Uncertain Graph Processing through Representative Instances and Sparsification

Panos Parchas
supervised by Dimitris Papadias
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
pparchas@cs.ust.hk

ABSTRACT

Data in several applications can be represented as an uncertain graph, whose edges are labeled with a probability of existence. Currently, most query and mining tasks on uncertain graphs are based on Monte-Carlo sampling, which is rather time consuming for the large uncertain graphs commonly found in practice (e.g., social networks). To overcome the high cost, in this doctoral work we propose two approaches. The first extracts deterministic representative instances that capture structural properties of the uncertain graph. The query and mining tasks can then be efficiently processed using deterministic algorithms on these representatives. The second approach sparsifies the uncertain graph (i.e., reduces the number of its edges) and redistributes its probabilities, minimizing the information loss. Then, Monte-Carlo sampling applied to the reduced graph becomes much more efficient.

1. INTRODUCTION

Graphs constitute an expressive data representation paradigm used to describe entities (vertices) and their relationships (edges) in a wide range of applications. Sometimes the existence of the relationship between two entities is uncertain due to noisy measurements, inference and prediction models, or explicit manipulation. For instance, in biological networks, vertices represent genes and proteins, while edges correspond to *interactions* among them. Since these interactions are observed through noisy and error-prone experiments, each edge is associated with an uncertainty value [2]. In large social networks, uncertainty arises for various reasons; the edge probability may denote the accuracy of a *link prediction*, or the influence of one person on another, e.g., in *viral marketing* [10]. Uncertainty can also be injected intentionally for obfuscating the identity of users for privacy reasons [5].

In all these applications the data can be modeled as an *uncertain graph*, whose edges are labeled with a probability of existence. This probability represents the confidence that the relation corresponding to the edge holds in reality. More formally let $\mathcal{G} = (V, E, p)$ be an uncertain graph, where function $p : E \rightarrow (0, 1]$ assigns a probability of existence to each edge. Following the literature, we consider the edge probabilities independent [18, 8], and we assume *possible-world* semantics [1]. Specifically, the possible-world semantics interprets $\mathcal{G}$ as a set $\{G = (V, E_G)\}_{E_G \subseteq E}$ of $2^{|E|}$ possible deterministic graphs, each defined by a subset of E.

As one cannot afford to materialize $2^{|E|}$ possible worlds, a common query processing solution is based on Monte-Carlo sampling, i.e., assess the query on a subset of randomly selected possible worlds. However, sampling is not always a viable option for large graphs because: i) sampling a possible world has a non-negligible cost as it requires generating a random number for each edge $e \in E$, and ii) processing on every sample may be extremely expensive, especially for large graphs. For instance, betweenness centrality, a measure of vertex importance in the graph, involves all-pairs-shortest-path computations, which cannot be performed many times (i.e., for each sample) in any graph of moderate size.

Motivated by the above, we propose two approaches that facilitate query processing and data mining in uncertain graphs. The first aims at removing the uncertainty by producing representative instances of uncertain graphs. Queries can then be processed efficiently on the deterministic instance using conventional graph algorithms. The second approach generates another uncertain graph with fewer edges, but similar properties in which Monte-Carlo sampling is more efficient. To the best of our knowledge there has not been previous work on either of the proposed directions.

In order to achieve accuracy, the representative and sparsified graphs should preserve the expected (underlying) structure of the original uncertain graph. Starting form the observation that the vertex degree is one of the most fundamental properties of the structure of a graph [12, 13], we conjecture that by preserving the expected degree of each vertex, we capture the essence of the underlying uncertain graph, and thus accurately approximate other properties. Additionally, we investigate more general structural properties, focusing on the expected behavior of *groups* of vertices (e.g. triples).

Regarding the first direction, we propose three methods for generating representative instances: Average Degree Rewiring (ADR), Approximate B-matching (ABM) and GAME. ADR involves two phases: first, it generates an instance with the same average vertex degree as the uncertain graph; then, it randomly rewires edges if they lead to better approximation of the vertex degrees. ABM applies b-matching [7] to obtain an initial instance, which then improves using weighted maximum bipartite matching. Finally, GAME applies *best response dynamics* [14] to extract representatives that preserve the expected structure for groups of vertices.

