Reliable Clustering on Uncertain Graphs

Lin Liu, Ruoming Jin
Department of Computer Science
Kent State University
Kent, OH, USA
{*lliu, jin*}*@cs.kent.edu*

Charu Aggarwal
T. J. Watson Research Center
IBM
Yorktown Heights, NY, USA
charu@us.ibm.com

Yelong Shen
Department of Computer Science
Kent State University
Kent, OH, USA
yshen@cs.kent.edu

***Abstract*—Many graphs in practical applications are not deterministic, but are probabilistic in nature because the existence of the edges is inferred with the use of a variety of statistical approaches. In this paper, we will examine the problem of clustering uncertain graphs. Uncertain graphs are best clustered with the use of a *possible worlds* model in which the most *reliable* clusters are discovered in the presence of uncertainty. Reliable clusters are those which are not likely to be disconnected in the context of different instantiations of the uncertain graph. In this paper we provide a generalized reliability measurement from two basic intuitions (purity and size balance) to overcome the challenges from standard reliability criterion, and develop a novel k-means algorithm to solve the uncertain clustering problem. We present experimental results which illustrate the effectiveness and efficiency of our model and approachs.**

***Keywords*-uncertain graph; clustering; reliability;**

I. Introduction

With the increasing number of applications in which the edges are constructed in the network through uncertain or statistical inference [4], the problem of mining uncertain graphs has become increasingly important. Examples of such networks include protein-protein interaction networks with experimentally inferred links [5], [25], [9], sensor networks with uncertain connectivity links [19], or social networks, which are augmented with inferred friendship, similarity, or trust links [21], [6]. Therefore, a number of mining problems have recently been studied in the context of uncertain graphs [10], [12], [13], [14], [16], [34], [32], [23], [33], [24], [18], [30], [17], though many fundamental problems such as clustering still remain unexplored.

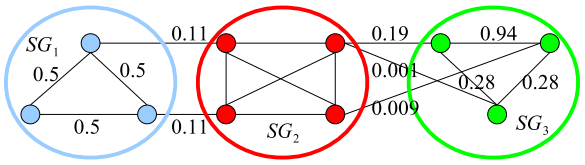


Figure 1: An Motivation Example: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$

An obvious approach is to convert uncertain graph clustering problem into the deterministic scenario by using edge probabilities as edge weights. However, by disregarding the possible world semantic of uncertain graphs, as will be described later, such an approach hence fails to reflect the *connectivity* of uncertain graphs correctly. Connectivity is a fundamental graph property and plays an important role in graph clustering. Let us take uncertain graph $\mathcal{G}$ in Figure 1 for an example. Connectivity of deterministic clusters (subgraphs) is generally measured by the concept of *cut*, which is defined as the sum of the weights of edges across the partitions[26]. The bigger the cut, the harder to separate the two subgraphs. In Figure 1, *cut* $C(SG_1, SG_2) = 0.22$ and $C(SG_2, SG_3) = 0.20$, as implies that SG_2 is harder to be separated from SG_1 than from SG_3. However, with the possible world semantics, we know the probability to separate SG_1 and SG_2 is $(1 - 0.11)^2 = 0.79$, and that to separate SG_2 and SG_3 is $(1-0.19)(1-0.001)(1-0.009) = 0.80$. Hence, in fact SG_2 is closer to SG_3 than to SG_1. This suggests that by ignoring possible world semantics, a deterministic approach could produce very poor results.

It is also worthwhile for us to consider, whether the connectivity of a cluster should only be defined by the nodes inside it. For example, terrorist leaders may not directly contact their members, but instead through the non-members to pass their messages. Proteins interact with each other through other kinds of proteins (enzymes). Therefore, a good uncertain graph clustering criterion is able to capture not only traditional cluster connectivity constraints, but also the connectivity behavior in the context of the overall network.

In this context, standard uncertain graph reliability [8] is a good criterion, because it evaluates the connectivity of a set of nodes in the context of the entire network and meanwhile utilizes the possible world model. However, the use of such a criterion results in a number of challenges, because it is #P-Complete to compute reliability for any candidate set. Furthermore, the number of such candidate sets may be exponentially related to the number of nodes. Therefore, in this paper we propose a *generalized reliability criterion*, which is designed from an information-theoretic perspective, and show that the use of such a criterion enables the design of an extremely simple and efficient version of the k-means algorithm, while retaining the desired qualitative properties.

Specifically, we make the following contributions:

1) To the best of our knowledge, we are the first to

formulate the uncertain graph clustering problem, as it relates to connectivity issues in the presence of uncertainty.
2) We propose an information-theoretic generalized reliability criterion, which is not only qualitatively effective, but also enables very efficient cluster discovery.
3) We conduct extensive experimental studies to demonstrate the effectiveness and efficiency of our algorithms.

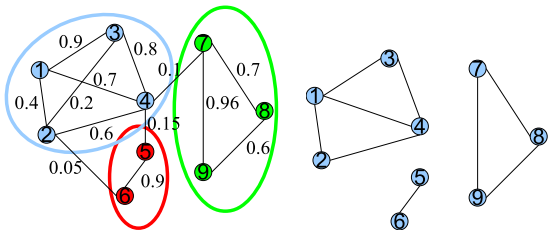
The rest of this paper is organized as follows. In Section II, we introduce the problem formulation. The properties of generalized reliability criterion are introduced in Section III. In Section IV, we leverage this criterion to develop two efficient algorithms for clustering uncertain graphs. The experimental study is presented in Section V. Section VI discusses related work. Finally, we provide conclusions in Section VII.

II. Problem Formulation

In this section, we present the notations, definitions and the problem formulation.

A. Basics

An uncertain graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$, is defined over a set of vertices $\mathcal{V}$, a set of edges $\mathcal{E}$, and a set of probabilities $\mathcal{P}$ of edge existence.



(a) Uncertain Graph $\mathcal{G}$ (b) Possible Graph G_1

(c) Possible Graph G_2 (d) Possible Graph G_3

Figure 2: Uncertain graph and its possible graphs

Let $G = (V_G, E_G)$ be the possible graph (possible world) which is realized by sampling each edge in $\mathcal{G}$ according to the probability $p(e)$ (denoted as $G \sqsubseteq \mathcal{G}$). Clearly, we have $E_G \subseteq \mathcal{E}$ and $V_G \subseteq \mathcal{V}$. The probability $\Pr[G]$ of sampling this possible graph is as follows:

$$\Pr[G] = \prod_{e \in E_G} p(e) \prod_{e \in \mathcal{E} \setminus E_G} (1 - p(e)).$$

There are a total of $2^{|\mathcal{E}|}$ possible graphs, since each edge provides us with a binary sampling decision. Examples of uncertain graphs together with possible world samples are illustrated in Figure 2. In our example, the graph $\mathcal{G}$ has 2^{13} possible subgraphs.

