Polynomial-Time Algorithm for Finding Densest Subgraphs in Uncertain Graphs

Zhaonian Zou
School of Computer Science and Technology
Harbin Institute of Technology, China
znzou@hit.edu.cn

ABSTRACT

This paper studies the problem of finding the densest subgraph in an uncertain graph. Due to uncertainty in graphs, the traditional definitions of dense subgraphs are not applicable to uncertain graphs. In this paper, we introduce the expected density of an uncertain graph. Based on the expected density, we formalize the problem that, given an uncertain graph $\mathcal{G} = (V, E, P)$ and a set of vertices $R \subseteq V$, finds an induced subgraph $\mathcal{G}' = (V', E', P')$ of $\mathcal{G}$ of the maximum expected density such that $R \subseteq V'$. We show that the optimal solution can be found in $O(nm \log(n^2/m))$ time using maximum flow techniques, where $n = |V|$ and $m = |E|$. Moreover, unlike the existing models of uncertain graphs, the model used in this paper is quite general, which doesn't assume the existence of edges is mutually independent.

Categories and Subject Descriptors

G.2.2 [**Discrete Mathematics**]: Graph Theory — *Graph algorithms*; H.2.8 [**Database Management**]: Database Applications — *Data mining*

General Terms

Measurement, algorithms, performance

Keywords

Uncertain graph, marginal constraint, expected density, parametric maximum flow

1. INTRODUCTION

Dense subgraph discovery is a fundamental problem in the research on graph databases. In literature, a number of algorithms have been proposed for finding dense subgraphs in a given graph, where a variety of definitions of dense subgraphs have been used, e.g., cliques [23], quasi-cliques [1], k-cores [10], k-truss [24], and so on. In this paper, we consider the density measure that assesses the ratio of the number of edges to the number of vertices [12]. More precisely, given a graph $G = (V, E)$, the density of G is defined by $\rho(G) = |E|/|V|$. This definition of density of graph G equivalently measures the average degree of G because $2|E|/|V|$ is equal to the average degree of G. Based on this density measure, many studies have been carried out on the problem of finding a subgraph or an induced subgraph of the maximum density in a given graph [4, 8, 9, 12].

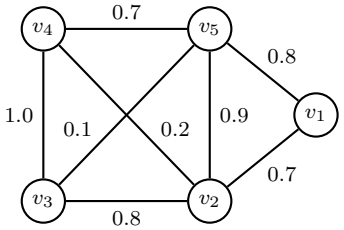

Figure 1: Uncertain graph $\mathcal{G}$.

Recently, uncertainty has been recognized to be intrinsic in large graph databases due to errors of measurements, delayed updates of data, and data integration. Managing and mining uncertain graph data have attracted a lot of research attentions [14, 15, 16, 17, 18, 20, 21, 25, 26, 27]. In our prior work [27], we define an uncertain graph by a triple $\mathcal{G} = (V, E, P)$, where each edge $e \in E$ has a probability of $P(e)$ to exist in practice. Due to uncertainty, the traditional definition of density $\rho(G)$ of a graph G doesn't make sense on an uncertain graph. Consider the uncertain graph $\mathcal{G} = (V, E, P)$ in Figure 1, where the real number on each edge e is $P(e)$. If we think of $\mathcal{G}$ as an exact graph, the density of $\mathcal{G}$ is $8/5$. However, since edges (v_2, v_4) and (v_3, v_5) exist with very low probability, the density of $\mathcal{G}$ should actually be much lower than $8/5$, and be close to $6/5$.

In this paper, we first formalize the problem of finding a densest subgraph in an uncertain graph. According to our uncertain graph model, an uncertain graph $\mathcal{G} = (V, E, P)$ exists as an exact graph $G' = (V, E')$ in practice, where each edge $e \in E$ exists in E' with probability $P(e)$. More formally, we say that $\mathcal{G}$ implicates G. Let $\Omega(\mathcal{G})$ be the set of exact graphs implicated by $\mathcal{G}$. The uncertain graph $\mathcal{G}$ essentially represents a probability mass function p over $\Omega(\mathcal{G})$, where $p(G)$ is equal to the probability of $\mathcal{G}$ implicating G for all $G \in \Omega(\mathcal{G})$. For each exact graph $G = (V, E) \in \Omega(\mathcal{G})$, the density of G is $\rho(G) = |E|/|V|$. Therefore, we evaluate the density of $\mathcal{G}$ by the expected value of density of an exact graph G chosen at random from $\Omega(\mathcal{G})$ according to probability mass function p. Namely, this measure is called the

Eleventh Workshop on Mining and Learning with Graphs (MLG). Chicago, Illinois, USA.

expected density of $\mathcal{G}$. Hence, the densest subgraph problem on uncertain graphs can be formalized as follows. Given an uncertain graph $\mathcal{G} = (V, E, P)$ and a set $R \subseteq V$, find an induced subgraph $\mathcal{G}[V']$ of $\mathcal{G}$ of the maximum expected density such that $R \subseteq V'$. The input R of the problem is a constraint on the output induced subgraph. If $R = \emptyset$, the output is an induced subgraph of $\mathcal{G}$ of the maximum density.

It is worth noting that the model of uncertain graphs proposed in this paper is quite general. Unlike the existing work on managing and mining uncertain graphs [14, 15, 16, 17, 18, 20, 21, 25, 26, 27], we don't assume that the existence of edges of an uncertain graph is mutually independent. In fact, any probability mass function over $\Omega(\mathcal{G})$ that satisfies the *marginal constraint* given later can be used in our work.

