Blending Uncertainty and Reasoning for advanced Query Answering at Scale

Jacopo Urbani
Vrije Universiteit Amsterdam

Expressive query answering over uncertain databases is a long-standing challenge which importance cannot be understated, especially now that deep neural networks (DNNs) are regularly employed to produce high volumes of uncertain data.

A popular language for executing complex queries is Datalog. One of the features that makes Datalog ideal for this task is that it supports recursion by reasoning using *if-then* rules [1]. Unfortunately, Datalog is a language that is designed to only on "classic" symbolic databases, lacking any support for uncertainty. To address this limitation, some extensions were proposed to support uncertainty under the possible world semantics (PWS) [5]. Blending Datalog with PWS is ideal because it combines the expressive power of Datalog rules with an elegant probabilistic view of uncertain data. Moreover, such an approach has been recently further extended to include DNNs predictions [3], thus showing how we can leverage the best of DNNs and symbolic reasoning. Such encouraging results suggest that rule-based Datalog reasoning under PWS is a promising way to integrate learning and logic. To do so, however, we must first address a scalability problem that comes from this integration.

First, it is well-known that already non-probabilistic Datalog reasoning can be challenging when applied over very large KBs [6]. This problem has motivated a wealth of research, which has ultimately conflated into a series of high-performance Datalog reasoners that *materialize* the KB, i.e., compute a universal model that allows us to perform query answering efficiently. A significant development in this space consists of organizing the inferences in a data structure called *Trigger Graph* (TG), which can be roughly seen as a blueprint that contains all instructions to compute a model without inferring duplicate derivations [6]. It has been shown that reasoning with TGs has an excellent scalability: In their largest experiment, the authors of [6] have shown that it could perform non-trivial reasoning over a KB with 17 billion facts, which is a number of facts that exceeds the size of the largest KBs available on the Web.

Unfortunately, extending Datalog reasoning to support PWS makes the materialization process much more challenging because we are required to compute *all* proofs associated with an inferred fact, in contrast to non-probabilistic Datalog where computing *one* proof is enough. Computing all proofs is needed to build the *lineage* and hence the probability of inferred facts [5]. A well-known technique to do so is the $Tc_{\mathcal{P}}$ operator [9] implemented ProbLog [4], but it comes with an important performance bottleneck [7]. To overcome this issue, the authors of [7] proposed an alternative approach, but there still remain inputs where reasoning leads to combinatorial explosion of derivations. To avoid those, Scallop [2] proposed to store only the k best proofs, but there are no known error bounds for such an approximation.

Recently, we discovered an extension of TGs that allows their usage for reasoning under PWS in a way that significantly improves the scalability of query answering [8]. The results of our research, which will be presented at the upcoming SIGMOD 2023 conference, have been implemented into a **new probabilistic database engine** called LTGs. LTGs is a open-source engine that implements Datalog reasoning using "probabilistic" TGs, compressing the storage of the proofs to reduce the combinatorial explosion produced by prior art. This allows LTGs to improve the runtime of query answering and enable exact reasoning over benchmarks that so far could only be handled with approximation techniques.

The goal of our presentation is to introduce LTGs to the Dutch-Belgian Database research community, illustrating its main features and performance improvements, hoping that this lead to a wider adoption and stimulate further research on this timely issue.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of databases*, volume 8. Addison-Wesley Reading, 1995.

[2] J. Huang, Z. Li, B. Chen, K. Samel, M. Naik, L. Song, and X. Si. Scallop: From Probabilistic Deductive Databases to Scalable Differentiable Reasoning. In *Advances in Neural Information Processing Systems*, volume 34, pages 25134–25145, 2021.

[3] R. Manhaeve, S. Dumančić, A. Kimmig, T. Demeester, and L. De Raedt. Neural probabilistic logic programming in DeepProbLog. *Artificial Intelligence*, 298:103504, 2021.

[4] L. D. Raedt, A. Kimmig, and H. Toivonen. ProbLog: A Probabilistic Prolog and Its Application in Link Discovery. In *IJCAI*, pages 2462–2467, 2007.

[5] D. Suciu, D. Olteanu, C. Ré, and C. Koch. Probabilistic Databases. *Synthesis Lectures on Data Management*, 3(2):1–180, 2011.

[6] E. Tsamoura, D. Carral, E. Malizia, and J. Urbani. Materializing Knowledge Bases via Trigger Graphs. *Proceedings of the VLDB Endowment (PVLDB)*, 14(6):943, 2021.

[7] E. Tsamoura, V. Gutiérrez-Basulto, and A. Kimmig. Beyond the Grounding Bottleneck: Datalog Techniques for Inference in Probabilistic Logic Programs. In *AAAI*, pages 10284–10291, 2020.

[8] E. Tsamoura, J. Lee, and J. Urbani. Probabilistic Reasoning at Scale: Trigger Graphs to the Rescue. In *Upcoming SIGMOD 2023*, page n.a., 2023.

[9] J. Vlasselaer, G. V. d. Broeck, A. Kimmig, W. Meert, and L. D. Raedt. Anytime Inference in Probabilistic Logic Programs with Tp-Compilation. In *IJCAI*, pages 1852–1858, 2015.