B. Standard Reliability

In connected deterministic graphs, all pairs of nodes are guaranteed to be connected. In probabilistic graphs, the concept of *reliability* is used to generalize this concept by capturing the probability that a set of vertices are connected. More formally, the reliability of a set of vertices is defined as follows.

Definition 1: **(Reliability [8]:)** Given an uncertain graph $\mathcal{G}$, and a set of vertices $V_s \subseteq \mathcal{V}$, the reliability $R(V_s)$ for vertex set V_s is defined as follows:

$$R(V_s) = \sum_{G_i \sqsubseteq \mathcal{G}} \Pr(G_i) \mathcal{I}(V_s, G_i), \tag{1}$$

where $\mathcal{I}(V_s, G_i)$ is 1 when V_s is contained in a connected component in G_i, and 0 otherwise.

The concept of reliability generalizes the connectivity concept to the probabilistic scenario. While deterministic connectivity is a binary value, the reliability value lies in the continuous range $(0, 1)$, and quantifies the probability of the vertices remaining connected in a possible world. It is important to note that the aforementioned definition does not constrain the connectivity to exist only through vertices included in V_s. Rather the vertices of V_s could remain connected with the use of vertices which do not belong to V_s.

The determination of reliable clusters presents several computational changes. First, it is #P-Complete even to *evaluate* the reliability of a *given* set of vertices [8]. Furthermore, natural ways of *enumerating* such candidate vertex sets have an exponential complexity with graph size. While traditional Monte-Carlo methods can alleviate the first problem, the second issue is much harder to address. This serves as our motivation for the development of the concept of generalized reliability.

C. Generalized Reliability

In this subsection, we derive the concept of generalized reliability from two simple intuitions. As a generative model, an uncertain graph $\mathcal{G}$ can generate $2^{|\mathcal{E}|}$ different possible worlds by sampling each edge e with edge probability $p(e)$. Each such sample may break the graph into several connected components, or "fragments". Intuitively, give a possible world G of $\mathcal{G}$, if node u and v are from the same underlying cluster, they should be contained in the same fragment of G. We summarize this desired quality of the clustering as follows.

Desiderata 2.1 (Purity): For each connected component (fragment) of a possible world resulted from uncertain graph

sampling, the number of distinct clusters to which the different nodes belong should be as small as possible, and one of the clusters should dominate the fragment.

More formally, suppose we have the clustering $C_1, C_2, \cdots, C_K$, and $2^{|\mathcal{E}|}$ sampled graphs $\{G_i\}$. For the possible world corresponding to G_i, there are L_i connected components (fragments) denoted by $CC^{i,1}, \cdots, CC^{i,L_i}$. We also have a partition for each component according to the vertex cluster label, and the nodes corresponding to the kth cluster in $CC^{i,j}$ are denoted by $CC_k^{i,j}$. Therefore, we have $CC^{i,j} = \bigcup_{k=1}^{K} CC_k^{i,j}$, and $CC_k^{i,j} \subseteq C_k$. Then, the purity can be defined with the use of cluster label entropy as follows:

$$\mathcal{F}_p = \sum_{G_i \sqsubseteq \mathcal{G}} \Pr(G_i) \sum_{j=1}^{L_i} |CC^{i,j}| H(\bigcup_k CC_k^{i,j}), \quad (2)$$

where

$$H(\bigcup_k CC_k^{i,j}) = -\sum_{k=1}^{K} \frac{|CC_k^{i,j}|}{|CC^{i,j}|} \log(\frac{|CC_k^{i,j}|}{|CC^{i,j}|}),$$

is the entropy of cluster labels for fragment j in possible world G_i.

In addition, a balance in cluster size is particularly important in uncertain graph clustering, because the above definition tends to bias the process towards unbalanced clusters. As a result, if only a purity criterion is used, then the resulting solution will typically contain a single cluster containing most of the nodes. Therefore, in order to alleviate this issue, it is important to add size balance to the clustering criterion.

Desiderata 2.2 (Size Balance): Given a clustering of graph $\mathcal{G}$, the clusters should not be too unbalanced in terms of the number of nodes.

More formally, consider the clustering $C_1, C_2, \cdots, C_K$, where $C_i \subseteq \mathcal{V}$, $C_i \cap C_j = \emptyset, i \neq j$, and $\bigcup_{i=1}^{K} C_i = \mathcal{V}$. Then, in order to make the different values of $|C_i|$ similar, we could maximize the following function:

$$\begin{aligned} \mathcal{F}_e &= \sum_{G_i \sqsubseteq \mathcal{G}} \Pr(G_i) |\mathcal{V}| H(\bigcup_k C_k) \\ &= |\mathcal{V}| H(\bigcup_k C_k), \end{aligned}$$

Here

$$H(\bigcup_k C_k) = -\sum_{k=1}^{K} \frac{|C_k|}{|\mathcal{V}|} \log(\frac{|C_k|}{|\mathcal{V}|})$$

is the entropy of cluster size.

We combine the aforementioned motivations in order to create the following problem definition:

Definition 2: **(Generalized Reliable Clustering:)** Given an uncertain graph $\mathcal{G}$, and cluster number K, obtain a vertex partition $\{C_1, \cdots, C_K\}$, in order to minimize $\mathcal{F} = \mathcal{F}_p - \mathcal{F}_e$.

It turns out that our problem can be reduced to minimizing a much simpler function.