Except the theoretical importance, the problem of finding induced subgraphs of the maximum expected density from an uncertain graph also has many practical applications. For example, the densest subgraphs have been used as interesting regions of annotated biological networks, in which valuable cross genome patterns can be found [5]. In fact, due to the inherent uncertainty of high-throughput biological experiments, biological networks are uncertain graphs [6, 13]. Therefore, it is of practical significance for biologists to find the densest subgraphs from uncertain biological networks to get more reliable patterns.

The densest subgraph problem has also been applied in community detection in large networks [7]. Indeed, a substantial number of networks such as social networks are uncertain graphs due to the volatile nature of relationships [2]. Therefore, it is very important for analysts to find subgraphs of the maximum expected density from uncertain social networks to get more reliable communities.

The traditional densest subgraph problem defined on exact graphs has attracted considerable research attentions. Goldberg [12] proposed an algorithm that requires $O(\log n)$ maximum flow computations to find a subgraph of the maximum density. Charikar [9] developed a simple greedy algorithm that finds a subgraph of density within a factor 2 of the optimum. Most recently, Bahmani et al. [7] studied the problem in a data stream model, and presented algorithms that find a subgraph of density within a factor $2(1+\epsilon)$ of the optimum by making $O(\log_{1+\epsilon} n)$ passes over the input graph stream, where $\epsilon > 0$. For the variant of the problem with size constraint, Anderson et al. [4] gave a 3-approximation algorithm for the problem of finding an densest subgraph induced by at least k vertices. Bhaskara et al. [8] studied the problem of finding an densest subgraph induced by exactly k vertices, and showed that the problem can be approximated within a ratio of $O(n^{1/4})$ in $n^{O(\log n)}$ time.

Although the existing algorithms for the densest subgraph problem on exact graphs guarantee good approximation ratios, they can't be used on uncertain graphs. From the aspect of semantics, all these algorithms don't consider uncertainties, so the outputs of the algorithms are unable to be explained with respect to uncertainties. In addition, while some algorithms find densest subgraphs that satisfy size constraints, they can't find a subgraph of the maximum expected density that consists of a set R of specified vertices.

In this paper, we first study the special case of the problem in which the input R is an empty set. That is, the output of the problem is an induced subgraph of the input uncertain graph $\mathcal{G} = (V, E, P)$ of the maximum expected density. We show that this problem is equivalent to the problem of finding a densest induced subgraph in a weighted exact graph. Thus, it can be solved in $O(nm \log(n^2/m))$ time [11], where $n = |V|$ and $m = |E|$.

We next study the problem when the input R is not an empty set. The method is very interesting. Let $\lambda \geq 0$ be a real value that we guessed for the maximum expected density of an induced subgraph that contains R. We reduce the densest subgraph problem to the problem of searching the desired value of λ. Interestingly, we can tell whether λ is too big or too small by computing a minimum cut of a flow network constructed with respect to λ. We show that, starting from an arbitrary guessed value of λ, we can find the desired value of λ by carrying out at most $n + 1$ minimum cut computations, which in turn can be solved by maximum flow techniques. When the desired value of λ is found, we can construct an induced subgraph of the maximum expected density that contains R from the minimum cut of the flow network constructed with respect to the desired value of λ. Note that the computation shared by the series of minimum cut computations can be saved by parametric maximum flow techniques. Thus, the densest induced subgraph that contains R can be found in $O(nm \log(n^2/m))$ time by carrying out the parametric maximum flow algorithm [11] implemented using dynamic trees [22], where $n = |V|$ and $m = |E|$.

The rest of the paper is organized as follows. Section 2 defines the densest subgraph problem on uncertain graphs. Section 3 presents a method for finding a densest induced subgraph of the input uncertain graph $\mathcal{G}$ when the input set R is empty. Section 4 gives an algorithm for finding the densest induced subgraph containing R when the input R is not empty. Finally, the paper is concluded in Section 5.

2. PROBLEM STATEMENT

In this section, we introduce a model of uncertain graphs, define the expected density of an uncertain graph, and give a formal statement of the densest subgraph problem on uncertain graphs. We also introduce some helpful notation.

2.1 Uncertain Graphs

An *uncertain graph* is a triple $\mathcal{G} = (V, E, P)$ in which V is a set of vertices, E is a set of edges, and P is a function from E to $(0, 1]$ that associate each edge $e \in E$ with a quantity $P(e) \in (0, 1]$ which represents the probability of e existing in practice. If $P(e) = 1$, edge e certainly exists.

Because it is uncertain whether an edge e with $P(e) < 1$ exists in practice, an uncertain graph $\mathcal{G} = (V, E, P)$ actually exists as an exact graph $G = (V, E')$ which satisfies that $\{e | e \in E, P(e) = 1\} \subseteq E' \subseteq E$, that is, (1) all the edges e with $P(e) = 1$ exist, and (2) some of the edges e with $P(e) < 1$ may be absent. Following up the terminology in [27], we say that the uncertain graph $\mathcal{G} = (V, E, P)$ *implicates* the exact graph $G = (V, E')$, denoted by $\mathcal{G} \Rightarrow G$. Let $\Omega(\mathcal{G})$ be the set of exact graphs implicated by $\mathcal{G}$. One can readily verify that $|\Omega(\mathcal{G})| = 2^{|\{e | e \in E, P(e) < 1\}|}$.

