Query-Directed Width Measures for Probabilistic Graph Homomorphisms

M2 internship topic proposal, Inria Lille, team LINKS

Advisors: Antoine Amarilli and Mikaël Monet

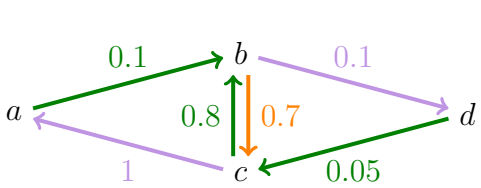
Abstract: We are interested in the following problem, which is motivated by query evaluation in probabilistic databases: given a graph G, and a graph H whose edges are annotated with independent probability values of existence, we want to compute the probability that there exists a homomorphism from G to H. It is known that, for every fixed graph G, this problem can be solved in polynomial time if we restrict the graphs H to have *bounded treewidth*. The goal of this internship is to define and study a width measure that would be parameterized by a graph G, that would be weaker than bounded treewidth while still ensuring that the problem is tractable when this measure is bounded.

Keywords: Probabilistic query evaluation, treewidth, homomorphisms, graph theory

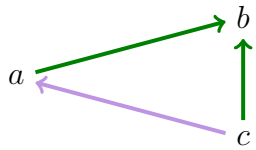
1 Topic presentation

Homomorphisms for probabilistic graphs. We consider directed graphs with colored edges, that is, triples $H = (V, E, \lambda)$ with V a finite set of *vertices*, $E \subseteq V^2$ a set of *edges*, and $\lambda : E \to \sigma$ a function that maps every edge to a *color* in a set of colors σ. A *graph homomorphism* h from some graph $G = (V_G, E_G, \lambda_G)$ to some graph $H = (V_H, E_H, \lambda_H)$ is a function $h : V_G \to V_H$ such that, for all $(u, v) \in E_G$, we have $(h(u), h(v)) \in E_H$ and further $\lambda_H((h(u), h(v))) = \lambda_G((u, v))$. We write $G \rightsquigarrow H$ when there exists a homomorphism from G to H. For example, calling H_1 the graph from Figure 1a (ignore for now the numerical annotations on the edges) and G_1 that of Figure 1c, then we have $G_1 \rightsquigarrow H_1$, as witnessed by the homomorphism h that sends w to a, x to b, y to c and z to b again.

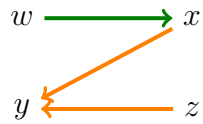
For a subset $U \subseteq E$ of the edges of a graph $H = (V, E, \lambda)$, let $H[U] := (\bigcup_{(a,b)\in U}\{a, b\}, U, \lambda_{|U})$ be the *subgraph of H with edges U*, that is, the graph whose nodes are the nodes appearing in U and whose edges are U. What we call a *probabilistic graph* is simply a pair (H, π) where H is a graph and π is a probability function $\pi : E \to [0; 1]$ that maps every edge e of H to a probability value $\pi(e)$. Such a probabilistic graph defines a probability distribution on the set of subsets U of E, where an edge $e \in E$ is in U with probability $\pi(e)$; which we can equivalently see as a probability distribution on the subgraphs of G. Formally we have $\Pr_\pi(U) := \prod_{e\in U} \pi(e) \times \prod_{e\in E\setminus U}(1 - \pi(e))$. For instance, Figure 1b depicts the subgraph of H_1 obtained by the set of edges $U = \{(a, b), (c, b), (c, a)\}$, and for the probabilistic graph (H_1, π) from Figure 1a (where probabilities are given on the edges) we have $\Pr_\pi(U) = 0.1 * 0.8 * 1 * (1 - 0.7) * (1 - 0.1) * (1 - 0.05)$.



(a) Probabilistic graph H_1



(b) Induced subgraph $H_1[U]$ for $U = \{(a, b), (c, b), (c, a)\}$



(c) Query graph G_1

Figure 1: Examples

Given a graph G and a probabilistic graph (H, π) with edge set E, we then define the *probability that there exists a homomorphism from G to H under* $\Pr_\pi$, denoted $\Pr_\pi(G \rightsquigarrow H)$, to be the sum of the probabilities of all subgraphs of H to which G has a homomorphism. Formally:

$$\Pr_\pi(G \rightsquigarrow H) := \sum_{\substack{U \subseteq E \\ \text{s.t. } G \rightsquigarrow H[U]}} \Pr_\pi(U).$$

Continuing the example, let us consider again the graphs G_1 and H_1 from Figure 1. We could compute $\Pr(G_1 \rightsquigarrow H_1)$ by summing over the possible subgraphs of H_1, but this process is generally intractable, since there are $2^{|E|}$ possible edge subsets. Here, by noticing that $G_1 \rightsquigarrow H_1[U] \iff (b, c) \in U \wedge ((a, b) \in U \vee (c, b) \in U)$, we can obtain $\Pr(G_1 \rightsquigarrow H_1) = 0.7 \times (1 - (1 - 0.1) \times (1 - 0.8))$.