Observation 1: The minimization of $\mathcal{F}$ is equivalent to minimization of the following function:

$$\mathcal{F} = \sum_{G_i \sqsubseteq \mathcal{G}} \Pr(G_i) \sum_{k=1}^{K} |C_k| H(\bigcup_j CC_k^{i,j}). \quad (3)$$

Proof Sketch: First we can see that

$$\begin{aligned} \mathcal{F}_p &= \mathcal{F}_p + \sum_{k=1}^{K} (|C_k| \log |C_k| - |C_k| \log |C_k|) && (4) \\ &\quad + |\mathcal{V}| \log |\mathcal{V}| - |\mathcal{V}| \log |\mathcal{V}| && (5) \\ &= -\sum_{G_i \sqsubseteq \mathcal{G}} \Pr(G_i) \sum_{j=1}^{L_i} \sum_{k=1}^{K} |C_k^{i,j}| \log(\frac{|C_k^{i,j}|}{|C_k|}) && (6) \\ &\quad + \sum_{G_i \sqsubseteq \mathcal{G}} \Pr(G_i) \sum_{j=1}^{L_i} |C^{i,j}| \log(\frac{|CC^{i,j}|}{|\mathcal{V}|}) && (7) \\ &\quad - |\mathcal{V}| \sum_{k=1}^{K} \frac{|C_k|}{|\mathcal{V}|} \log \frac{|C_k|}{|\mathcal{V}|}. && (8) \end{aligned}$$

Equation (7) is a constant, and Equation (8) equals $\mathcal{F}_e$. Therefore, after remove $\mathcal{F}_e$ from $\mathcal{F}_p$, our goal is actually to minimize Equation (6). □

What is the relationship between Equation (3) and the standard reliability definition, as proposed in Equation (1)? One can easily note that $f_d^i = 1 - \mathcal{I}(C_k, G_i)$ is a binary version of $f_c^i = H(\bigcup_j C_k^{i,j})$, which is a continuous function in terms of $p_k^{i,j} = \frac{|C_k^{i,j}|}{|C_k|}$. Therefore, f_c^i is not only relevant to connectivity, but also considers the distribution of the sizes of the different clusters C_k when disconnected. More interestingly, when we replace f_c^i with f_d^i in Equation (3), we can see that to minimize our objective function approximately equals to maximize $\sum_{k=1}^{K} |C_k| \cdot R(C_k)$, which is essentially a cluster-size weighted summation of the standard reliability over different clusters.

D. Re-visiting the Computational Challenge

As mentioned earlier, the Monte-Carlo sampling method can be used in order to estimate the underlying reliability. In this technique, we create N possible-world instantiations (samples) of the uncertain graph with the use of edge-sampling probabilities. The N sampled graphs are denoted by $G_1, \cdots, G_N$. Each sampled graph G_i can be represented with a list of connected components (fragments) $CC^{i,1}, \cdots, CC^{i,L_i}$. Through this sampling, our objective function can be restated as follows:

$$\mathcal{F}_s = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} |C_k| H(\cup_j CC_k^{i,j}). \quad (9)$$

It is evident that $\mathcal{F}_s$ is an unbiased estimator of $\mathcal{F}$, and therefore $E(\mathcal{F}_s) = \mathcal{F}$.

More importantly, by applying the Chernoff-Hoeffding Bound[7], [15], we can determine whether $\mathcal{F}_s$ is close to $\mathcal{F}$ with high probability.

Lemma 1: For any $\epsilon > 0$ and $0 < \delta \leq 1$, with sample size $N \geq \frac{2}{\epsilon^2} \ln \frac{2}{\delta}$, we have

$$\Pr(|\mathcal{F}_s - \mathcal{F}| \geq \epsilon) \leq \delta.$$

The above lemma can help us directly determine the sample size N for given error ϵ and confidence level $1-\delta$. To solve the second computational challenge of candidate set *enumeration*, our generalized reliability criterion helps us directly produce reasonable candidate subgraphs. Therefore, we will first understand some properties of our generalized clustering criterion in the next section.

III. Properties of Generalized Reliability

In this section, we study some fundamental properties of generalized reliability, which enable efficient cluster discovery. Specifically, we study its relationship with the concept of graph coding, and leverage this understanding to develop a coded k-means algorithm. We also characterize the cluster properties through our objective function.

A. Graph Coding Interpretation of Objective Function

The sampling-based cluster objective function of Equation (9) can be understood from a graph coding view, and this can also be used for better cluster interpretability. First, we define the concept of *auxiliary cluster table* - denoted by $\mathcal{T}_i$ - for possible world G_i.

Definition 3: (**Auxiliary Cluster Table)** Given a possible graph G_i, with L_i connected components denoted by $CC^{i,1}, CC^{i,2}, \cdots, CC^{i,L_i}$, and each vertex $v \in \mathcal{V}$ assigned to one of K clusters, the auxiliary cluster table $\mathcal{T}_i$ is constructed as follows. Each row of $\mathcal{T}_i$ corresponds to a cluster and each column to a connected component ($CC^{i,j}$). The cell in kth row and jth column, $\mathcal{T}_i(k,j)$ contains a set of vertices $CC_k^{i,j}$ from the jth component and kth cluster.

An example of the auxiliary cluster table is illustrated in Figure 3(b). The number of rows in the auxiliary cluster table is equal to the number of clusters, and we have shown different cases in Figure 3(b). For example, when there are $K = 3$ clusters, as in Figure 3(a), there are three rows in $\mathcal{T}_i$ (the bottom one in Figure 3(b)). We can see that each $\mathcal{T}_i$ not only provides us a structured view of G_i, but also contains all the information for G_i, including the cluster and fragment identifier for each vertex. Therefore, we can encode each table $\mathcal{T}_i$ for G_i. To represent $\mathcal{T}_i$, we need to code the vertex cluster identifier and fragment identifier (data coding), and to record coding complexity (coding book). Here we only care about data coding for its close relationship with our objective function.

Note that in this paper we utilize the entropy encoding, especially Huffman coding, which is a lossless data compression scheme.

Cluster Identifier Coding: According to *Shannon's source coding theorem*, the best coding length for a cluster label C_k equals $-\log(\frac{|C_k|}{|\mathcal{V}|})$. Here we use $c(C_k)$ to denote the code word. Therefore, the total coding length for entire vertex set $\mathcal{V}$ is

$$\begin{aligned}\mathcal{L}_1 &= \sum_{v \in \mathcal{V}} |c(C_k | v \in C_k)| = -\sum_{k=1}^{K} |C_k| \log(\frac{|C_k|}{|\mathcal{V}|}) \\ &= -|\mathcal{V}| H(\bigcup_k C_k).\end{aligned}$$

For instance, to transmit the cluster information for each vertex in the sampled graph G_i in Figure 3(a), we need $9 \times (-\frac{4}{9}\log\frac{4}{9} - \frac{2}{9}\log\frac{2}{9} - \frac{3}{9}\log\frac{3}{9})$ bits. We can see that $\mathcal{L}_1$ corresponds to our size constraint $\mathcal{F}_e$.