Regarding the second direction, we propose Backbone Linear Programming (BLP) algorithm, for sparsifying uncertain graphs. BLP involves two steps. The first step generates a backbone graph with the required number of edges. Then, a second step applies Linear Programming to assign probabilities on the edges of the backbone graph, minimizing the discrepancy of the expected degrees

Proceedings of the VLDB 2015 PhD Workshop

between the original and the reduced graph.

Summarizing, our contributions are:

- We propose two novel frameworks for querying uncertain graphs, based on the extraction of representatives, and on uncertain graph sparsification. Both frameworks simplify the input uncertain graph in order to achieve efficiency without sacrificing accuracy.
- We investigate the relevant properties of uncertain graphs that should be preserved by the representative instances and the sparsified graphs.
- We present ADR , ABM and GAME, for extracting representatives, and BLP for uncertain graph sparsification. These methods are applicable to massive uncertain graphs of millions of vertices and edges.

The rest of our work is organized as follows. Section 2 briefly describes the related work. Section 3 presents our contributions. Finally, Section 4 contains directions for future research.

2. RELATED WORK

Processing on uncertain graphs can be classified into three main approaches: i) queries on shortest-path distances and reliability, ii) pattern mining and graph decomposition, and iii) subgraph search.

Towards the first direction, Jin *et al.* [9] introduce the *distance-constrained reachability* query, which, given two vertices *s* and *t*, and a threshold *d*, returns the probability that the distance from *s* to *t* is less than *d*. The authors propose two estimators for the distance-constrained reachability query that have provably less variance than naïve Monte Carlo methods. Potamias *et al.* [18] redefine traditional nearest neighbor queries by using statistical distance metrics (e.g. majority, median). These metrics are computed by applying Dijkstra's algorithm on possible worlds that are sampled on the fly.

In the second line of research, Zou *et al.* [20] investigate top-*k* maximal cliques in uncertain graphs. Moustafa *et al.* [15] propose efficient algorithms for subgraph pattern matching for graphs, where in addition to edges, vertices are also uncertain. In the third direction of research, Yuan *et al.* [19] propose a feature-based framework for subgraph search. In a rather different type of research Boldi *et al.* [5], intentionally inject uncertainty in a social graph in order to obfuscate the identity of its users. Finally, in a recent work, Li *et al.* [11] improve the naïve Monte Carlo by performing a smarter sampling that has provably smaller variance.

To the best of our knowledge there is no previous work on either extraction of deterministic representatives, or sparsification of uncertain graphs. Concerning sparsification in *deterministic* graphs, the related work can be classified into *sparsifiers* and *spanners*. *Sparsifiers* [6] aim at reducing the size of the graph, mainly for storage purposes, while maintaining properties such as the size of the cuts, or the energy in the electric equivalent circuits [4]. *Spanners*, aim at reducing the graph size for query efficiency, while maintaining the shortest path distances among the nodes [3].

3. CONTRIBUTIONS

Section 3.1 formally defines the problems investigated in the thesis, and Section 3.2 presents the proposed algorithms.

3.1 Problem Definitions

Given an uncertain graph $\mathcal{G} = (V, E, p)$ and a vertex $u \in V$, the *expected degree* of u in $\mathcal{G}$ is the summation of the probabilities of u's adjacent edges:

$$[deg(u, \mathcal{G})] = \sum_{e=(u,v)\in E} p_e$$

When the uncertain graph is implied, we write for convenience $[deg_u]$. Let G be a deterministic instance of $\mathcal{G}$. We define the *discrepancy* $dis(u, G)$ of a vertex u in $G \sqsubseteq \mathcal{G}$ as the difference of $u's$ degree in G to its expected degree, i.e., $dis(u, G) = deg(u, G) - [deg_u]$. Given the individual vertex discrepancies, we define the overall discrepancy Δ of a possible graph G as follows:

DEFINITION 1. *Given an uncertain graph $\mathcal{G} = (V, E, p)$, the* discrepancy *of any possible graph $G \sqsubseteq \mathcal{G}$ is defined as*

$$\Delta(G) = \sum_{u\in V} |dis(u, G)| \tag{1}$$

The first problem we tackle is the following:

PROBLEM 1 (*degree*-REPRESENTATIVE INSTANCE). *Given an uncertain graph $\mathcal{G} = (V, E, p)$, find a possible graph $G^* \sqsubseteq \mathcal{G}$ such that:*

$$G^* = \arg\min_{G\sqsubseteq\mathcal{G}} \Delta(G). \quad \square$$

Going a step further, we generalize the notion of vertex degree to the concept of *n-clique cardinality* of a vertex u, i.e., the number of cliques of size n that contain u. In particular, the degree of a vertex u is equivalent to its 2-clique cardinality (i.e., the number of cliques of size 2 that contain u), whereas u's triangle connectivity corresponds to its 3-clique cardinality. The notion of n-clique is extended naturally to $n > 3$. Intuitively, this generalization aims at capturing the expected structure of neighboring vertices.

Formally, given an uncertain graph $\mathcal{G}$, an integer $n \geq 2$ and a vertex $u \in V$, the expected *n*-clique cardinality of u is:

$$[\gamma_n(u)] = \sum_{c\in\mathcal{Q}_n(u)} \prod_{\substack{i<j\in c,\\ e=(i,j)}} p_e \tag{2}$$

where $\mathcal{Q}_n(u)$ is the set containing all cliques of size n that involve vertex u in $\mathcal{G}$. The notion of discrepancy of Definition 1 is extended accordingly:

DEFINITION 2. *The discrepancy $dis_n(u, G)$ of a vertex u in an instance $G \sqsubseteq \mathcal{G}$ is the difference of u's n-clique cardinality in G to its expected n-clique cardinality, i.e., $dis_n(u, G) = \gamma_n(u, G) - [\gamma_n(u, \mathcal{G})]$.*

DEFINITION 3. *Given an uncertain graph $\mathcal{G} = (V, E, p)$ and an integer $n \geq 2$, the* discrepancy *$\Delta_n(G)$ of a possible graph $G \sqsubseteq \mathcal{G}$ is defined as*

$$\Delta_n(G) = \sum_{u\in V} |dis_n(u)| \tag{3}$$

The second problem we tackle in this work is the following:

PROBLEM 2 (*clique*-REPRESENTATIVE INSTANCE). *Given an uncertain graph $\mathcal{G} = (V, E, p)$ and two integers $2 \leq l \leq n$, find a possible graph $G^*_{l,n} \sqsubseteq \mathcal{G}$ such that:*

$$G^*_{l,n} = \arg\min_{G\sqsubseteq\mathcal{G}} \sum_{m=l}^{n} \Delta_m(G) \quad \square$$

Problem 2 aims at extracting an instance that, in addition to the vertex degree, preserves the m-clique connectivity of the vertices, within given range of $[l, n]$. Intuitively, Problem 2 captures the neighborhood connectivity of the vertices. Problem 1 is a special case of Problem 2, where $l = n = 2$.

In the second direction, uncertain sparsification aims at producing an uncertain graph with fewer edges, but similar properties to the original one. Formally, given an uncertain graph $\mathcal{G} = (V, E, p)$ and an integer $k < |E|$ our goal is to produce another uncertain graph $\mathcal{G}' = (V, E', p')$ with $|E'| = k$ such that the expected degree of each vertex is preserved. We first extend the notion of vertex degree discrepancy, to uncertain graphs:

DEFINITION 4. *The discrepancy $dis(u, \mathcal{G}')$ of a vertex u in an uncertain graph $\mathcal{G}'$ is the difference of u's expected degree in $\mathcal{G}'$ to its expected degree in $\mathcal{G}$, i.e., $dis(u, \mathcal{G}') = [deg(u, \mathcal{G}')] - [deg(u, \mathcal{G})]$. Accordingly, the overall discrepancy of $\mathcal{G}'$ is $\Delta(\mathcal{G}') = \sum_{u \in V} dis(u, \mathcal{G}')$.*

Thus, the third problem we aim at solving is:

PROBLEM 3 (SPARSIFICATION). *Given an uncertain graph $\mathcal{G} = (V, E, p)$ and an integer $k < |E|$, find an uncertain graph $\mathcal{G}' = (V, E', p')$ such that:*

$$\mathcal{G}^* = \arg\min_{\mathcal{G}'} \Delta(\mathcal{G}'), \text{ with } |E'| = k \quad \square$$