Motivation. The motivation for this problem comes from the world of *probabilistic databases* [SORK11]. In this context, a probabilistic database (D, π) is a relational database D in which every tuple is annotated with a probability of existence. Given a Boolean query q, one can then define the *probability that the database satisfies q*, and study the associated computational problem. The connection is the following: we can see a probabilistic graph (H, π) as a probabilistic database over binary relations (where each relation corresponds to a color), and then computing $\Pr_\pi(G \rightsquigarrow H)$ amounts to computing the probability that the database satisfies the *conjunctive query* associated to G. For instance, the conjunctive query associated to G_1 is $\exists wxyz\ \mathrm{Green}(w, x) \wedge \mathrm{Orange}(x, y) \wedge \mathrm{Orange}(y, z)$. For this reason, we will call G the *query graph*, and H the *data graph*.

Problem studied and goal of the internship. In practice, the size of the query is negligible compared to that of the data, so we study what is called the *data complexity* of the problem: we consider that the query graph G is *fixed* and we measure the complexity only as a function of the size of the data graph. Formally, for each fixed query graph G, we wish to study the following computational problem, denoted $\mathsf{PHom}(G)$:

PROBLEM :	$\mathsf{PHom}(G)$
INPUT :	A probabilistic data graph (H, π)
OUTPUT :	$\Pr_\pi(G \rightsquigarrow H)$

Dalvi and Suciu [DS12] have shown a dichotomy result for this problem: they proved that, for every query graph G, $\mathsf{PHom}(G)$ is either solvable in polynomial time or is intractable (namely, $\mathrm{FP}^{\#\mathrm{P}}$-hard). For instance, it can be shown that for the graph G_1 from Figure 1c the problem $\mathsf{PHom}(G_1)$ can be solved in polynomial time (can you prove it?). On the other hand, for the graph $G := w \to x \to y$, $\mathsf{PHom}(G)$ is $\mathrm{FP}^{\#\mathrm{P}}$-hard. One way of making the problem tractable for every graph G is to restrict the structure of the data graph. This has been done by Amarilli, Bourhis and Senellart in [ABS15], where they consider the

treewidth of data graphs. Treewidth is a graph parameter that measures how far a graph is from being a tree; for instance a tree has treewidth 1, a cycle has treewidth 2, and a $k \times k$ grid has treewidth $O(k)$. For an integer k, consider the problem $\mathsf{PHom}_k(G)$, which is exactly like $\mathsf{PHom}(G)$ but where input data graphs are restricted to having treewidth less than k. Their results imply that $\mathsf{PHom}_k(G)$ is in polynomial time for every k and every query graph G; in other words, bounding the treewidth of data graphs is sufficient to make $\mathsf{PHom}(G)$ tractable, for every graph G.

The goal of this internship would be to define and study a variant of the treewidth measure that would be parameterized by a query graph G, and that would be less restrictive than treewidth itself. For instance, for a given query graph G, one would define a measure tw_G ensuring that if we bound $\mathrm{tw}_G(H)$ for the input graphs H then $\mathsf{PHom}(G)$ is tractable. In the long run, the goal of this line of research would be to re-explain the tractability results of [DS12] by using the treewidth approach, for instance by showing that if $\mathsf{PHom}(G)$ is tractable then tw_G is in fact always bounded, or showing that if $\mathsf{PHom}(G)$ is hard then tw_G can be unbounded.

2 Context and advisors

The internship will be carried out in LINKS[1], which is a joint research team between Inria Lille[2], the University of Lille[3], and the CRIStAL laboratory[4]. It will be supervised by Mikaël Monet[5] and co-supervised by Antoine Amarilli[6]. Mikaël Monet is an Inria full-time researcher working on theoretical aspects of uncertain data management, knowledge compilation, and more recently on applying symbolic and logical approaches to explainable AI. Antoine Amarilli is an associate professor at Télécom Paris and works on database theory, knowledge compilation, and enumeration complexity.

3 How to apply

This proposal is for an end-of-study internship (6 months), and we are looking for someone with a good background in complexity theory and a certain taste for combinatorial problems. Contact us at mikael.monet@inria.fr and a3nm@a3nm.net if you are interested!

References

[ABS15] Antoine Amarilli, Pierre Bourhis, and Pierre Senellart. Provenance circuits for trees and treelike instances. In *International Colloquium on Automata, Languages, and Programming*, pages 56–68. Springer, 2015.

[DS12] Nilesh N. Dalvi and Dan Suciu. The dichotomy of probabilistic inference for unions of conjunctive queries. *J. ACM*, 59(6):30, 2012.

[SORK11] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic Databases*. Morgan & Claypool, 2011.

[1] https://team.inria.fr/links/
[2] https://www.inria.fr/en/centre-inria-lille-nord-europe
[3] https://www.univ-lille.fr/home/
[4] https://www.cristal.univ-lille.fr/en/
[5] https://mikael-monet.net/
[6] https://a3nm.net/