Component Identifier Coding: To represent the connected component identifier of a vertex, we have two differnt cases. Without considering vertex cluster identifiers, and also according to *Shannon's source coding theorem*, the optimal code length for a component identifier $CC^{i,j}$ is $-\log\frac{|C^{i,j}|}{|\mathcal{V}|}$. Here we denote its code word as $c(CC^{i,j})$. Then the expected coding length for a vertex v is $\sum_{G_i \sqsubseteq \mathcal{G}} \Pr(G_i) |c(CC^{i,j} | v \in CC^{i,j})|$. Then the total coding length is

$$\begin{aligned}\mathcal{L}_2 &= \sum_{v \in \mathcal{V}} \sum_{G_i \sqsubseteq \mathcal{G}} \Pr(G_i) |c(CC^{i,j} | v \in CC^{i,j})| \\ &= -\sum_{G_i \sqsubseteq \mathcal{G}} \Pr(G_i) |\mathcal{V}| \sum_{j=1}^{L_i} \frac{|CC^{i,j}|}{|\mathcal{V}|} \log \frac{|CC^{i,j}|}{|\mathcal{V}|}.\end{aligned}$$

Note that the above coding scheme does not consider the rows of the auxiliary cluster table.

Instead, if we code component identifiers with the information of vertex cluster label information, we could code the components row by row for each auxiliary table $\mathcal{T}_i$. Then, similarly, the total coding length of component identifier with cluster information is calculated as

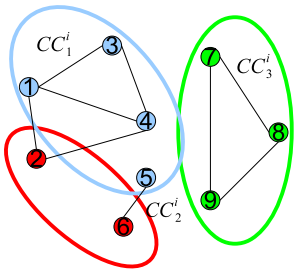
$$\mathcal{L}_3 = -\sum_{G_i \sqsubseteq \mathcal{G}} \Pr(G_i) \sum_{k=1}^{K} |C_k| H(\bigcup_j CC_k^{i,j}).$$

Obviously, $\mathcal{L}_3$ equals our objective function $\mathcal{F}$, that is, Equation (3). In our example for sampled graph G_i, the connected component information can be represented with $4 \times (-\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4}) + 2 \times (-\frac{1}{2}\log\frac{1}{2} \times 2) + 3 \times (-\log 1 + 1) \approx 9$ bits.

Interestingly, based on the above analysis, we find the underlying meanings of Observation 1 from coding perspective. That is, we have that

$$\mathcal{F}_p + \mathcal{L}_2 = \mathcal{F} + \mathcal{F}_e.$$

Therefore, we can see that to do data coding we have two different angles: row ($\mathcal{F}$ and $\mathcal{F}_e$) or column ($\mathcal{F}_p$ and $\mathcal{L}_2$) of auxiliary tables.



(a) Sampled Graph G_i

	$CC^{i,1}$	$CC^{i,2}$	$CC^{i,3}$
C_1^i:	{1,2,3,4}	{5,6}	{7,8,9}

	$CC^{i,1}$	$CC^{i,2}$	$CC^{i,3}$
C_1^i:	{1,3,4}	{5}	ø
C_2^i:	{2}	{6}	ø
C_3^i:	ø	ø	{7,8,9}

(b) 1 and 3 Cluster Table for G_i

	$CC^{i,1}$	$CC^{i,2}$	$CC^{i,3}$	
C_1^i:	1	01	00	$Cost=14$

	$CC^{i,1}$	$CC^{i,2}$	$CC^{i,3}$	
C_1^i:	0	1		
C_2^i:	0	1		$Cost=9$
C_3^i:			0	

(c) Huffman Codes

Figure 3: An Running Example

As we will see in the next subsection, the view provided by the auxiliary table $\mathcal{T}_i$ view is very useful from the perspective of interpretability. In addition, it provides us with the mechanisms required to design efficient clustering algorithms.

B. Cluster Properties

In this section, we will show that the minimization of Equation (9), results in several interesting properties of the clustering, which further helps in algorithm design.

Lemma 2: The cost $\mathcal{F}(G_i)$ for each individual sampled graph G_i can be minimized only if for every connected component $CC^{i,j}$ there exists a k, such that $CC^{i,j} \subseteq C_k$.

Proof Sketch: Suppose initially $CC^{i,j}$ is divided into K parts, as $CC_1^{i,j}, \cdots, CC_K^{i,j}$, and we have the previous code word $c(CC_k^{i,j})$ for each cell in $\mathcal{T}_i$. Now there exists one cell $\mathcal{T}_i(k,j)$ of $\mathcal{T}_i$, such that $|c(CC_k^{i,j})| \leq |c(CC_l^{i,j})|$ for $1 \leq l \neq k \leq K$. Therefore, by moving $CC^{i,j}$ entirely into the kth cluster, $\mathcal{F}(G_i)$ can be reduced. Then by recalculating, $\mathcal{F}(G_i)$ will be further reduced. Hence, putting one component into each cluster could always reduce $\mathcal{F}(G_i)$, which completes our proof. □

Lemma 2 implies that when each individual possible world (sampled graph) is considered, our objective function tends to group vertices in the same component together. Therefore, our objective function naturally captures the modularity (fragments) of each possible world.

Based on on Lemma 2, we have the following corollary.

Corollary 1: If vertex set V_s is always contained in one fragment in all sampled graphs, there exists a k, such that, $V_s \subseteq C_k$.

Corollary 1 has an interesting connection with reliability. If $R(V_s) = 1$, then all vertices of V_s should belong to the same cluster. This extreme case illustrates the connections between clustering and reliability. Of course, in practice, the precise cluster membership of vertices in V_s is much more complicated, and involves a tradeoff between $|V_s|$ and $R(V_s)$. We also note that the minimization of Equation (9) guarantees that there are no empty clusters. This is implied by the following lemma.

Lemma 3: The optimal value $\mathcal{F}$ for K clusters is smaller than that for $K-1$ clusters.

Proof Sketch: Given the optimal clustering for $K-1$ clusters and the corresponding coding for each $\mathcal{T}_i$, we can move one vertex v from C_k, where $1 \leq k \leq K-1$ to a new cluster C_K. Obviously, $\mathcal{F}_{K-1}$ is reduced to $\mathcal{F}_K^{'}$. Therefore, the solution with $K-1$ clusters is always sub-optimal. □

Finally, we want to provide an example to illustrate that the nodes within a cluster, which are obtained through minimization of Equation (9) may not necessarily connected only through nodes within the cluster, but may use nodes from outside the cluster. For example, suppose $\mathcal{G}$ consists of several highly connected components, which form a star structure, and given $K = 2$, the clusters failing to contain the center components will be disconnected.

IV. Uncertain Graph Clustering Algorithm

In this section, we will first develop an efficient partition-based algorithm to minimize $\mathcal{F}$. However, since the result will not guarantee cluster connectivity, we extend this algorithm by a greedy procedure to add connectivity.

