodes encountered as the algorithm runs. The nodes in $\mathcal{H}_{topk}$ are compared by their maximal-clique probabilities, $\Pr(mcliq(C))$. The maximal-clique probability of the root of $\mathcal{H}_{topk}$ is recorded in a variable τ. Initially, τ is set to zero. The basic algorithm also maintains a max-heap $\mathcal{H}_{ext}$ to store the nodes yet to be searched. The nodes in $\mathcal{H}_{ext}$ are compared by their clique probabilities, $\Pr(cliq(C))$. Initially, for each vertex v of G, the singleton set $\{v\}$ is inserted into $\mathcal{H}_{ext}$, and the key of $\{v\}$ in $\mathcal{H}_{ext}$ is $\Pr(cliq(\{v\})) = P_V(v)$.

The basic algorithm performs *best-first branch-and-bound search* on the search tree. The algorithm repeatedly extracts the node at the root of $\mathcal{H}_{ext}$, which has the largest clique probability among all nodes in $\mathcal{H}_{ext}$. For the extracted node C, the algorithm performs the following steps.

Step 1 (Pruning). If it is not true that $\mathcal{H}_{topk}$ is full and that $\Pr(cliq(C)) \leq \tau$, then there might be top-k maximal cliques in the subtree rooted at C, thus we need to carry out steps 2, 3 and 4 below. Otherwise, the subtree rooted at C can be pruned from the search tree since the maximal-clique probability of any node C' in this subtree is upper bounded by $\Pr(cliq(C))$, thus $\Pr(mcliq(C')) \leq \tau$, i.e. C' can not be in the set of top-k maximal cliques. Hence, the algorithm can skip the following steps. The correctness of this pruning rule will be shown in Lemma 4.

Step 2 (Computing Maximal-Clique Probability). Find the set $N(C)$ of vertices adjacent to all vertices in C, i.e. $N(C) = \{v | \forall u \in C, (u, v) \in E\}$. For each vertex $v \in N(C)$, we obtain a new clique $C' = C \cup \{v\}$ and compute $\Pr(cliq(C'))$ by multiplying $\Pr(cliq(C))$ by $P_V(v) \prod_{u \in C} P_E((u, v)|u, v)$ according to Lemma 2. After that, we compute the maximal-clique probability of C using Equation 4.

Step 3 (Expansion). For each vertex $v \in N(C)$, if $u \prec v$ for every vertex $u \in C$, then $C' = C \cup \{v\}$ is a child of C in the search tree. If $\mathcal{H}_{topk}$ is full and $\Pr(cliq(C')) \leq \tau$, then the subtree rooted at C' can be pruned, otherwise insert C' into $\mathcal{H}_{ext}$ as a promising candidate to be searched.

Step 4 (Update). If $|C| \geq s$, then update the temporary top-k maximal cliques in $\mathcal{H}_{topk}$ according to the following cases.

- If $\mathcal{H}_{topk}$ is not full, then insert C into $\mathcal{H}_{topk}$.
- If $\mathcal{H}_{topk}$ is full and $\Pr(mcliq(C)) > \tau$, then remove the root of $\mathcal{H}_{topk}$ and insert C into $\mathcal{H}_{topk}$ since the root of $\mathcal{H}_{topk}$ can not be in the set of top-k maximal cliques.

Termination. The basic algorithm terminates when either of two events below happens. (1) $\mathcal{H}_{ext}$ is empty. (2) $\mathcal{H}_{topk}$ is full and the clique probability, $\Pr(cliq(C))$, of the root C of $\mathcal{H}_{ext}$ is less than τ. The first event means that all nodes not pruned in the search tree have been searched. The second event indicates that all subtrees rooted at the nodes in $\mathcal{H}_{ext}$ can be pruned. Before termination, all elements in $\mathcal{H}_{topk}$ are output as the answers.

We can easily prove the pruning condition used in the basic algorithm from Corollary 1 and Lemma 3.

Lemma 4: For any two vertex sets C and C' in an uncertain graph, if $C \subseteq C'$, then

$$\Pr(mcliq(C')) \leq \Pr(cliq(C')) \leq \Pr(cliq(C)).$$

B. Optimized Algorithm

In this subsection, some optimization techniques are proposed to speedup the basic algorithm.

1) Optimized Pruning Rules: The pruning rule used in the basic algorithm is very simple and can be used without knowing the neighborhood of a vertex set in the uncertain graph. To further enhance the power of pruning, we develop three optimized pruning rules using the neighborhood information. The proof of the correctness of the optimized pruning rules could be found in the full paper [4].

The Size-Based Pruning. For any node C in the search tree, let $ch(C)$ denote the set of children of C in the search tree If $|C|+|ch(C)| < s$, then the subtree rooted at C can be pruned.

The Look-Ahead Pruning. This pruning rule is only applicable to the nodes on the first $s+1$ levels of the search tree. For any node C with $|C| \leq s$, let $ch(C) = \{C_1, C_2, \ldots, C_t\}$ be the set of children of C in the search tree, and suppose $\frac{\Pr(cliq(C_i))}{\Pr(cliq(C))} \geq \frac{\Pr(cliq(C_{i+1}))}{\Pr(cliq(C))}$ for $1 \leq i \leq t-1$. If the min-heap $\mathcal{H}_{topk}$ is full and

$$\Pr(cliq(C)) \prod_{i=1}^{s-|C|} \frac{\Pr(cliq(C_i))}{\Pr(cliq(C))} \leq \tau, \tag{5}$$

then the subtree rooted at C can be pruned.

