New Perspectives for Fuzzy Datalog (Extended Abstract)

Matthias Lanzinger[1], Stefano Sferrazza[1,2] and Georg Gottlob[1]

[1]University of Oxford, Department of Computer Science, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom

[2]TU Wien, Institut für Logic and Computation, Favoritenstraße 9, A-1040 Wien, Austria

1. Introduction

Today, new trends in data management introduce complex datasets and knowledge graphs where data is obtained via crowd-sourcing or from observations made by artificial intelligence systems. Moreover, existing knowledge graphs (KGs) are now commonly enriched using link prediction and KG-completion techniques [1]. Such data points are often associated with a degree of certainty, expressing the level of confidence of the human or AI source in the truth of the datum. These developments raise new challenges for data management and the need for formalisms that can take into account degrees of certainty in data[1].

Fuzzy logic has a long history as a tool for combining logical reasoning with the different types of uncertainty that are encountered in real-world settings by interpreting degrees of certainty as *degrees of truth*. This naturally motivates the study of Datalog with fuzzy logic semantics as a reasoning formalism for large databases and KGs with uncertainty. Many variants of fuzzy logic programming have been proposed in the literature [2, 3, 4, 5, 6, 7], with the most active research focusing on complex multi-adjoint settings [8, 7], or Prolog-derived semantics based on fuzzy similarity of constants and fuzzy unification procedures [5]. Alternative frameworks for reasoning with uncertainty like Markov Logic Networks [9] and Probabilistic Soft Logic [10] require extensive grounding before inference that can quickly become prohibitive when reasoning over large amounts of data. Our proposed language t-Datalog aims to be a simpler alternative focused on effective reasoning in large databases and KGs with uncertainty and aims to be the fuzzy analogue of standard Datalog. In this paper, we present the t-Datalog formalism and report on ongoing research.

In particular, we present a simple and efficient fixpoint procedure for computing minimal fuzzy models for t-Datalog. Furthermore, we show how Datalog with fuzzy semantics relates to

Datalog 2.0 2022: 4th International Workshop on the Resurgence of Datalog in Academia and Industry, September 05, 2022, Genova - Nervi, Italy

matthias.lanzinger@cs.ox.ac.uk (M. Lanzinger); stefano.sferrazza@cs.ox.ac.uk (S. Sferrazza); georg.gottlob@cs.ox.ac.uk (G. Gottlob)

0000-0002-7601-3727 (M. Lanzinger)

CEUR Workshop Proceedings (CEUR-WS.org)

[1]Uncertainty is generally not the same as a probability and typical settings mentioned in the introduction often do not fit the probabilistic database framework.

the recently proposed Datalog$^\circ$ formalism [11].

2. Datalog over t-norms

A *t-norm* is a commutative, monotone, and associative function $\odot\colon [0,1] \times [0,1] \to [0,1]$ with identity element 1. They generalise Boolean conjunction to the interval $[0,1]$ of *truth degrees*. Commonly studied t-norms include $\min$ (the Gödel t-norm), the Łukasiewicz t-norm $a \odot_L b = \max\{0, a+b-1\}$, or the real product. But t-norms can also be significantly more complex, e.g., all functions $f_p(x,y) = (x^p + y^p - 1)^{\frac{1}{p}}$ for $p < 0$ are t-norms (part of the Schweizer-Sklar family of t-norms, see [12]). For an extensive overview of t-norms and fuzzy logic, we refer to Hájek [13].

The following presentation extends recent work on a restricted version of t-Datalog that allows only the Łukasiewicz t-norm [6]. A *Datalog over t-norms (t-Datalog)* program Π is a finite set of rules where each rule ρ is of the form

$$R_1(\mathbf{x_1}) \odot_\rho \cdots \odot_\rho R_k(\mathbf{x_k}) \to R_h(\mathbf{x_h}) \tag{1}$$

where $\odot_\rho$ is some t-norm[2]. Note that a program is not limited to one specific t-norm but rather each rule can use a different t-norm. This is important to express different fuzzy behaviour in different rules. That is, in some rules, it may be natural to use the pessimistic interpretation of the Łukasiewicz t-norm $\odot_L$, while other rules in the same program may be more natural under the relatively optimistic interpretation of the Gödel t-norm $\odot_G$. Similar formalisms have been studied extensively in more general settings, see [14, 15].