A. Coded k-means Algorithm

Different samples generating the auxiliary tables $\mathcal{T}_i$ may contain different number of components L_i. However, the number of rows K is constant through all the N tables. Therefore, if we fix the coding for each row in each sampled graph, each vertex v could always choose to move to the row which reduces its coding cost. The minimization of $\mathcal{F}$ for N sampled graphs is more complicated, though we can still develop a two stage procedure. More specifically, we initially assign each vertex randomly to one of the K clusters, and repeat the following two stages until $\mathcal{F}$ converges.

Coding Computation: Given the current vertex assignment C_k, and corresponding sampled components $CC_k^{i,j}$, where $1 \leq k \leq K, 1 \leq j \leq L_i$, then the coding for each row of $\mathcal{T}_i$ is computed using Huffman coding. This corresponds to the code $c(\mathcal{T}_i(k,j))$ $(c(CC_k^{i,j}))$ for the cell in kth row and jth column for each table $\mathcal{T}_i$. While it is possible to use other coding methods, we use *Huffman coding* for the purpose of the paper. A coding example is illustrated in Figure 3(c).

Vertex Assignment: Given the current row coding $c(\mathcal{T}_i(k,j))$ for each table $\mathcal{T}_i$, where $1 \leq k \leq K$, $1 \leq j \leq L_i$ and $1 \leq i \leq N$, we can assign each vertex v to one cluster which could minimize the total coding cost for v through all the sampled graphs. Therefore, each vertex v is assigned to cluster k^*, such that

$$k^* = \arg\min_k \sum_{i=1}^{N} |c(\mathcal{T}_i(k, f(v,i)))|,$$

where $f(v,i)$ represents the connected component id v belongs to in sampled graph G_i.

Intuitively, through the above iterative process, $\mathcal{F}$ could find its (local) optimal value, as will be proved later in this section. The detailed algorithm is depicted in Algorithm 1.

Algorithm 1 Coded k-means($\{CC^{i,j}, 1 \leq j \leq L_i, 1 \leq i \leq N\}$, K)

Require: $\{CC^{i,j}, 1 \leq j \leq L_i, 1 \leq i \leq N\}$: fragments;
Require: K: cluster number;
1: $C_k = \emptyset$, $1 \leq k \leq K$;
2: **for all** $v \in \mathcal{V}$ **do**
3: $C_k = C_k \cup \{v\}$, where k is randomly chosen from $\{1, \cdots, K\}$;
4: **end for**
5: **while** $\mathcal{F}$ hasn't converge **do**
6: **Stage** 1**:**
7: **for all** $1 \leq i \leq N$ **do**
8: **for all** $1 \leq k \leq K$ **do**
9: $\{c(CC_k^{i,j}), 1 \leq j \leq L_i\}$ = **HuffC**($\{CC_k^{i,l}, 1 \leq l \leq L_i\}$);
10: **end for**
11: **end for**
12: **Stage** 2**:**
13: $C_k = \emptyset$, $1 \leq k \leq K$;
14: **for all** $v \in \mathcal{V}$ **do**
15: **for all** $1 \leq i \leq N$ **do**
16: **for all** $1 \leq k \leq K$ **do**
17: $CodeLength_k(v) += |c(CC_k^{i,f(v,i)})|$;
18: **end for**
19: **end for**
20: $C_{k^*} = C_{k^*} \cup \{v\}$, where $k^* = \arg\min_{1 \leq k \leq K}\{CodeLength_k(v)\}$;
21: **end for**
22: **end while**
23: **return**$\{C_1, 1 \leq k \leq K\}$;

In Algorithm 1, from Line 1 to Line 4 we assign a vertex v to a random cluster label C_k, where $1 \leq k \leq K$. After that, we enter a two-stage iterative process (Line 5 to Line 22). In the first stage (Line 6 to Line 11), we utilize Huffman coding algorithm to generate code words $c(CC_k^{i,j})$ for every subcomponent $CC_k^{i,j}$ according to the vertex number distribution on $\{CC_k^{i,j}, 1 \leq j \leq L_i\}$. Note that each component $CC^{i,j}$ has K Huffman code words corresponding to its K subcomponents. In Line 9, $c(CC_k^{i,j})$ stands for Huffman code for kth subcomponent of $CC^{i,j}$. After calculating the current coding, in the second stage (Line 12 to Line 21), we sum up the coding length $CodeLength_k(v)$ when v is encoded by code words in the kth cluster. Then, we assign v to the cluster C_{k^*} that has the minimum code length for v (Line 20). In order to define the convergence condition (Line 5), we could either set a maximum iteration number or set a maximum objective function difference between consecutive iterations.

Time Complexity In the first stage, we calculate the Huffman code for the cells in all auxiliary tables $\mathcal{T}_i$. The best algorithm to code one row of $\mathcal{T}_i$ by Huffman code requires $O(L_i \log L_i)$ time complexity. Therefore, the time complexity for the first step is $O(NK\overline{L} \log \overline{L})$, where L is the average component number for sampled graphs. In the second step, we have calculate the sum of code word length for each vertex v with different colors; in the worst case the time complexity is $O(|\mathcal{V}|NK)$. Given $|\mathcal{V}| \gg \overline{L} \log \overline{L}$, the entire complexity of our algorithm is $O(l|\mathcal{V}|NK)$, where l is the maximum iteration number.

As mentioned before, Algorithm 1 always converges to (local) optimal value, as can be formally depicted below.

Lemma 4: $\mathcal{F}$ always converges to (local) optimal value through Algorithm 1.

Proof Sketch: After the initial assignment, Algorithm 1 enters into the iterative process. Suppose, $\mathcal{F}$ is finite.

Stage 1: Let us look at the case for sampled graph G_i and cluster k. Suppose we already have existing coding $c(\overline{CC}_k^{i,j})$ for each set $\overline{CC}_k^{i,j}$. After the assignment (**Step** 2), the current vertex set for each set becomes $CC_k^{i,j}$. Then, we have $\sum_{j=1}^{L_i} |CC_k^{i,j}| * |c(\overline{CC}_k^{i,j})| \geq \sum_{j=1}^{L_i} |CC_k^{i,j}| * |c(CC_k^{i,j})|$, where $c(CC_k^{i,j}), 1 \leq j \leq L_i$ is the code word calculated using current vertex distribution $CC_k^{i,j}$. The inequality holds based on the optimality of Huffman coding. Because the code words for different cluster C_k and different sampled graph G_i are independent, we have $\sum_{i=1}^{N} \sum_{k=1}^{K} |C_k| H(C_k^{i,j}, 1 \leq j \leq L_i)$ is also minimized. Therefore, Equation (9) is reduced in this step.