Given an uncertain graph $\mathcal{G} = (V, E, P)$, if the existence of edges of $\mathcal{G}$ is mutually independent, the probability that $\mathcal{G}$ implicates an exact graph $G = (V, E')$ is given by

$$\Pr[\mathcal{G} \Rightarrow G] = \prod_{e \in E'} P(e) \cdot \prod_{e \in E \setminus E'} (1 - P(e)) \ .$$

Therefore, the function $p(x) = \Pr[\mathcal{G} \Rightarrow x]$ is a probability mass function over $\Omega(\mathcal{G})$. For a proof, please refer to [27].

In this paper, we don't assume that the existence of edges of an uncertain graph $\mathcal{G} = (V, E, P)$ is independent. In fact, any probability mass function $p(x) = \Pr[\mathcal{G} \Rightarrow x]$ over $\Omega(\mathcal{G})$ that satisfies the following property can be used.

Marginal constraint: For all $e \in E$, we require that

$$\sum_{G=(V,E')\in\Omega(\mathcal{G}),e\in E'} \Pr[\mathcal{G} \Rightarrow G] = P(e) \ .$$

Given an uncertain graph $\mathcal{G} = (V, E, P)$ and a set of vertices $V' \subseteq V$, we call $\mathcal{G}' = (V', E', P')$ a *subgraph* of $\mathcal{G}$, denoted by $\mathcal{G}' \subseteq \mathcal{G}$, if $E' \subseteq E$ and $P' = P|_{E'}$, i.e., $P'(e) = P(e)$ for all $e \in E'$. Let $p(x) = \Pr[\mathcal{G} \Rightarrow x]$ be the probability mass function over $\Omega(\mathcal{G})$ which satisfies the marginal constraint. We define, for all $G' \in \Omega(\mathcal{G}')$,

$$\Pr[\mathcal{G}' \Rightarrow G'] = \sum_{G\in\Omega(\mathcal{G}),G'\subseteq G} \Pr[\mathcal{G} \Rightarrow G] \ ,$$

where $G' \subseteq G$ represents that G' is a subgraph of G. One can verify that the function $p'(x) = \Pr[\mathcal{G}' \Rightarrow x]$ is a probability mass function over $\Omega(\mathcal{G}')$, and satisfies the marginal constraint with respect to $\mathcal{G}'$.

Given a set $V' \subseteq V$, the subgraph of an uncertain graph $\mathcal{G} = (V, E, P)$ *induced* by V', denoted by $\mathcal{G}[V']$, is the uncertain graph $\mathcal{G}' = (V', E', P')$, where $E' = \{(u, v)|(u, v) \in E, u \in V', v \in V'\}$ and $P' = P|_{E'}$. For convenience, we denote E' by $E[V']$.

2.2 Densest Subgraph Problem on Uncertain Graphs

The *density* of an exact graph $G = (V, E)$, denoted by $\rho(G)$, is defined by $\rho(G) = |E|/|V|$. Since $2|E|/|V|$ is equal to the average degree of G, $\rho(G)$ essentially measures the average degree of G.

Based on the model of uncertain graphs, we are now ready to define the density of an uncertain graph. Given an uncertain graph $\mathcal{G} = (V, E, P)$, the *expected density* of $\mathcal{G}$, denoted by $\bar{\rho}(\mathcal{G})$, is defined by

$$\bar{\rho}(\mathcal{G}) = \sum_{G\in\Omega(\mathcal{G})} \rho(G) \Pr[\mathcal{G} \Rightarrow G] \ .$$

In other words, $\bar{\rho}(\mathcal{G})$ is the expected value of density of an exact graph G chosen at random from $\Omega(\mathcal{G})$ according to probability mass function $p(G) = \Pr[\mathcal{G} \Rightarrow G]$, that is, $\bar{\rho}(\mathcal{G}) = \mathbf{E}[\rho(G)]$.

Thus, the *densest subgraph problem* on uncertain graphs can be stated as follows:

Input: an uncertain graph $\mathcal{G} = (V, E, P)$ and a set $R \subseteq V$, where $|V| = n$, $|E| = m$, and the probability mass function $p(x) = \Pr[\mathcal{G} \Rightarrow x]$ over $\Omega(\mathcal{G})$ satisfies the marginal constraint.

Output: the induced subgraph $\mathcal{G}[V']$ of $\mathcal{G}$ of the maximum expected density such that $R \subseteq V'$.

3. METHOD FOR FINDING DENSEST SUBGRAPHS

In this section, we investigate the special case of the densest subgraph problem in which the input $R = \emptyset$. That is, the output of the problem is the induced subgraph of $\mathcal{G}$ of the maximum expected density. The main theorem of the section is as follows.

Theorem 1. Given an uncertain graph $\mathcal{G} = (V, E, P)$, where $|V| = n$, $|E| = m$, and the probability mass function $p(x) = \Pr[\mathcal{G} \Rightarrow x]$ over $\Omega(\mathcal{G})$ satisfies the marginal constraint, the induced subgraph of $\mathcal{G}$ of the maximum expected density can be computed in $O(nm \log(n^2/m))$ time.

As a proof of the theorem, we provide a method in the rest of this section to find the induced subgraph of the maximum expected density. We first show the following proposition.

Proposition 1. Given an uncertain graph $\mathcal{G} = (V, E, P)$, the expected density of a subgraph $\mathcal{G}' = (V', E', P')$ of $\mathcal{G}$ can be evaluated by

$$\bar{\rho}(\mathcal{G}') = \frac{1}{|V'|} \sum_{e\in E'} P(e) \ . \quad (1)$$

Proof. For all $e \in E$, let X_e be a random variable following the Bernoulli distribution below.

$$\Pr[X_e = 1] = P(e) \ ,$$
$$\Pr[X_e = 0] = 1 - P(e) \ .$$

In other words, $X_e = 1$ represents the event that e exists in practice, and $X_e = 0$ represents the event that e doesn't exist in practice. Then, we have $\mathbf{E}[X_e] = P(e)$.

