Uncertain Graph Sparsification (Extended Abstract)

Panos Parchas
Amazon Web Services
parchp@amazon.com

Nikolaos Papailiou
NTUA
npapa@cslab.ece.ntua.gr

Dimitris Papadias
HKUST
dimitris@cs.ust.hk

Francesco Bonchi
ISI Foundation & Eurecat
francesco.bonchi@isi.it

***Abstract*—Uncertain graphs are prevalent in several applications including communications systems, biological databases and social networks. The ever increasing size of the underlying data renders both graph storage and query processing extremely expensive. Sparsification has often been used to reduce the size of deterministic graphs by maintaining only the important edges. However, adaptation of deterministic sparsification methods fails in the uncertain setting. To overcome this problem, we introduce the first sparsification techniques aimed explicitly at uncertain graphs. The proposed methods reduce the number of edges and redistribute their probabilities in order to decrease the graph size, while preserving its underlying structure. The resulting graph can be used to efficiently and accurately approximate any query and mining tasks on the original graph, including clustering coefficient, page rank, reliability and shortest path distance.**

I. Introduction

Uncertain graphs, where edges are associated with a probability of existence, have been used widely in numerous applications. For instance, in communication systems, each edge (u, v) is often associated with a reliability value that represents the probability that the channel from u to v will not fail. In biological databases, uncertain edges between vertices representing proteins are due to error-prone laboratory measurements. In social networks, edge probabilities can model the influence between friends, or the likelihood that two users will become friends in the future.

Several techniques have been proposed for diverse query processing and mining tasks on uncertain graphs (e.g. [3]–[7]), most of which assume *possible-world* semantics. Specifically, let $\mathcal{G} = (V, E, p)$ be an uncertain (also called probabilistic) graph[1], where $p : E \rightarrow (0, 1]$ assigns a probability to each edge. $\mathcal{G}$ is interpreted as a set $\{G = (V, E_G)\}_{E_G \subseteq E}$ of $2^{|E|}$ possible deterministic graphs, each defined on a subset of E. Exact processing requires query evaluation on *all possible worlds* and aggregation of the partial results[2].

Consequently, exact processing is prohibitive even for uncertain graphs of moderate size due to the exponential number of worlds. Thus, most techniques provide approximate results by applying Monte-Carlo (MC) sampling on a random subset of possible worlds. However, even MC may be very expensive for large uncertain graphs because generating a sample is time consuming as it involves sampling each edge. Moreover, due to the high *entropy* of the uncertain graphs, there is significant variance among the possible worlds, which implies the need of numerous samples for accurate query estimation. This imposes huge overhead at query processing cost because the query must be executed at every sample.

This work was supported by grants 16201615, 16205117 from HK RGC.

[1] $\mathcal{G}$ is assumed simple, unweighted, undirected and connected.

[2] In general, the probability of a query predicate Q is derived by the sum of probabilities of all possible worlds G for which $Q(G) = true$.

II. Motivation

In order to tackle the high cost, we develop techniques for *uncertain graph sparsification*. Specifically, given $\mathcal{G}$ and a parameter $\alpha \in (0, 1)$, the proposed methods generate a sparsified probabilistic subgraph $\mathcal{G}' = (V, E', p')$, which contains a fraction of the edges, i.e., $E' : E' \subset E$, $|E'| = \alpha|E|$. $\mathcal{G}'$ preserves the structural properties of $\mathcal{G}$, has less entropy, and can be used to approximate the result of a wide range of queries on $\mathcal{G}$. Sparsification yields significant benefits in execution time because the cost of sampling is linear to the number of edges. Moreover, the required number of samples is proportional to the graph's entropy, which is lower in our sparsified graphs. Finally, similar to the case of deterministic graphs, sparsification reduces the storage cost, and facilitates visualization of complex networks.

To the best of our knowledge, this is the first work on uncertain graph sparsification. On the contrary, sparsification has received considerable attention in the deterministic graph literature ([1], [2], [8]) . In that context, most techniques aim at approximating all shortest path distances up to a multiplicative or additive factor, or preserving all cuts up to an arbitrarily small multiplicative error. As we demonstrate in our experimental evaluation, the adaptation of such methods to uncertain graphs yields poor results. On the other hand, our sparsification techniques achieve high accuracy and small variance for common graph tasks by capturing the *expected* node degrees, or the *expected* cut sizes up to a certain value. Summarizing, the contributions of this work are: 1) We propose a novel framework of uncertain graph sparsification with entropy reduction. 2) We design algorithms that reduce the number of edges and tune the probability of the remaining ones to preserve crucial properties. 3) We experimentally demonstrate that the sparsified graphs are effective for a variety of common tasks including *shortest path distance*, *reliability*, *page rank* etc.

III. Problem Definition

A prevalent goal of deterministic graph sparsification is preservation of the cut sizes [2]. The notion of a cut can be extended naturally to uncertain graphs. In this case, due to the linearity of expectation, the *expected* size of a cut is the sum

of the probabilities of the edges involved in the cut. We define the *discrepancy* $\delta(S)$ of a vertex set S in a sparsified graph $\mathcal{G}'$ as the difference of S's expected cut size in $\mathcal{G}'$ to its expected cut size in $\mathcal{G}$.

Motivated by the work in deterministic sparsification, we aim at cut-preserving sparsified graphs, or, using our notation, at minimizing discrepancy δ. The exponential number of cuts renders their exhaustive enumeration intractable. To overcome this, we target cuts of sets S with specific cardinality k.