Stage 2: In this step each vertex v is reassigned to a cluster C_{k^*} such that with cluster k^*, the total coding length for v, that is, $\sum_{i=1}^{N} |c(CC_{k^*}^{i,f(v,i)})|$ is smaller than or equal to the length with other colors. That means, $\sum_{i=1}^{N} |c(CC_{k^*}^{i,f(v,i)})| \leq \sum_{i=1}^{N} |c(CC_k)^{i,f(v,i)}|$, where C_k is the cluster v belongs to, in the previous step. Then, we have:

$$\sum_{v \in \mathcal{V}} \sum_{i=1}^{N} |c(CC_{k^*}^{i,f(v,i)})|$$
$$\leq \sum_{v \in \mathcal{V}} \sum_{i=1}^{N} |c(CC_k)^{i,f(v,i)}| = N\mathcal{F}.$$

Therefore, we can see that after vertex reassignment, the value of $\mathcal{F}$ further decreases. Thus, $\mathcal{F}$ is always reduced through the iterative process, which completes our proof. □

B. Connectivity Constraint

Note that through Algorithm 1, our clustering criterion (Equation (9)) does not enforce each cluster to be a connected subgraph in uncertain graph $\mathcal{G}$, for the non-connectivity is reasonable in uncertain scenario.

We design an approach which enforces connectivity with methods, that utilize the clustering result from Algorithm 1. More formally, the following steps are used in order to force connectivity.

1) *With the output of Algorithm 1, $\{C_1, \cdots, C_K\}$, we do DFS or BFS to check the connectivity of each cluster C_i. If C_i is not connected, we can split C_i into several connected new clusters. In this way, we obtain a new clustering $\{C_1^{'}, \cdots, C_L^{'}\}$, where C_i is connected for $1 \leq i \leq L$ and $L \geq K$.*
2) *If $L = K$, the program stops; otherwise ($L > K$), we merge the neighboring clusters, which could minimally increase $\mathcal{F}$. More formally, we choose to merge $C_{s^*} = C_{s^*} \cup C_{t^*}$, where $s^* \neq t^*$ and there exits some edges $e = (u, v)$ and $u \in C_{s^*}$ and $v \in C_{t^*}$, such that suppose $f_i = (|C_s| + |C_t|)H(C_s^{i,j} \cup C_t^{i,j}) - |C_s|H(C_s^{i,j}) - |C_t|H(C_t^{i,j})$ then we have*

$$(s^*, t^*) = \arg \min_{1 \leq s,t \leq L} \sum_{i=1}^{N} f_i.$$

3) *$L = L - 1$; go to 2).*

Using the above procedure, we greedily merge two clusters which minimally increase our objective function based on the result from Algorithm 1. Therefore, the time complexity of the above procedure can be estimated as $O(N\overline{L} \log \overline{L})$ given $L, K \ll N$, where $\overline{L}$ is the average number of connected components for all sampled graphs.

V. Experimental Study

In this section, we present experimental results studying the effectiveness and efficiency of our method. We tested for the following measures:

1. Accuracy: We tested cluster purity and balance in the network size. We tested both variations of our approach, which corresponds to the coded k-means method, with or without the greedy enhancement.

2. Efficiency: We tested the efficiency of the methods in terms of running time.

The experiments were conducted on a 2.0GHz Dual Core AMD Opteron CUP with 4.0GB RAM running Red Hat Linux. All algorithms were implemented in C++ and the Standard Template Library (STL) was used.

Table I: Data Set Summary

	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	AVG(P)
DBLP	326186	1432920	0.22821
PPI	513	1618	0.44
KARATE	34	78	0.38
BOOK	105	441	0.41

A. Experiment Setup

In this subsection, we report our experiment setup.

Datasets: We used four different real data sets as follows:

DBLP: This network is created by authors in [24] based on a snapshot of the DBLP database. When two authors coauthored a paper before 2001, one edge is added between them. The edge probability is generated based on an exponential cdf of mean 2 to the number of coauthored papers.

PPI: This dataset is obtained by integrating the BioGRID database with the STRING database. Based on BioGRID database, a A. thaliana (thale cress) protein-protein interaction network is created and the labels of vertices are the COG functional annotations of proteins and edge probabilities are obtained from the STRING database. This data set has been used in [34].

Karate: This data set is a friendship network between 34 members of a karate club at a US university, and used to study the information flow among small groups of people [31]. The edge probability is randomly generated from a uniform distribution in the interval $(0, 1]$.

Politics-Books: The nodes in this data set[1] are books about US politics sold by *Amazon.com*. Two books are linked when they are bought by the same people. We assume that people are more likely to buy the books from the same category, such as "liberal", "neutral", or "conservative". Therefore, the probability between books from same category is uniformly generated in interval $(0.5, 1]$. Otherwise, the probability is generated in the interval $(0, 0.5]$.

The data set information is summarized in Table I.

Baselines and Parameter Settings:

The accuracy was tested with the use of three other methods: *spectral clustering, ensemble clustering[27] and MCL[29].* These methods are often used for deterministic graphs. Therefore, in order to use them, we transfer the uncertain graphs into weighted graphs, where the uncertainty probabilities are edge weights. To apply spectral and MCL methods, we directly cast the edge probabilities as edge weights. Meanwhile, the input for ensemble clustering methods are the results by running spectral clusterings 5 times. The source codes for the three benchmark methods can be obtained online[1], [2], [3]. As suggested by Lemma 1, the accuracy is greatly influenced by the sample size. In the following experiments, we set $\alpha = 0.05$ and $\delta = 0.02$, and the corresponding sample size $N \approx 5600$.

[1] http://www.orgnet.com/

Table II: ρ Comparison

	Kmeans	Greedy	MCL	Spectral	Ensemble
BOOK	1.01	1.01	1.01	1.01	1.01
KARATE	1.00	1.00	1.02	1.01	1.00
PPI	1.35	1.35	2.63	3.66	3.34
sDBLP	1.58	1.59	4.51	5.32	5.15

B. Effectiveness Results

In this subsection, we report two groups of effectiveness experiment results, based on generalized reliability (our objective function) and standard reliability respectively. The reason we use generalized reliability is that it essentially captures our two intuitions; meanwhile, standard reliability provides us with a more objective measurement for the results from different clustering methods.