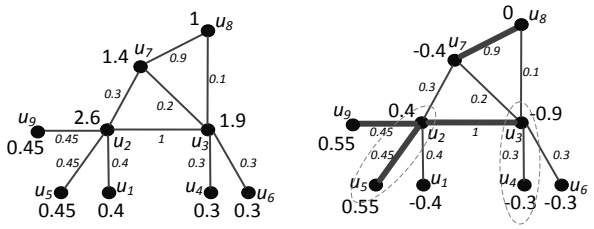
Problem 3 can be also generalized to capture properties for groups of vertices (n-cliques). In [17] we conjecture Problem 1 to be NP-Hard. Similarly, Problems 2 and 3 are expected to be NP-Hard. Therefore, we focus on approximate algorithms. A benchmark solution for all problems is based on the Most Probable (MP) graph. Specifically, for Problems 1 and 2, MP consists of the edges whose probability is at least 0.5. For Problem 3, MP contains the k uncertain edges with the highest probability.

3.2 Algorithms

Regarding Problem 1, in [17] we propose ADR and ABM, which generate approximate solutions that can yield accurate answers for a variety of tasks including clustering coefficient, betweenness centrality and shortest path distances. ADR involves two phases: 1) it creates an instance $G_0 = (V, E_0)$ of the uncertain graph that preserves the average vertex degree $[deg(\mathcal{G})]$ by choosing $\mathbf{P} = \frac{V}{2} \cdot [deg(\mathcal{G})]$ probable edges and 2) it iteratively improves G_0 by rewiring, i.e., replacing edges in E_0, so that the total discrepancy is reduced.

We illustrate the application of ADR on the uncertain graph of Figure 1(a), where edge probabilities are denoted with italics, and the expected degree is shown next to each vertex. Initially, ADR computes $\mathbf{P} = 4.4$ and rounds it to the closest integer $\lfloor \mathbf{P} \rceil = 4$. Then, it picks the 4 most probable edges of the graph and forms the set $E_0 = \{(u_2, u_3), (u_2, u_5), (u_2, u_9), (u_7, u_8)\}$. Figure 1(b) depicts the edges of E_0 with bold lines, and shows the resulting node discrepancies. The value of the total discrepancy at this stage is $\Delta = 3.8$. Next, ADR starts the second phase. Assume that at iteration 0 the algorithm randomly considers the replacement of $e_1 = (u_2, u_5) \in E_0$ with $e_2 = (u_3, u_4) \in E \setminus E_0$. Since such a replacement improves the overall discrepancy, the edges are swapped. Intuitively, the swapping reduces the overall discrepancy while maintaining the average expected degree $[deg(\mathcal{G})]$. The discrepancy of the new instance $E_1 = \{(u_2, u_3), (u_2, u_9), (u_3, u_4), (u_7, u_8)\}$ is $\Delta' = 3.5$.

ABM also involves two phases. The first performs rounding of the expected vertex degrees to the closest integers, and computes a maximal b-matching [7] using the rounded values as capacity constraints. Specifically, this phase considers all edges in random order and includes in the representative those that do not violate any constraint (i.e., no vertex has degree greater than its capacity). The second phase, partitions the vertices according to their discrepancy

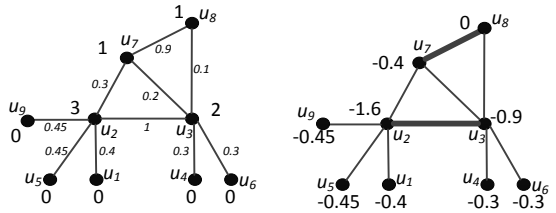


(a) initial graph (b) after Phase 1

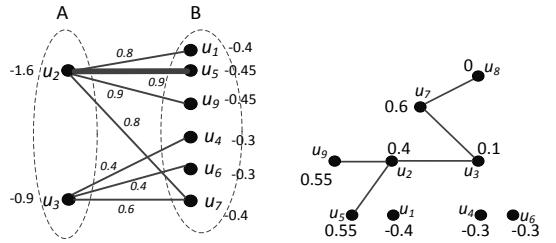
Figure 1: ADR example

and adds edges that improve the total discrepancy Δ, by performing a weighted bipartite matching.