Let G be an exact graph chosen at random from $\Omega(\mathcal{G}')$ according to probability mass function $p(G) = \Pr[\mathcal{G}' \Rightarrow G]$. The number of edges of G is therefore $\sum_{e\in E'} X_e$. Thus, the density of G is

$$\rho(G) = \frac{1}{|V'|} \sum_{e\in E'} X_e \ .$$

Due to the definition of expected density, we have $\bar{\rho}(\mathcal{G}') = \mathbf{E}[\rho(G)]$. By the linearity of expectation,

$$\bar{\rho}(\mathcal{G}') = \frac{1}{|V'|} \sum_{e\in E'} \mathbf{E}[X_e] = \frac{1}{|V'|} \sum_{e\in E'} P(e) \ .$$

Thus, the proposition holds. □

If we think of $P(e)$ as the weight of edge e, then Eq. (1) is identical to the density of a weighted exact graph, that is, the ratio of the sum of weights of edges to the number of vertices [12]. Thus, an induced subgraph of the uncertain graph $\mathcal{G} = (V, E, P)$ of the maximum expected density can be computed by Goldberg's algorithm [12] or a parametric maximum flow algorithm [11] which were used to find a densest induced subgraph in a weighted exact graph. Let $n = |V|$ and $m = |E|$. Goldberg's algorithm runs in $O(p(n, m)(p(n, m) + q(n, m)))$ time if the maximum flow algorithm uses $O(p(n, m))$ comparisons and $O(q(n, m))$ additions, where $p(n, m)$ and $q(n, m)$ are polynomials in n and m [12]. The parametric maximum flow algorithm implemented using dynamic trees runs in $O(nm \log(n^2/m))$ time [11], which is apparently faster than Goldberg's algorithm. This completes the proof of Theorem 1.

4. ALGORITHM FOR FINDING DENSEST SUBGRAPHS CONTAINING SPECIFIED VERTICES

In this section, we study the densest subgraph problem in which the input $R \neq \emptyset$. That is, the output of the problem is the induced subgraph of $\mathcal{G}$ of the maximum expected density which contains all the vertices in R. The main theorem of the section is as follows.

Theorem 2. Given an uncertain graph $\mathcal{G} = (V, E, P)$ and a set $R \subseteq V$, where $|V| = n$, $|E| = m$, and the probability mass function $p(x) = \Pr[\mathcal{G} \Rightarrow x]$ over $\Omega(\mathcal{G})$ satisfies the marginal constraint, the induced subgraph $\mathcal{G}[V']$ of $\mathcal{G}$ of the maximum expected density with $R \subseteq V'$ can be computed in $O(nm \log(n^2/m))$ time.

To prove this theorem, we propose an algorithm based on maximum flow techniques [3] in the rest of this section. We define

$$p = \sum_{e \in E} P(e) \ ,$$

and

$$p_v = \sum_{e=(v,u) \in E} P(e) \qquad \text{for all } v \in V \ .$$

Let $\lambda \geq 0$ be a real value that we guessed for the maximum expected density of an induced subgraph of $\mathcal{G}$ which consists of all the vertices in R. Then, we construct a flow network $G_N = (V_N, E_N)$ with respect to λ as follows. The vertex set of G_N is $V_N = V \cup \{s, r, t\}$, where s is the source, and t is the sink. For all $v \in R$, there is an arc $(r, v) \in E_N$ with capacity $c(r, v) = \infty$. For all $v \in V \setminus R$, there is an arc $(r, v) \in E_N$ with capacity $c(r, v) = p$. For all $v \in V$, there is an arc $(v, t) \in E_N$ with capacity $c(v, t) = 2\lambda + p - p_v$. For all $e = (u, v) \in E$, there is an arc $(u, v) \in E_N$ with capacity of $c(u, v) = P(e)$, and an arc $(v, u) \in E_N$ with capacity $c(v, u) = P(e)$. Moreover, there is an arc $(s, r) \in E_N$ with capacity $c(s, r) = np$. More precisely,

$$\begin{aligned}
&G_N = (V_N, E_N) \ , \\
&V_N = V \cup \{s, r, t\} \ , \\
&E_N = \{(s, r)\} \cup \{(r, v) | v \in V\} \cup \{(v, t) | v \in V\} \\
&\qquad \cup \{(u, v) | (u, v) \in E\} \cup \{(v, u) | (u, v) \in E\} \ , \\
&c(r, v) = \infty && \text{for all } v \in R \ , \\
&c(r, v) = p && \text{for all } v \in V \setminus R \ , \\
&c(v, t) = 2\lambda + p - p_v && \text{for all } v \in V \ , \\
&c(u, v) = P(e) && \text{for all } e = (u, v) \in E \ , \\
&c(v, u) = P(e) && \text{for all } e = (u, v) \in E \ , \\
&c(s, r) = np \ .
\end{aligned}$$

Therefore, $|V_N| = n + 3$, and $|E_N| = 2(n + m) + 1$. Figure 2 illustrates the flow network G_N.