Effectiveness on Generalized Reliability: Because Equation (3) essentially captures our intuitions (purity and balance), then it is a reasonable measurement for us to compare the clustering results for our methods and three baseline methods. To compare the accuracy through different data sets, we use the average coding length ρ per vertex, that is

$$\rho = \frac{\mathcal{F}}{|\mathcal{V}|},$$

and we set cluster number K equal to 3. The results are presented in Table II.

Note that sDBLP is an extracted subgraph of DBLP with 1000 vertices. From Table II, we can see that Spectral clustering method works worse than Ensemble clustering. This is reasonable because the latter congregates 5 results of the former one. Then, MCL works a litttle bit better than the previous two methods (Spectral and Ensemble) by reducing the average coding length ρ by 10% to 20%. Our coded k-means algorithm further reduces ρ dramatically, that is more than 50% from that of MCL. From this experiment, we can see that our methods are much more accurate than other techniques.

Effectiveness on Standard Reliability: To be more objective, here we use standard reliability to measure the accuracy of the results from different methods. Hence we developped two kinds of criteria based on standard reliability concept to measure clustering result: average vertex pairwise reliability (AVPR), denoted by θ and the average cluster reliability (ACR) denoted by β. More specifically, given a clustering $\mathcal{C} = \{C_1, \cdots, C_K\}$ of uncertain graph $\mathcal{G}$, AVPR

$$\theta(\mathcal{C}) = \frac{2\sum_{k=1}^{K}\sum_{u,v\in C_k} R(\{u,v\})}{\sum_{k=1}^{K}|C_k|(|C_k|-1)},$$

and ACR

$$\beta(\mathcal{C}) = \frac{\sum_{k=1}^{K}|C_k|R(C_k)}{|V|}.$$

We can see that both critera measures the average reliability of the entire clustering. We can see that θ emphasizes the pairwise reliability, and that β puts a stronger constraint that requires all vertices in one cluster should be connected simultaneously. Therefore, θ and β provide us a way to measure the quality of clustering results in different applications.

Table III: AVPR(θ) Comparison

	Kmeans	Greedy	MCL	Spectral	Ensemble
BOOK	0.8834	0.8834	0.8399	0.8614	0.6018
KARATE	0.9556	0.9556	0.9434	0.9456	0.9567
PPI	0.7099	0.7099	0.1009	0.2124	0.3776
sDBLP	0.3454	0.3454	0.04	0.063	0.045

Table IV: ACR(β) Comparison

	Kmeans	Greedy	MCL	Spectral	Ensemble
BOOK	0.0576	0.0576	0.0212	0.0383	0.0265
KARATE	0.4494	0.4494	0.3430	0.7558	0.8003
PPI	0.4299	0.4299	0.0081	0.0038	0.0010
sDBLP	0.2265	0.2265	0.0002	0.0008	0.0001

In the first groups of experiments, we set cluster number K equal to 4, and we compared with their AVPR and report our result in Table III.

From Table III we can see that in terms of methods, MCL works worst among all the methods. Spectral clustering method and Ensemble method work comparatively. The AVPR for clusterings of our k-means and Greedy algorithm are almost the same. In terms of datasets, for small graphs (BOOK and KARATE), all the methods produce similar AVPR. However, for large graphs (PPI and sDBLP), the AVPR of our methods are at least 7 times that of others methods.

In our second group of experiments, we also set cluster number K equal to 4. We compared the ACR for all the methods and the result is reported in Table IV.

From Table IV, similar as the the first groups of experiment, we can see MCL produces the clustering with the lowest ACR; and Spectral clustering and Ensemble clustering produce comparative results, and our methods produce the greatest ACR for all datasets. In term of datasets, for small networks (BOOK and KARATE), all methods produced comparative ACR for all datasets. However, for large networks (PPI and sDBLP), our methods (k-means and Greedy) generate clustering with ACR at least 50 times better than that of other methods.

In sum, our coded k-means algorithm with or without greedy enhancement methods consistently produce much better clusers than other methods (MCL, Spectral and Ensemble) on both our generalized and standard reliability criteria.

C. Experimental Results for Efficiency

In this experiment, we study the scalability of all the methods. To do this, we extract 5 subgraphs from the DBLP dataset, respectively with vertices from $20k$ to $100k$. Here two baseline methods (Spectral and Ensemble) can not scale up to graph with 20k vertices. Therefore, in this experiment

we only care about the efficiency of our methods and MCL. The result is reported in Figure 4.

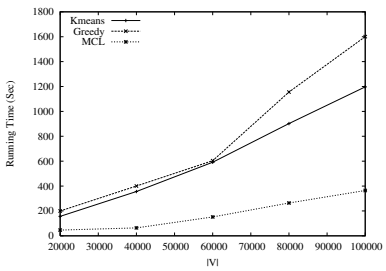


Figure 4: Efficiency Result

From the result we can see that MCL normally take approximately 27% time of that of coded k-means algorithm. Meanwhile, we can see that the running time of the coded k-means algorithm is linearly related to $|V|$. This is in agreement with our complexity analysis in Section IV. Meanwhile, by adding greedy merge step, our Greedy algorithm only takes more than about 20% of that of our coded k-means algorithms.

VI. Related Work

The work most relevant to ours [20] is that of clustering on uncertain graphs. Kollios *et al.* in [20] studied this problem from the perspective of *possible world semantics*. To the best of our knowledge, this is the only uncertain graph clustering work so far. In [20] Kollios *et al.* discover the clusters by minimizing the expected edit-distance $D(\mathcal{G}, Q)$ between uncertain graph $\mathcal{G}$ and cluster graph Q. Note that the (deterministic) cluster graph Q explicitly requires that each cluster be a clique. Their algorithm avoids enumeration of possible worlds by emphasizing vertex pairwise connectivity; however, their algorithm inevitably favors smaller clusters over large ones by using the clique concept. For example, in Figure 5, the uncertain graph $\mathcal{G}$ contains a subgraph G with three vertices a, b and c. Obviously, G is well separated from $\mathcal{G} \setminus G$ and forms a natural cluster. However, using the goodness function $D(G, Q)$ in [20], the clustering $\{\{b\}, \{a, c\}\}$ is preferred.