Figure 2 applies ABM on the uncertain graph of Figure 1(a). Figure 2(a) shows the vertex degrees after rounding. Phase 1 considers in turn edges $(u_2, u_3), (u_7, u_8)$, which are added to the representative. After that, no other edge can be included because it would cause a capacity violation. Figure 2(b) contains the node discrepancies after the termination of Phase 1, with respect to their original (i.e., before rounding) degree.



(a) rounded graph $\mathcal{G}$ (b) result of Phase 1



(c) input of bipartite (d) G^* of ABM

Figure 2: ABM example

Based on their discrepancies, Phase 2 partitions the vertices into three groups *A*, *B* and *C*. *A* contains nodes with discrepancy $dis(u) \leq -0.5$, *B* the nodes for which $-0.5 < dis(u) < 0$, and *C* the rest, i.e., nodes with $dis(u) \geq 0$. The partitioning is complete (i.e., $A \cup B \cup C = V$) and there is no overlap (i.e., $A \cap B \cap C = \emptyset$). In our running example the groups are $A = \{u_2, u_3\}$, $B = \{u_1, u_4, u_5, u_6, u_7, u_9\}$ and $C = \{u_8\}$. Only edges connecting vertices of groups *A* and *B* can improve the overall discrepancy.

Figure 2(c), illustrates the second phase of ABM i.e., the approximate bipartite matching among vertices of groups *A* and *B*. First, it picks the heaviest edge (u_2, u_5) and adds it to the result. Then, it updates the discrepancy of u_2 to $dis(u_2) = -1.6 + 1 = -0.6$; since $-1 < dis(u_2) < -0.5$, the weights of edges adjacent to u_2, (i.e., (u_2, u_1), (u_2, u_7) and (u_2, u_9)) must be updated as well. Edges (u_2, u_1) and (u_2, u_7) yield a negative weight, and are discarded. Similarly, edges (u_2, u_9) and (u_3, u_7) are added (see [17] for more details). Figure 2(d) shows the final output of ABM, which combines the edges added during the two phases. The discrepancy of the extracted graph is 3.2.

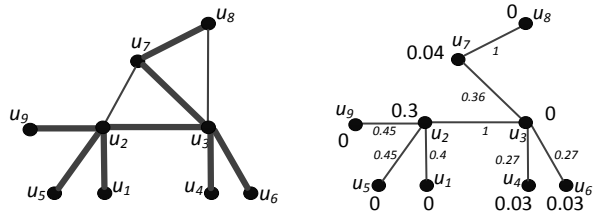
ADR and ABM focus explicitly on vertex degrees. In order to solve Problem 2, we propose GAME, a game theoretic framework that can efficiently extract representatives preserving the expected n-clique cardinality for $n \geq 2$. In our game, the players are the edges of the uncertain graph, which compete on the discrepancies of the vertices that belong to the same n-clique as their endpoints. Each edge e has two strategies: either to participate or not to participate in the deterministic representative. GAME applies *best response dynamics*, an iterative procedure during which every edge chooses selfishly its best strategy. The process stops when no player has incentive of changing its strategy, at which point the game has reached a *Nash Equilibrium*. The game is an *exact potential* game and always reaches an equilibrium, independently of the initial conditions. Our framework is generic and can be directly applied to extracting representatives G_n^* with arbitrary values of n. However, the complexity of finding n-cliques of a vertex is exponential to the value of n [16]. Thus, we focus on small clique sizes.

Regarding Problem 3, we propose BLP algorithm. BLP first generates a backbone graph that includes the k most probable edges covering all vertices. Then, these k edges absorb the probabilities of the eliminated ones by transforming Problem 3 to a Linear Program. Specifically, Theorem 1 describes this transformation.