Let $\bar{X} = V_N \setminus X$ for any $X \subseteq V_N$. We say that $(X, \bar{X})$ is a *cut* of G_N if $s \in X$ and $t \in \bar{X}$. The capacity of $(X, \bar{X})$, denoted by $c(X, \bar{X})$, is defined by

$$c(X, \bar{X}) = \sum_{e=(u,v) \in E_N, u \in X, v \in \bar{X}} c(u, v) \ ,$$

that is, the sum of capacities of arcs from a vertex in X to a vertex in $\bar{X}$. A cut of the minimum capacity is said to be a *minimum cut*. For more knowledge on network flows, please refer to [3].

The following two lemmas show that if vertex r is on the source side X of a minimum cut $(X, \bar{X})$ of G_N, all the vertices in R are also on the source side X; if r is on the sink side $\bar{X}$ of $(X, \bar{X})$, then $(X, \bar{X}) = (\{s\}, V \cup \{r, t\})$.

Lemma 1. Let $(X, \bar{X})$ be a minimum cut of the flow network G_N constructed with respect to a specific $\lambda \geq 0$. If $r \in X$, then $R \subseteq X \setminus \{s, r\}$.

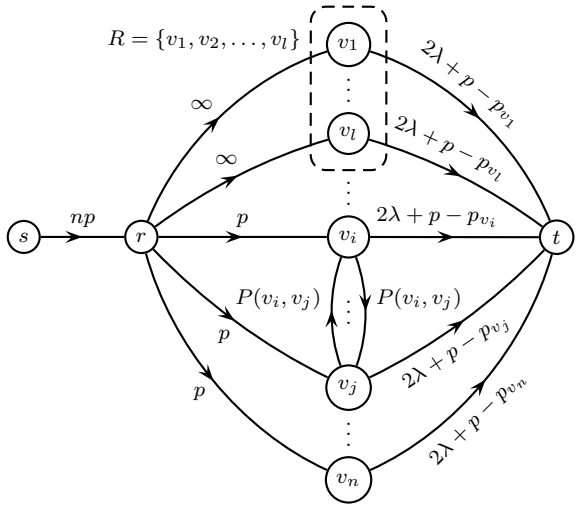

Figure 2: Flow network G_N.

Proof. Let $S = X \setminus \{s, r\}$ and $T = \bar{X} \setminus \{t\}$. If $R \subseteq S$, the capacity of $(X, \bar{X})$ is

$$\begin{aligned}
&c(X, \bar{X}) \\
&= \sum_{v \in T} c(r, v) + \sum_{v \in S} c(v, t) + \sum_{e=(u,v) \in E, u \in S, v \in T} c(u, v) \\
&= p|T| + p|S| + 2\lambda|S| - \sum_{v \in S} p_v + \sum_{e=(u,v) \in E, u \in S, v \in T} P(e) \\
&= np + 2|S|(\lambda - \bar{\rho}(\mathcal{G}[S])) \ . \qquad (2)
\end{aligned}$$

Otherwise, $c(X, \bar{X}) = \infty$. Thus, $R \subseteq S$. □

Lemma 2. Let $(X, \bar{X})$ be a minimum cut of the flow network G_N constructed with respect to a specific $\lambda \geq 0$. If $r \in \bar{X}$, then $X = \{s\}$.

Proof. Let $S = X \setminus \{s\}$ and $T = \bar{X} \setminus \{r, t\}$. If $S = \emptyset$, then $c(X, \bar{X}) = np$. Otherwise, the capacity of $(X, \bar{X})$ is

$$\begin{aligned}
c(X, \bar{X}) &= c(s, r) + \sum_{v \in S} c(v, t) + \sum_{e=(u,v) \in E, u \in S, v \in T} c(u, v) \\
&= np + \sum_{v \in S} (2\lambda + p - p_v) + \sum_{e=(u,v) \in E, u \in S, v \in T} P(e) \\
&> np \ ,
\end{aligned}$$

where the inequality holds because $\lambda \geq 0$, $p \geq p_v$ for all $v \in S$, and $P(e) > 0$ for all $e \in E$. Thus, $X = \{s\}$. □

The following theorem shows that we can tell whether the guessed value λ is too big or too small by computing a minimum cut of the flow network G_N constructed with respect to λ. Hereafter, we require that the maximum flow algorithm used in our method finds the minimum cut $(X, \bar{X})$ such that $|X|$ is the largest over all minimum cuts of G_N.

Theorem 3. Let ρ^* be the maximum expected density of an induced subgraph of $\mathcal{G}$ that consists of all the vertices in R, and let $(X, \bar{X})$ be a minimum cut of the network G_N constructed with respect to λ. We have the following results:

1. $\lambda < \rho^*$ if and only if $r \in X$ and $c(X, \bar{X}) < np$;
2. $\lambda > \rho^*$ if and only if $X = \{s\}$;

3. $\lambda = \rho^*$ if and only if $r \in X$ and $c(X, \bar{X}) = np$;

4. If $\lambda = \rho^*$, then $R \subseteq X \setminus \{s, r\}$, and the expected density of the subgraph of $\mathcal{G}$ induced by $X \setminus \{s, r\}$ is ρ^*.

Proof. We first prove result 1.

Efficacy of result 1: Let $S = X \setminus \{s, r\}$. Since $r \in X$, we have $R \subseteq S$ by Lemma 1. Then, by Eq. (2), we have $c(X, \bar{X}) = np + 2|S|(\lambda - \bar{\rho}(\mathcal{G}[S]))$. Since $c(X, \bar{X}) < np$, we get $\lambda < \bar{\rho}(\mathcal{G}[S])$. Thus, $\lambda < \rho^*$.