To simplify presentation, we assume some fixed global signature σ and domain Dom throughout this paper. We write $GAtoms$ for the set of all ground atoms with respect to σ and Dom. A *truth assignment* is a function $\nu\colon GAtoms \to [0,1]$, intuitively assigning a degree of truth in the real interval $[0,1]$ to every ground atom. We extend the application of a truth assignment ν to ground formulas γ, γ' inductively as follows[3].

$$\begin{aligned} \nu(\gamma \odot \gamma') &= \odot(\nu(\gamma), \nu(\gamma')) \\ \nu(\gamma \to \gamma') &= \min\{1, 1 - \nu(\gamma) + \nu(\gamma')\} \end{aligned}$$

With truth degrees for atoms, it also becomes interesting to consider degrees of satisfaction. For rational $K \in (0,1]$ we say that a rule ρ is *K-satisfied* by ν if for every grounding γ of ρ over Dom it holds that $\nu(\gamma) \geq K$. In particular, a rule $\gamma \to \gamma'$ is 1-satisfied by truth assignment ν exactly when $\nu(\gamma) \leq \nu(\gamma')$. For a program Π, we say that a truth assignment ν is a *$K-$fuzzy model* if all formulas in Π are K-satisfied by ν.

In place of the database in the classical setting, we have *(finite) partial truth assignments*, that is, partial functions $\tau : GAtoms \to (0,1]$ that are defined for a *finite* number of ground atoms. Let (Π, τ) be a pair where Π is a set of formulas and τ is a partial truth assignment, a

[2]By slight abuse of notation we use the same symbol for the t-norm and the corresponding connective in a Datalog rule

[3]Note that for technical reasons there is only a single $\to$ connective and it can not vary by rule. Note also that all standard implications in fuzzy logic behave the same when considering 1-satisfiability.

K-fuzzy model of (Π, τ) is a K-fuzzy model ν of Π where $\nu(G) = \tau(G)$ for every ground atom G for which τ is defined. For a partial truth assignment τ, we write τ^* for the induced truth assignment where $\tau^*(G) = \tau(G)$ if τ is defined for G, and $\tau^*(G) = 0$ otherwise.

The natural query task, given Π, τ and $c, K \in (0, 1]$ is whether a fact G is at least true to a target degree c in all K-fuzzy models of (Π, τ). For truth assignments ν, ν', we say $\nu \leq \nu'$ if for every fact G, $\nu(G) \leq \nu'(G)$. As in Datalog, if (Π, τ) is consistent, then it has a unique minimal K-fuzzy model that can be used to evaluate the query task described above.

3. An Efficient Fixpoint Procedure for Datalog over t-norms

A step-wise evaluation of t-Datalog programs that follows standard procedures for Datalog directly may have to revise the truth of a fact multiple times, as different paths of inference may imply different degrees of truth. In this section, we show that with the right kind of procedure this is not necessary and evaluation of t-Datalog requires at most as many steps of inference as classical Datalog.

For program Π, we first define

$$\mu_\Pi(\nu) = \max\{\, \nu(\mathrm{body}(\gamma)) \mid \gamma \in \mathit{GRules}_\Pi,\, \nu(\mathrm{head}(\gamma)) < \nu(\mathrm{body}(\gamma)) + K - 1\}$$

where GRules_Π is the set of all groundings of rules in Π. The *immediate maximal K-consequence operator* $T_{\Pi,K}$ is then defined as $T_{\Pi,K}(\nu) = \nu'$, where ν is a truth assignment and ν' is the truth assignment determined by the following:

1. if G is the head of ground rule γ with $\nu(\mathrm{body}(\gamma)) = \mu_\Pi(\nu) > 0$ that is not K-satisfied by ν, then set $\nu'(G) = \nu(\mathrm{body}(\gamma)) + K - 1$,
2. otherwise set $\nu'(G) = \nu(G)$.

Clearly the operator is monotone, i.e., if $\nu_1 \leq \nu_2$, then also $T_{\Pi,K}(\nu_1) \leq T_{\Pi,K}(\nu_2)$. Let $T^{(\alpha)}_{\Pi,K}$ denote the α-fold application of $T_{\Pi,K}$. It is known that $T_{\Pi,K}$ always reaches a fixpoint in a finite number of steps, we write $T^{(\infty)}_{\Pi,K}$ for the application of the operator until a fixpoint is reached. We call the smallest α where $T^{(\alpha)}_{\Pi,K}(\nu) = T^{(\alpha+1)}_{\Pi,K}(\nu)$ the *fixpoint index* of Π, ν (for K). Observe that when ν maps only to $\{0, 1\}$ (i.e., a classical database), $T_{\Pi,1}$ is precisely the standard immediate consequence operator for Datalog. It is not difficult to see that the procedure can be used for the evaluation of t-Datalog.