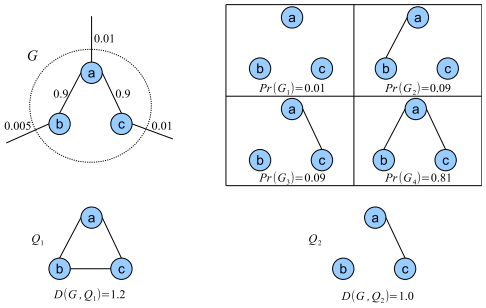


Figure 5: Counter intuitive example

Our generalized reliability criterion can also be used to solve the ensemble clustering (clustering aggregation, correlation clustering) problem [27], [28], [11], [22]. Algorithms for ensemble clustering normally discover the median partitions π which gain the most agreement from N different partitions $\pi_1, \cdots, \pi_N$. Our partition-based method can be employed to tackle these problems. However, unlike existing work, our method assumes that the cluster label for each vertex is consistent through all the partitions, and try to maximizes the purity for each cluster of each π_i.

Substantial work has also been done in uncertain graph analytics. In [10], it has been shown that Monte Carlo sampling can be combined with hypothesis testing in order to compute the shortest paths given that the edge weights are random variables. Zou *et al.* discover efficiently subgraph patterns in [34], [32] from uncertain graph databases and Top-k maximal cliques in [33] in an uncertain graph. Later in [23], Papapetrou *et al.* propose to use indexing techniques to speed up frequent subgraph mining in uncertain graph databases. Potamias [24] employ the sampling method to answer the k-nearest neighbor query in uncertain graphs. Jin *et al.* in [18] estimate the reachability of vertex pairs within certain distance with statistical estimators. In [30], Yuan *et al.* develop efficient method to answer the threshold-based subgraph query over large uncertain graphs. Hua [16] shows how to find the shortest weighted paths most likely to complete within a certain time constraint. Hintsanen *et al* [12], [13], [14] develop efficient algorithms to find the most reliable subgraphs in uncertain graphs. Jin *et al* [17] develop fast peeling algorithms to find highly reliable subgraphs.

VII. Conclusion

In this paper, we presented a method for uncertain graph clustering which is based on the notion of generalized reliability. This notion is inherently more effective than traditional methods for incorporating uncertainty in the clustering process. The approach uses coding methods from information theory for intermediate representation of the meta-structures in the clustering process. We present experimental results which show the advantages of our approach.

Acknowledgment

Ruoming Jin, Lin Liu and Yelong Shen were partially supported by the National Science Foundation, under CAREER grant IIS-0953950. Research of Charu Aggarwal was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NSF, the Army Research Laboratory, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] http://www.stat.washington.edu/spectral.

[2] http://www.lans.ece.utexas.edu/~strehl/soft.html.

[3] http://www.micans.org/mcl.

[4] C. C. Aggarwal, editor. *Managing and Mining Uncertain Data*. Advances in Database Systems. Springer, 2009.

[5] J. S. Bader, A. Chaudhuri, J. M. Rothberg, and J. Chant. Gaining confidence in high-throughput protein interaction networks. *Nature Biotechnology*, 22(1):78–85, December 2003.

[6] S. Chen, , A. Borthwick, and V. R. Carvalho. The case for cost-sensitive and easy-to-interpret models in industrial record linkage. In *VLDB, QDB workshop*, 2011.

[7] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Stat.*, 23:493–507, 1952.

[8] C. J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, Inc., 1987.

[9] C. M. Deane, Ł. Salwiński, I. Xenarios, and D. Eisenberg. Protein interactions: two methods for assessment of the reliability of high throughput observations. *Molecular & cellular proteomics : MCP*, 1(5):349–356, 2002.

[10] H. Frank. Shortest paths in probabilistic graphs. *Operations Research*, 17(4):pp. 583–599, 1969.

[11] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Trans. Knowl. Discov. Data*, 1(1), Mar. 2007.

[12] P. Hintsanen. The most reliable subgraph problem. In *PKDD*, pages 471–478, 2007.

[13] P. Hintsanen and H. Toivonen. Finding reliable subgraphs from large probabilistic graphs. *Data Min. Knowl. Discov.*, 17(1):3–23, 2008.

[14] P. Hintsanen, H. Toivonen, and P. Sevon. Fast discovery of reliable subnetworks. In *ASONAM*, pages 104–111, 2010.

[15] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[16] M. Hua and J. Pei. Probabilistic path queries in road networks: traffic uncertainty aware path selection. In *Proceedings of the 13th International Conference on Extending Database Technology*, EDBT '10, pages 347–358. ACM, 2010.

[17] R. Jin, L. Liu, and C. C. Aggarwal. Discovering highly reliable subgraphs in uncertain graphs. In *KDD'11*, pages 992–1000, 2011.

[18] R. Jin, L. Liu, B. Ding, and H. Wang. Distance-constraint reachability computation in uncertain graphs. In *Proceedings of the VLDB Endowment*, volume 4, 2011.

[19] H. Kawahigashi, Y. Terashima, N. Miyauchi, and T. Nakakawaji. Modeling ad hoc sensor networks using random graph theory. In *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*, pages 104 – 109, jan. 2005.

[20] G. Kollios, M. Potamias, and E. Terzi. Clustering large probabilistic graphs. *TKDE*, 99, 2011.

[21] U. Kuter and J. Golbeck. Sunny: a new algorithm for trust inference in social networks using probabilistic confidence models. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, pages 1377–1382.

[22] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Machine Learning*, 52(1):91–118, July 2003.

[23] O. Papapetrou, E. Ioannou, and D. Skoutas. Efficient discovery of frequent subgraph patterns in uncertain graph databases. In *EDBT/ICDT'11*, pages 355–366, 2011.

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

[25] D. R. Rhodes, S. A. Tomlins, S. Varambally, V. Mahavisno, T. Barrette, S. Kalyana-Sundaram, D. Ghosh, A. Pandey, and A. M. Chinnaiyan. Probabilistic model of the human protein-protein interaction network. *Nat Biotech*, 23(8):951–959, Aug. 2005.

[26] S. Schaeffer. Graph clustering. *Computer Science Review*, (1):27–64, Aug. 2007.

[27] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, Mar. 2003.

[28] A. Topchy, A. K. Jain, and W. Punch. Clustering ensembles: models of consensus and weak partitions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1866–1881, Oct. 2005.

[29] S. Van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.

[30] Y. Yuan, L. Chen, and G. Wang. Efficiently answering probability threshold-based shortest path queries over uncertain graphs. In *DASFAA*, pages 155–170, 2010.

[31] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.

[32] Z. Zou, H. Gao, and J. Li. Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics. In *KDD*, pages 633–642, 2010.

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

[34] Z. Zou, J. Li, H. Gao, and S. Zhang. Mining frequent subgraph patterns from uncertain graph data. *TKDE*, 22(9), 2010.