Necessity of result 1: Let $V^* \subseteq V$ be the set of vertices such that $R \subseteq V^*$ and $\bar{\rho}(\mathcal{G}[V^*]) = \rho^*$, and let $X^* = V^* \cup \{s, r\}$. By Eq. (2), we have $c(X^*, \bar{X}^*) = np + 2|V^*|(\lambda - \rho^*)$. Since $\lambda < \rho^*$, it yields that $c(X^*, \bar{X}^*) < np$. Since $(X, \bar{X})$ is a minimum cut, we have $c(X, \bar{X}) \leq c(X^*, \bar{X}^*) < np$.

We also have $r \in X$ because if $r \notin X$, then $X = \{s\}$ by Lemma 2, and hence $c(X, \bar{X}) = np$. Therefore, $(X, \bar{X})$ can not be a minimum cut, which leads to a contradiction.

Then, we prove result 2.

Efficacy of result 2: Since $X = \{s\}$, we have $c(X, \bar{X}) = np$. For any subset $S \subseteq V$ with $R \subseteq S$, the capacity of $(S \cup \{s, r\}, (V \setminus S) \cup \{t\})$ is greater than np because otherwise, $(X, \bar{X})$ will not be a minimum cut that has $|X|$ maximum. Then, by Eq. (2), we have $np + 2|S|(\lambda - \bar{\rho}(\mathcal{G}[S])) > np$, that is, $\lambda > \bar{\rho}(\mathcal{G}[S])$. Therefore, $\lambda > \max_S \bar{\rho}(\mathcal{G}[S]) = \rho^*$.

Necessity of result 2: We first prove that $r \notin X$. Assume $r \in X$ and let $S = X \setminus \{s, r\}$. It follows from Lemma 1 that $R \subseteq S$. Then, by Eq. (2), we have $c(X, \bar{X}) = np + 2|S|(\lambda - \bar{\rho}(\mathcal{G}[S]))$. Since $\lambda > \rho^*$, it produces that $\lambda > \bar{\rho}(\mathcal{G}[S])$. Thus, $c(X, \bar{X}) > np$. Since the capacity of the cut $(\{s\}, V \cup \{s, r\})$ is np, $(X, \bar{X})$ is certainly not a minimum cut, which leads to a contradiction. Therefore, $r \notin X$. By Lemma 2, we have $X = \{s\}$.

Next, we prove result 3.

Efficacy of result 3: We first prove that $\lambda \leq \rho^*$. Let $S = X \setminus \{s, r\}$. Since $r \in X$, we have $R \subseteq S$ by Lemma 1. Then, by Eq. (2), we have $c(X, \bar{X}) = np + 2|S|(\lambda - \bar{\rho}(\mathcal{G}[S]))$. Since $c(X, \bar{X}) = np$, it produces that $\lambda = \bar{\rho}(\mathcal{G}[S])$. Thus, $\lambda \leq \rho^*$.

Then, we prove that $\lambda \geq \rho^*$. Assume $\lambda < \rho^*$ and let V^* be the set of vertices such that $R \subseteq V^*$ and $\bar{\rho}(\mathcal{G}[V^*]) = \rho^*$. The capacity of $(V^* \cup \{s, r\}, (V \setminus V^*) \cup \{t\})$ is $np + 2|V^*|(\lambda - \rho^*)$, which is less than np. Therefore, $(X, \bar{X})$ is not a minimum cut, which leads to a contradiction.

Consequently, $\lambda = \rho^*$.

Necessity of result 3: Without loss of generality, let $V^* \subseteq V$ be the unique set of vertices such that $R \subseteq V^*$, $\bar{\rho}(\mathcal{G}[V^*]) = \rho^*$, and $|V^*|$ is maximized. Since $\lambda = \rho^*$, we have $np + 2|V^*|(\lambda - \rho^*) = np$, i.e., the capacity of $(V^* \cup \{s, r\}, (V \setminus V^*) \cup \{t\})$ is np.

We first show that $r \in X$. Assume $r \notin X$. It follows from Lemma 2 that $X = \{s\}$ and thus $c(X, \bar{X}) = np$. Since $X = \{s\} \subset V^* \cup \{s, r\}$, the maximum flow algorithm does not output $(X, \bar{X})$, which leads to a contradiction.

Since $r \in X$, we have $R \subseteq X \setminus \{s, r\}$ by Lemma 1. For any subset $S \subseteq V$ with $R \subseteq S$ and $S \neq V^*$, the capacity of $(S \cup \{s, r\}, (V \setminus S) \cup \{t\})$ is $np + 2|S|(\lambda - \bar{\rho}(\mathcal{G}[S]))$. Since $\lambda = \rho^* \geq \bar{\rho}(\mathcal{G}[S])$, the capacity of $(S \cup \{s, r\}, (V \setminus S) \cup \{t\})$ is no less than np. In addition, since $|S| < |V^*|$, the maximum flow algorithm will finally produce $X = V^* \cup \{s, r\}$. By Eq. (2), we have $c(X, \bar{X}) = np + 2|V^*|(\lambda - \bar{\rho}(\mathcal{G}[V^*]))$. Since $\lambda = \bar{\rho}(\mathcal{G}[V^*]) = \rho^*$, we have $c(X, \bar{X}) = np$.

Now consider the general case. Let $V_1, V_2, \ldots, V_l$ be the subsets of V such that $R \subseteq V_i$, $\bar{\rho}(\mathcal{G}[V_i]) = \rho^*$, and $|V_i|$ is maximized for $i = 1, 2, \ldots, l$. By above arguments, there must exist an integer $1 \leq i \leq l$ such that $X = V_i \cup \{s, r\}$. Therefore, $r \in X$ and $c(X, \bar{X}) = np$.