Theorem 1. *Let Π be a t-Datalog program and let τ be a finite partial truth assignment. Then $T^{(\infty)}_{\Pi,K}(\tau^*)$ is the minimal K-fuzzy model of (Π, τ).*

Simpler immediate consequence operators have been studied in similar settings [7, 15] but without a computational focus. The important distinction from a practical perspective is the restriction to only considering the *immediate maximal K-consequences*, i.e., only those immediate consequences with maximum truth degree. This awareness of the truth degree reveals an important property of the procedure: the truth value of each $G \in \mathit{GAtoms}$ changes at most once during the computation of $T^{\infty}_{\Pi,K}$. In consequence, the computation of a fuzzy minimal model for a t-Datalog program requires at most as many steps as the standard fixpoint

procedure in the classical setting. In the following we write ν_Δ to refer to the truth assignment that sets $\nu_\Delta(G) = 1$ if $\nu(G) > 0$, and $\nu_\Delta(G) = 0$ otherwise.

Theorem 2. *Let Π be a t-Datalog program and let τ be a finite partial truth assignment. The fixpoint index of Π, τ^* for K is less or equal to the fixpoint index of Π, τ^*_Δ for K.*

Theorem 2 also holds when we adapt the operator $T_{\Pi,K}$ to only change one atom at a time. That is, t-Datalog reasoning requires the materialisation of at most as many atoms as reasoning in Datalog over the corresponding crisp database. This is a significant improvement over the previously mentioned Markov Logic Networks [9] and Probabilistic Soft Logic [10] which require extensive grounding before inference and suggests that reasoning in t-Datalog can be feasible even for large datasets by efficiently maintaining a list of unsatisfied rules ordered by truth degree of the body[4].

We see that the semantics of t-Datalog closely follow classical Datalog semantics, both in terms of minimal models as well as fixpoint semantics. This provides a tight link to existing Datalog literature and thus opens up clear directions for extensions analogous to work for classical Datalog. Among them, we are particularly interested in fuzzy extensions of the Datalog$^\pm$ family of languages, t-Datalog with stratified negation, as well as integration with existing Datalog reasoning systems where implementations are based on the immediate consequence operator.

4. t-norms and Datalog$^\circ$

Very recently, Abo Khamis, et al. [11] introduced Datalog$^\circ$ as an extension of Datalog over partially ordered pre-semirings. This is motivated by the desire to express numerical recursive tasks, such as linear regression or shortest path computations in a Datalog-style language. Beyond numerical applications, Fitting's three-valued logic and Belnap's four-valued logic are also explored as interesting applications of Datalog$^\circ$. Here we note that this connection to many-valued logics can be significantly expanded. Observe that for any t-norm $\odot$, $([0,1], \max, \odot, 0, 1)$ is a semiring (that also enjoys all necessary other technical properties required by Datalog$^\circ$). One can then show that for a t-Datalog program Π that mentions the same t-norm $\odot$ in all rules, the minimal 1-fuzzy models of (Π, τ) are exactly the same as the least fixpoints considered in the semantics of Datalog$^\circ$ over the semiring corresponding to $\odot$ with the natural partial-ordering of $[0,1]$ by $\leq$[5].

This raises the question of whether the Datalog$^\circ$ framework and in particular its convergence conditions, can be extended to consider individual semirings per rule (even with a shared addition monoid). This would allow full expression of t-Datalog in Datalog$^\circ$, while also opening up the possibility of combining complex fuzzy reasoning with various forms of aggregation.

[4]Note that a practical implementation of $T_{\Pi,K}$ does not actually require the materialisation of the full set $GRules_\Pi$ to compute μ_Π.

[5]It remains unclear if the same is also possible for K-fuzzy models where $K < 1$

5. Conclusion & Outlook

We reported on new motivations and ongoing research in the intersection between fuzzy logic and Datalog. The procedure introduced in Section 3 provides an important foundation for future work. We are in the process of extending the Vadalog system [16] to support arbitrary t-norms in rule bodies following the ideas presented there. Following the implementation, we plan for a large-scale experimental evaluation to verify the feasibility of t-Datalog over large uncertain datasets. Furthermore, the procedure provides a foundation for fuzzy extensions of Datalog$^{\pm}$ languages as it can naturally be extended to a chase procedure. Additionally, the observed connections to Datalog$^{\circ}$ present further promising directions for future research.