THEOREM 1. *For any graph represented by an incidence matrix* $\mathbf{A}$ *of size* $|V| \times |E'|$ *and expected degree vector* $\mathbf{d}$ *of size* $|V| \times 1$*, an optimal solution to Problem 3 is given by the following Linear Program:*

$$\begin{aligned} \max \quad & \mathbf{1}^T \mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} \leq \mathbf{d} \\ & \mathbf{x} \in [0,1]^{|E|} \end{aligned}$$

Figure 3 illustrates the application of BLP in the running example of Figure 1(a) for $k = 8$. Figure 3(a) depicts the edges of the backbone graph with bold lines. Figure 3(b) shows the probability assignment of LP next to each edge, and the resulting degree discrepancies next to each vertex. For instance, the probability of edge (u_7, u_8) increases from 0.9 to 1, to compensate for the elimination of edge (u_3, u_8). Similarly, edge (u_3, u_7) is increased from 0.2 to 0.36. The overall discrepancy Δ becomes 0.4. Note that this assignment is optimal for the backbone graph of Figure 3.



(a) backbone graph (b) after LP

Figure 3: BLP example ($k = 8$)

In [17], we show that the representatives generated by ADR and ABM significantly outperform MP in terms of accuracy for several queries, including clustering coefficient, betweenness centrality and shortest path distance. In turn, GAME yields even lower error for neighborhood based queries (e.g., clustering coefficient), especially in dense graphs with high edge probabilities. Similarly, BLP generates a sparse uncertain graph, whose average vertex discrepancy is many times lower than that of MP. Note that the extraction of representatives or sparsified graphs is a one time effort that can be performed offline. Even for large graphs (tens of millions of edges), the execution time of all algorithms is in the order of minutes, which is a small fraction of the processing cost of most queries using Monte Carlo sampling.

4. FUTURE DIRECTIONS

This thesis facilitates uncertain graph processing and mining using (i) representative deterministic instances and (ii) sparsified uncertain graphs. In the future we intend to extend our work on both directions. For representative instances we plan to investigate additional properties that maybe of interest for specialized tasks. It will also be interesting to generate and combine multiple representatives for better approximation. Our involvement with the sparsification problem is rather recent. Accordingly, we will study alternative algorithms for generating backbone graphs and reassigning probabilities to edges. It is also possible to devise diverse algorithmic solutions e.g. based on combinatorial or game theoretic approaches. Finally, we aim at extending our methods to alternative uncertain settings such as, time dependent or streaming graphs, attributed graphs etc., where the extraction of representative or sparse models is even more challenging.

5. REFERENCES

[1] S. Abiteboul, P. Kanellakis, and G. Grahne. On the representation and querying of sets of possible worlds. In *SIGMOD*, pages 34–48, 1987.

[2] S. Asthana, et al. Predicting protein complex membership using probabilistic network reliability. *Genome Res.*, 14:1170–1175, 2004.

[3] S. Baswana and S. Sen. A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs. *Random Structures & Algorithms*, 30(4):532–563, 2007.

[4] J. Batson, et al. Spectral sparsification of graphs: theory and algorithms. *Communications of the ACM*, 56(8):87–94, 2013.

[5] P. Boldi, F. Bonchi, A. Gionis, and T. Tassa. Injecting uncertainty in graphs for identity obfuscation. *PVLDB*, 5(11):1376–1387, 2012.

[6] W. S. Fung, et al. A general framework for graph sparsification. In *ACM symposium on Theory of Computing*, 71–80, 2011.

[7] S. Hougardy. Linear time approximation algorithms for degree constrained subgraph problems. In *Research Trends in Combinatorial Optimization*, pages 185–200. Springer, 2009.

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

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

[10] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[11] R.-H. Li, et al. Efficient and accurate query evaluation on uncertain graphs via recursive stratified sampling. In *ICDE*, 892–903, 2014.

[12] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011.

[13] P. Mahadevan, et al. Systematic topology analysis and generation using degree correlations. *SIGCOMM*, 36(4):135–146, 2006.

[14] D. Monderer and L. S. Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.

[15] W. E. Moustafa, et al. Subgraph pattern matching over uncertain graphs with identity linkage uncertainty. In *ICDE*, 2014.

[16] J. Nešetřil and S. Poljak. On the complexity of the subgraph problem. *Commentationes Mathematicae Universitatis Carolinae*, 26(2):415–419, 1985.

[17] P. Parchas, F. Gullo, D. Papadias, and F. Bonchi. The pursuit of a good possible world: extracting representative instances of uncertain graphs. In *SIGMOD*, pages 967–978, 2014.

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

[19] Y. Yuan et al. Efficient subgraph search over large uncertain graphs. *PVLDB*, 4(11):876–886, 2011.

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