The Anti-Monotonicity-Based Pruning. We have an observation that for any two nodes C and C' in the search tree with $C \subseteq C'$, neither the monotone property, i.e. $\Pr(mcliq(C)) \leq \Pr(mcliq(C'))$, nor the anti-monotone property, i.e. $\Pr(mcliq(C)) \geq \Pr(mcliq(C'))$, always holds. However, once some special conditions are satisfied, the maximal-clique probabilities of the nodes in a subtree may become anti-monotone. We utilize this property to develop the anti-monotonicity-based pruning. For any node C in the search tree, let C^* be the child of C that has the maximum value of $\frac{\Pr(cliq(C^*))}{\Pr(cliq(C))}$, and let $sc(C)$ denote the set of super-cliques of C, each of which has $|C|+1$ vertices. If the min-heap $\mathcal{H}_{topk}$ is full, $\Pr(mcliq(C)) \leq \tau$, and

$$\frac{\Pr(cliq(C^*))}{\Pr(cliq(C))} \leq \prod_{S \in sc(C)} \left(1 - \frac{\Pr(cliq(S))}{\Pr(cliq(C))}\right), \tag{6}$$

then the subtree rooted at C can be pruned.

The advantages of all optimized pruning rules can be summarized as follows. First, all of them are very effective as can be verified in the experimental evaluation. Second, all of them can be easily and seamlessly integrated into the basic algorithm without incurring significant overheads.

2) Two-Phase Branch-and-Bound Search: We have two observations on the basic algorithm. First, the best-first search used by the basic algorithm may become breadth-first search in the worst case, thus consuming large amount of memory. Second, the vertex sets of size less than s can't be inserted into $\mathcal{H}_{topk}$, so they can not increase the value of τ. Thus, in the worst case, the pruning rules will not take effect during the search on the first s levels of the search tree.

To deal with this problem, a *two-phase branch-and-bound search strategy* is proposed, which consists of two phases. The first phase is the *depth-first* branch-and-bound search on the first $s+1$ levels of the search tree. The second phase is the *best-first* branch-and-bound search from the $(s+1)^{th}$ level of the search tree.

The detailed description of the optimized algorithm could be found in the full version of the paper [4].

C. Preprocessing Methods

This subsection briefly introduces two preprocessing methods to improve the efficiency of the proposed algorithm.

Degree Filtering. Note that a vertex with degree less than $s-1$ must not be a member of any clique of size at least s. Thus, we can iteratively remove vertices with degree less than $s-1$ as well as edges incident on them until all vertices left have degree no less than $s-1$.

Approximation of the k^{th} Ranked Clique. If we could fill up $\mathcal{H}_{topk}$ with the k^{th} ranked clique at the beginning of the algorithm, then the power of the pruning rules would reach the maximum extent. However, it is NP-hard to obtain the k^{th} ranked clique. Hence, we propose a *beam search* method to approximate the k^{th} ranked clique. The details of the method could be found in the full paper [4].

VI. Experiments

Extensive experimental results on real uncertain graphs could be found in the full version of the paper [4].

VII. Conclusions

This paper motivated and formalized the top-k maximal cliques problem and addressed it using a new branch-and-bound algorithm. The proposed algorithm reduces a significant number of nodes to be traversed in the search tree by the pruning rules, and saves large amount of memory by the two-phase branch-and-bound search. The preprocessing methods also improves the efficiency of the algorithm by reducing the size of the input uncertain graph and increasing the initial pruning power of the pruning rules. The high accuracy and robustness of protein complex prediction resulted from the utilization of top-k maximal cliques verifies the advantage of mining top-k maximal cliques from uncertain graphs.

Acknowledgements. This work was supported partially by the NSF of China under Grant No. 60773063, the NSFC-RGC of China under Grant No. 60831160525, the 973 Program of China under Grant No. 2006CB303000 and the Key Program of the NSF of China under Grant No. 60533110.

References

[1] Z. Zou, J. Li, H. Gao, and S. Zhang, "Mining frequent subgraph patterns from uncertain graph data," *IEEE Trans. Knowledge and Data Eng.*, to be published, TKDESI-2009-03-0254.

[2] S. Suthram, T. Shlomi, E. Ruppin, R. Sharan, and T. Ideker, "A direct comparison of protein interaction confidence assignment schemes," *BMC Bioinformatics*, vol. 7, no. 1, p. 360, 2006.

[3] G. D. Bader and C. W. V. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics*, vol. 4, p. 2, 2003.

[4] Z. Zou, J. Li, H. Gao, and S. Zhang, "Finding top-k maximal cliques in an uncertain graph," *Technichal Report*, 2009.

[5] K. Makino and T. Uno, "New algorithms for enumerating all maximal cliques," in *SWAT*, 2004, pp. 260–272.