Finally, we prove result 4. Let $S = X \setminus \{s, r\}$. Since $r \in X$, we have $R \subseteq S$ by Lemma 1. Then, by Eq. (2), we have $c(X, \bar{X}) = np + 2|S|(\lambda - \bar{\rho}(\mathcal{G}[S]))$. Since $c(X, \bar{X}) = np$ and $\lambda = \rho^*$, we have $\bar{\rho}(\mathcal{G}[S]) = \rho^*$.

Thus, the theorem holds. □

Define

$$p_R = \sum_{e \in E[R]} P(e) \ .$$

Since $|R| \leq |S| \leq n$, and

$$p_R \leq \sum_{e \in E[S]} P(e) \leq p$$

for all $R \subseteq S \subseteq V$, we have

$$\frac{p_R}{n} \leq \bar{\rho}(\mathcal{G}[S]) \leq \frac{p}{|R|}$$

Intuitively, we may use a maximum flow algorithm as a subroutine to test whether λ is too big or too small by Theorem 3, and find the desired value of λ, that is, the maximum expected density of an induced subgraph of $\mathcal{G}$ that consists of all the vertices in R, within interval $[p_R/n, p/|R|]$ by a search method such as binary search, monotonic search, or Megiddo's method [19]. However, since $P(e) \in (0, 1]$ is a real number for all $e \in E$, the number of all possible values of λ is infinite. Thus, the simple search methods such as binary search, monotonic search, or Megiddo's method need not terminate at all.

Interestingly, by the following property of minimum cuts of the flow network G_N, we can find the desired value of λ in polynomial time. Let $c(\lambda)$ be the capacity of a minimum cut of G_N constructed with respect to λ. It is known that $c(\lambda)$ is a piecewise linear and concave function of λ with at most $|V_N| - 1 = n + 2$ breakpoints [11]. Each line segment between two consecutive breakpoints at $\lambda = \lambda_1$ and $\lambda = \lambda_2$, where $\lambda_1 < \lambda_2$, in the graph of $c(\lambda)$ corresponds to a distinct minimum cut which remains a minimum cut for $\lambda_1 \leq \lambda \leq \lambda_2$. By Theorem 3, the rightmost breakpoint on the graph of $c(\lambda)$ is at $\lambda = \rho^*$, and $c(\lambda) = np$ for $\lambda \geq \rho^*$. Figure 3 gives an illustration of the graph of $c(\lambda)$.

Thus, we can compute the maximum expected density ρ^* of an induced subgraph of $\mathcal{G}$ that contains all the vertices in R by searching the rightmost breakpoint on the graph of $c(\lambda)$. This can be done using *parametric maximum flow algorithms* [11]. First, we consider λ as a variable rather than a constant. Therefore, the flow network G_N is a parametric flow network in which the capacities of all arcs into the sink t are linear functions of λ. Then, we use a parametric maximum flow algorithm, e.g., the one proposed by Gallo et al. [11], to find the rightmost breakpoint on the graph of $c(\lambda)$.

Let $(X, \bar{X})$ be a minimum cut of G_N produced by the parametric maximum flow algorithm for $\lambda = \rho^*$ such that $|X|$ is the largest. By Theorem 3, we have $R \subseteq X \setminus \{s, r\}$, and

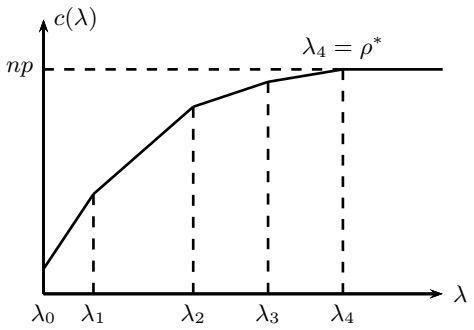

Figure 3: Graph of $c(\lambda)$.

the subgraph of $\mathcal{G}$ induced by $X \setminus \{s, r\}$ has the maximum expected density ρ^*.

We are now ready to provide an algorithm that, given an uncertain graph $\mathcal{G} = (V, E, P)$ and a nonempty set $R \subseteq V$, finds the induced subgraph of $\mathcal{G}$ of the maximum expected density which consists of all the vertices in R. The algorithm, denoted by DS, is presented in Figure 4.

Input: an uncertain graph $\mathcal{G} = (V, E, P)$ and a set of vertices $R \subseteq V$

Output: an induced subgraph of $\mathcal{G}$ of the maximum expected density which consists of all the vertices in R

Step 1. Let λ be a variable representing the guessed value for the maximum expected density ρ^* of an induced subgraph of $\mathcal{G}$ that consists of all the vertices in R. Construct a parametric flow network $G_N = (V_N, E_N)$ with respect to λ.

Step 2. Use the parametric maximum flow algorithm [11] implemented using dynamic trees to find the rightmost breakpoint on the graph of $c(\lambda)$ which is at $\lambda = \rho^*$. Let $(X, \bar{X})$ be the minimum cut of G_N produced by the parametric maximum flow algorithm for $\lambda = \rho^*$.

Step 3. Output the subgraph of $\mathcal{G}$ induced by $X \setminus \{s, r\}$.

Figure 4: Algorithm DS.