Acknowledgements

Stefano Sferrazza was supported by the Austrian Science Fund (FWF):P30930. Georg Gottlob is a Royal Society Research Professor and acknowledges support by the Royal Society in this role through the "RAISON DATA" project (Reference No. RP\R1\201074). Matthias Lanzinger acknowledges support by the Royal Society "RAISON DATA" project (Reference No. RP\R1\201074).

References

[1] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, ACM Trans. Knowl. Discov. Data 15 (2021) 14:1–14:49. URL: https://doi.org/10.1145/3424672. doi:10.1145/3424672.

[2] Á. Achs, A. Kiss, Fuzzy extension of datalog, Acta Cybernetica 12 (1995) 153–166.

[3] R. Ebrahim, Fuzzy logic programming, Fuzzy Sets Syst. 117 (2001) 215–230. URL: https://doi.org/10.1016/S0165-0114(98)00300-5. doi:10.1016/S0165-0114(98)00300-5.

[4] P. Eklund, F. Klawonn, Neural fuzzy logic programming, IEEE Trans. Neural Networks 3 (1992) 815–818. URL: https://doi.org/10.1109/72.159071. doi:10.1109/72.159071.

[5] P. J. Iranzo, F. Sáenz-Pérez, A fuzzy datalog deductive database system, IEEE Trans. Fuzzy Syst. 26 (2018) 2634–2648. URL: https://doi.org/10.1109/TFUZZ.2018.2806923. doi:10.1109/TFUZZ.2018.2806923.

[6] M. Lanzinger, S. Sferrazza, G. Gottlob, MV-Datalog+-: Effective Rule-based Reasoning with Uncertain Observations, 2022.

[7] J. Medina, M. Ojeda-Aciego, P. Vojtás, A procedural semantics for multi-adjoint logic programming, in: Proc. EPIA, volume 2258 of *Lecture Notes in Computer Science*, Springer, 2001, pp. 290–297. URL: https://doi.org/10.1007/3-540-45329-6_29. doi:10.1007/3-540-45329-6_29.

[8] M. E. Cornejo, D. Lobo, J. Medina, Syntax and semantics of multi-adjoint normal logic programming, Fuzzy Sets Syst. 345 (2018) 41–62. URL: https://doi.org/10.1016/j.fss.2017.12.009. doi:10.1016/j.fss.2017.12.009.

[9] M. Richardson, P. M. Domingos, Markov logic networks, Mach. Learn. 62 (2006) 107–136. URL: https://doi.org/10.1007/s10994-006-5833-1. doi:10.1007/s10994-006-5833-1.

[10] S. H. Bach, M. Broecheler, B. Huang, L. Getoor, Hinge-Loss Markov Random Fields and Probabilistic Soft Logic, J. Mach. Learn. Res. 18 (2017) 109:1–109:67. URL: http://jmlr.org/papers/v18/15-631.html.

[11] M. A. Khamis, H. Q. Ngo, R. Pichler, D. Suciu, Y. R. Wang, Convergence of datalog over (pre-) semirings, in: Proc. PODS, ACM, 2022, pp. 105–117. doi:10.1145/3517804.3524140.

[12] X. Zhang, H. He, Y. Xu, A fuzzy logic system based on Schweizer-Sklar t-norm, Sci. China Ser. F Inf. Sci. 49 (2006) 175–188. URL: https://doi.org/10.1007/s11432-006-0175-y. doi:10.1007/s11432-006-0175-y.

[13] P. Hájek, Metamathematics of Fuzzy Logic, volume 4 of *Trends in Logic*, Kluwer, 1998. URL: https://doi.org/10.1007/978-94-011-5300-3. doi:10.1007/978-94-011-5300-3.

[14] J. Medina, M. Ojeda-Aciego, P. Vojtás, Multi-adjoint logic programming with continuous semantics, in: Proc. LPNMR, volume 2173 of *Lecture Notes in Computer Science*, Springer, 2001, pp. 351–364. URL: https://doi.org/10.1007/3-540-45402-0_26. doi:10.1007/3-540-45402-0_26.

[15] P. Vojtás, Fuzzy logic programming, Fuzzy Sets Syst. 124 (2001) 361–370. URL: https://doi.org/10.1016/S0165-0114(01)00106-3. doi:10.1016/S0165-0114(01)00106-3.

[16] L. Bellomarini, E. Sallinger, G. Gottlob, The vadalog system: Datalog-based reasoning for knowledge graphs, Proc. VLDB Endow. 11 (2018) 975–987. URL: http://www.vldb.org/pvldb/vol11/p975-bellomarini.pdf. doi:10.14778/3213880.3213888.