Formally, given an integer k, we define the k-discrepancy Δ_k of a graph $\mathcal{G}'$ as the sum of the absolute values of the discrepancies for all sets with cardinality k:

$$\Delta_k(\mathcal{G}') = \sum_{S \subseteq V, |S|=k} |\delta(S)|$$

We aim at minimizing the sum of Δ_i for $1 \leq i \leq k$, or equivalently at preserving the size of all cuts up to k. Accordingly, the problem we tackle in this work is:

Problem 1: Given an uncertain graph $\mathcal{G} = (V, E, p)$, and a sparsification ratio $\alpha \in (0, 1)$, find an uncertain graph $\mathcal{G}^* = (V, E^*, p^*)$, with $|E^*| = \alpha|E|$ that minimizes the sum of discrepancies $\sum_{i=1}^{k} \Delta_i(\mathcal{G}^*)$ up to a given $k \geq 1$.

In addition to discrepancy minimization, our methods aim at entropy reduction. Observe that the two objectives are not independent because, since the sparsified graph has fewer edges, it is likely to have lower entropy as well. Minimization of discrepancy refers to the *quality* of the sparsified graph, while entropy reduction relates to the *efficiency* of query processing. The proposed techniques apply a gradient descent framework that finds a local minimum in terms of discrepancy, but adjusts the gradient step with the aim of reducing entropy.

IV. Algorithms and Evaluation

The proposed framework starts with an initialization step that generates a connected unweighted, deterministic graph G_b out of the input uncertain graph. G_b can be thought of as a backbone that ensures that connectivity is preserved in the resulting sparsified graph. We first compute a maximum spanning tree of G, where the probabilities act as weights. Then, we remove the tree edges from G and insert them to G_b. This process is repeated until G_b consists of $\alpha|E|$ edges. Then, two different techniques operate on G_b in order to produce the sparsified graph, namely *Gradient Descent Backbone* (GDB) and *Expectation Maximization Degree* (EMD). Both techniques aim at minimizing the objective function of our problem definition, i.e., preserving the expected cuts of the original graph.

Given the backbone graph $G_b = (V, E_b)$, *Gradient Descent Backbone* (GDB) initially generates a seed uncertain graph $\hat{\mathcal{G}} = (V, E_b, \hat{p})$, $\hat{p} = p$, and proceeds in iterations.. At each iteration, GDB optimizes the probability p'_e of each edge $e = (u_0, v_0)$, considering the remaining probabilities fixed.

Since GDB only updates the edge probabilities of the backbone graph $G_b = (V, E_b)$ (without inserting or removing edges), it is sensitive to the choice of G_b. On the other hand, *Expectation-Maximization Degree* (EMD) modifies both E_b and the edge probabilities. EMD is inspired by *Expectation-Maximization*, which is an iterative optimization framework that estimates two sets of interdependent unknown parameters. In our case, EMD estimates the following sets of parameters: i) the set of edges in the sparsified graph and ii) their probabilities.

Similarly to GDB, EMD starts with the input backbone graph, and the corresponding probabilities p of $\mathcal{G}$. Then, it enters the iterative process, which consists of two phases. E-phase replaces edges of E_b with edges from $E \backslash E_b$ considering the edge probabilities fixed. The new graph is denoted by $G'_b = (V, E'_b)$. M-phase calls GDB to optimize the edge probabilities considering $G'_b = (V, E'_b)$ as fixed.

In our experimental evaluation [6], we use two real undirected uncertain graphs with various sizes, densities, and edge probabilities. In order to assess the behaviour of the methods in graphs with increasing density, we also use 4 synthetic undirected datasets. We compare EMD and GDB against two benchmarks NI and SS: NI constitutes the adaptation of a cut-based deterministic sparsification method, whereas SS extends a spanner-based technique to the uncertain setting.

Summarizing the experiments, as shown in [6] the proposed techniques accurately capture the structural properties of the input uncertain graphs. The preservation of structural properties leads to precise results for various queries with different characteristics. Moreover, by reducing the entropy of the uncertain graph, our methods decrease the variance of the Monte Carlo estimator of all evaluated queries. This reduces the processing time, as considerably fewer samples are required for accurate query estimation. As opposed to the proposed methods, techniques based on deterministic sparsification (e.g., NI and SS) usually fail, in terms of result quality, variance and execution time. Finally, our algorithms are efficient and applicable to large uncertain graphs. For instance, our most expensive algorithm EMD, sparsifies *Flickr* [6] (one of our largest real datasets consisting of more than 10M edges), in just few seconds using an Intel Xeon E5-2660 with 2.20GHz CPU and 96GB RAM.

References

[1] K. J. Ahn, S. Guha and A. McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *PODS* 2012.

[2] W. S. Fung, R. Hariharan, N. J. Harvey, and D. Panigrahi. A general framework for graph sparsification. In *STOC*, 2011.

[3] A. Mukherjee, P. Xu, and S. Tirthapura. Mining maximal cliques from an uncertain graph. In *ICDE*, 2015.

[4] P. Parchas, F. Gullo, D. Papadias, and F. Bonchi. The pursuit of a good possible world: Extracting representative instances of uncertain graphs. In *SIGMOD*, 2014.

[5] P. Parchas, F. Gullo, D. Papadias, and F. Bonchi. Uncertain graph processing through representative instances. *ACM Transactions on Database Systems (TODS)*, 40(3):20, 2015.

[6] P. Parchas, N. Papailiou, D. Papadias, and F. Bonchi. Uncertain Graph Sparsification. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 30(12): 2435–2449, 2018 .

[7] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. K-nearest neighbors in uncertain graphs. *PVLDB*, 3(1-2):997–1008, 2010.

[8] V. Satuluri, S. Parthasarathy, and Y. Ruan. Local graph sparsification for scalable clustering. In *SIGMOD*, 2011.