Finally, we analyze the time complexity of DS. The parametric flow network $G_N = (V_N, E_N)$ consists of $|V_N| = n+3$ vertices and $|E_N| = 2(n+m)+1$ arcs. Therefore, G_N can be constructed in $O(|V_N| + |E_N|) = O(n+m)$ time in Step 1. The value of ρ^* and the minimum cut $(X, \bar{X})$ of G_N with respect to $\lambda = \rho^*$ can be computed by the parametric maximum flow algorithm implemented using dynamic trees [22] in $O(|V_N||E_N| \log(|V_N|^2/|E_N|)) = O(nm \log(n^2/m))$ time [11] in Step 2. Thus, DS runs in $O(nm \log(n^2/m))$ time.

5. CONCLUSIONS

In this paper, we have given a general model of uncertain graphs in which the existence of edges is not required to be independent. Based on this model, we have defined the concept of expected density of an uncertain graph. We have also shown that, given an uncertain graph $\mathcal{G} = (V, E, P)$ and a set of vertices $R \subseteq V$, we are able to find an induced subgraph $\mathcal{G}' = (V', E', P')$ of $\mathcal{G}$ of the maximum expected density such that $R \subseteq V'$ in $O(nm \log(n^2/m))$ time, where $n = |V|$ and $m = |E|$.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61173023, and the Fundamental Research Funds for the Central Universities of China under Grant No. HIT.NSRIF.201180.

6. REFERENCES

[1] J. Abello, M. Resende, and S. Sudarsky. Massive quasi-clique detection. In *LATIN*, pages 598–612, 2002.

[2] E. Adar and C. Re. Managing uncertainty in social networks. *IEEE Data Eng. Bull.*, 30(2):15–22, 2007.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows - theory, algorithms and applications.* Prentice Hall, 1993.

[4] R. Andersen and K. Chellapilla. Finding dense subgraphs with size bounds. In *WAW*, pages 25–37, 2009.

[5] P. Anderson, A. Thor, J. Benik, L. Raschid, and M.-E. Vidal. Pang: finding patterns in annotation graphs. In *SIGMOD Conference*, pages 677–680, 2012.

[6] S. Asthana, O. D. King, F. D. Gibbons, F. P. Roth, E. Alerting, S. Asthana, O. D. King, F. D. Gibbons, and F. P. Roth. Predicting protein complex membership using probabilistic network reliability. *Genome Research*, 14:1170–1175, 2004.

[7] B. Bahmani, R. Kumar, and S. Vassilvitskii. Densest subgraph in streaming and mapreduce. *PVLDB*, 5(5):454–465, 2012.

[8] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an $o(n^{1/4})$ approximation for densest k-subgraph. In *STOC*, pages 201–210, 2010.

[9] M. Charikar. Greedy approximation algorithms for finding dense components in a graph. In *APPROX*, pages 84–95, 2000.

[10] J. Cheng, Y. Ke, S. Chu, and M. T. Özsu. Efficient core decomposition in massive networks. In *ICDE*, pages 51–62, 2011.

[11] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.*, 18(1):30–55, 1989.

[12] A. V. Goldberg. Finding a maximum density subgraph. Technical report, Univerisity of California at Berkeley, Berkeley, California, USA, 1984.

[13] R. Jiang, Z. Tu, T. Chen, and F. Sun. Network motif identification in stochastic networks. *PNAS*, 103(25):9404–9409, 2006.

[14] R. Jin, L. Liu, and C. C. Aggarwal. Discovering highly reliable subgraphs in uncertain graphs. In *KDD*, pages 992–1000, 2011.

[15] R. Jin, L. Liu, B. Ding, and H. Wang. Distance-constraint reachability computation in uncertain graphs. *PVLDB*, 4(9):551–562, 2011.

[16] G. Kollios, M. Potamias, and E. Terzi. Clustering large probabilistic graphs. *IEEE Trans. Knowl. Data*

Eng., 25(2):325–336, 2013.

[17] J. Li, Z. Zou, and H. Gao. Mining frequent subgraphs over uncertain graph databases under probabilistic semantics. *VLDB J.*, 21(6):753–777, 2012.

[18] L. Liu, R. Jin, C. C. Aggarwal, and Y. Shen. Reliable clustering on uncertain graphs. In *ICDM*, pages 459–468, 2012.

[19] N. Megiddo. Combinatorial optimization with rational objective functions. In *STOC*, pages 1–12, 1978.

[20] O. Papapetrou, E. Ioannou, and D. Skoutas. Efficient discovery of frequent subgraph patterns in uncertain graph databases. In *EDBT*, pages 355–366, 2011.

[21] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. k-nearest neighbors in uncertain graphs. *PVLDB*, 3(1):997–1008, 2010.

[22] D. D. Sleator and R. E. Tarjan. A data structure for dynamic trees. *JCSS*, 26(3):362–391, 1983.

[23] E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theor. Comput. Sci.*, 363(1):28–42, 2006.

[24] J. Wang and J. Cheng. Truss decomposition in massive networks. *PVLDB*, 5(9):812–823, 2012.

[25] Y. Yuan, G. Wang, L. Chen, and H. Wang. Efficient subgraph similarity search on large probabilistic graph databases. *PVLDB*, 5(9):800–811, 2012.

[26] Z. Zou, J. Li, H. Gao, and S. Zhang. Finding top-k maximal cliques in an uncertain graph. In *ICDE*, pages 649–652, 2010.

[27] Z. Zou, J. Li, H. Gao, and S. Zhang. Mining frequent subgraph patterns from uncertain graph data. *IEEE Trans. Knowl. Data Eng.*, 22(9):1203–1218, 2010.