A Short Introduction to Probabilistic Soft Logic

Angelika Kimmig[1,2], **Stephen H. Bach**[1], **Matthias Broecheler**[3], **Bert Huang**[1], **Lise Getoor**[1]
[1]University of Maryland, [2]KU Leuven, [3]Aurelius LLC

Abstract

Probabilistic soft logic (PSL) is a framework for collective, probabilistic reasoning in relational domains. PSL uses first order logic rules as a template language for graphical models over random variables with soft truth values from the interval $[0, 1]$. Inference in this setting is a continuous optimization task, which can be solved efficiently. This paper provides an overview of the PSL language and its techniques for inference and weight learning. An implementation of PSL is available at `http://psl.umiacs.umd.edu/`.

1 Introduction

Many problems in AI require to deal with both relational structure and uncertainty. As a consequence, there is a growing need for tools that facilitate the development of complex probabilistic models with relational structure. These tools should combine high-level modeling languages with general purpose algorithms for inference in the resulting probabilistic models or probabilistic programs. A variety of such frameworks has been developed recently, based on ideas from graphical models, relational logic, or programming languages [6, 5]. In this paper, we provide an overview of recent work on *probabilistic soft logic* (PSL) [4], a framework for collective, probabilistic reasoning in relational domains. PSL models have been developed in various domains, including collective classification [3], ontology alignment [4], personalized medicine [2], opinion diffusion [1], trust in social networks [7], and graph summarization [8]. A key distinguishing feature of PSL is its use of soft truth values in the interval $[0, 1]$. This allows one to directly incorporate similarity functions, both on the level of individuals and on the level of sets. For instance, when modeling opinions in social networks, PSL allows one to not only model different types of relations between users, such as friendship or family relations, but also multiple notions of similarity, for instance based on hobbies, beliefs, or opinions on specific topics. Technically, PSL represents the domain of interest as logical atoms. It uses first order logic rules to capture the dependency structure of the domain, based on which it builds a joint probabilistic model over all atoms. Each rule has an associated non-negative weight that captures the rule's relative importance. Due to the use of soft truth values, inference in PSL is a continuous optimization problem, which can be solved efficiently. In the following, we provide an overview of the PSL modeling language and its efficient algorithms for most probable explanation and marginal inference.

2 PSL Semantics

A PSL program consists of a set of first order logic rules with conjunctive bodies and single literal heads. Rules are labeled with non-negative weights. The following example program encodes a simple model to predict voter behavior based on a social network with two types of links denoting *friend* and *spouse* relationships:

$$0.3 : \mathit{friend}(B, A) \wedge \mathit{votesFor}(A, P) \rightarrow \mathit{votesFor}(B, P) \tag{1}$$

$$0.8 : \mathit{spouse}(B, A) \wedge \mathit{votesFor}(A, P) \rightarrow \mathit{votesFor}(B, P). \tag{2}$$

Consider any concrete persons a and b and party p instantiating logical variables A, B, and P respectively. The first rule states that if a is a friend of b and votes for party p, there is a chance that

b votes for p as well, whereas the second makes the same statement for spouses. The rule weights indicate that spouses are more likely to vote for the same party than friends.

While PSL shares the syntax of its rules with first order logic, it uses *soft truth values* from the interval $[0, 1]$ instead of the extremes 0 (false) and 1 (true) only. Given a set of atoms $\ell = \{\ell_1, \ldots, \ell_n\}$, we call the mapping $I : \ell \to [0, 1]^n$ from atoms to soft truth values an *interpretation*. PSL defines a probability distribution over interpretations that makes those satisfying more ground rule instances more probable. In the example above, we prefer interpretations where a person's vote agrees with many friends, that is, satisfies many groundings of Rule (1), and in case of a tradeoff between a friend and a spouse, agreement with the spouse is preferred due to the higher weight of Rule (2).

To determine the degree to which a ground rule is satisfied, PSL uses the *Lukasiewicz t-norm* and its corresponding *co-norm* as the relaxation of the logical AND and OR, respectively. These relaxations are exact at the extremes, but provide a consistent mapping for values in-between. Given an interpretation I, the formulas for the relaxation of the logical conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$) are as follows:

$$\begin{aligned} \ell_1 \tilde{\wedge} \ell_2 &= \max\{0, I(\ell_1) + I(\ell_2) - 1\}, \\ \ell_1 \tilde{\vee} \ell_2 &= \min\{I(\ell_1) + I(\ell_2), 1\}, \\ \tilde{\neg} l_1 &= 1 - I(\ell_1), \end{aligned}$$

where we use $\tilde{}$ to indicate the relaxation from the Boolean domain. For a ground PSL rule $r \equiv r_{body} \to r_{head} \equiv \tilde{\neg} r_{body} \tilde{\vee} r_{head}$, an interpretation I over the atoms in r determines whether r is satisfied, and, if not, its distance to satisfaction. Abusing notation, we expand the usage of I to logical formulas. The truth value of a formula is obtained by applying the above definitions of the logical operators starting from the truth values of atoms as specified by I. Given I, a rule r is satisfied, i.e., $I(r) = 1$, if and only if $I(r_{body}) \leq I(r_{head})$, that is, the head has at least the same truth value as the body. Again, this coincides with the usual definition of satisfaction of a rule when truth values are restricted to 0 and 1. The rule's *distance to satisfaction* under interpretation I then measures the degree to which this condition is violated:

$$d_r(I) = \max\{0, I(r_{body}) - I(r_{head})\}. \tag{3}$$

For instance, consider the interpretation $I = \{spouse(b, a) \mapsto 1, votesFor(a, p) \mapsto 0.9, votesFor(b, p) \mapsto 0.3\}$, and let r be the corresponding ground instance of Rule (2) above. We get $I(r_{body}) = \max\{0, 1 + 0.9 - 1\} = 0.9$ and thus $d_r(I) = \max\{0, 0.9 - 0.3\} = 0.6$, whereas the distance would be 0 if the head had truth value 0.9 or greater.

Given a set of ground atoms ℓ of interest, a PSL program induces a distribution over possible interpretations I. Let R be the set of all ground rules that are instances of a rule in the program and only mention atoms in ℓ. The probability density function f over I is:

$$f(I) = \frac{1}{Z} \exp[-\sum_{r \in R} \lambda_r (d_r(I))^p] \quad ; \quad Z = \int_I \exp[-\sum_{r \in R} \lambda_r (d_r(I))^p], \tag{4}$$

where λ_r is the weight of the rule r, Z is the continuous version of the normalization constant used in discrete Markov random fields, and $p \in \{1, 2\}$ provides a choice of two different loss functions. Informally, the linear loss function ($p = 1$) favors interpretations that completely satisfy one rule at the expense of higher distance from satisfaction for conflicting rules, whereas the quadratic loss function ($p = 2$) favors interpretations that satisfy all rules to some degree, which typically have truth values farther away from the extreme values. The values of individual atoms ℓ_i can be further restricted by linear equality and inequality constraints. We set $f(I) = 0$ whenever any such constraint is violated and constrain the domain of integration for the normalization constant Z accordingly. This allows one to encode additional domain knowledge, such as a predicate being functional. For instance, in the voter example, each voter a cannot vote for more than one of the participating parties $p_1, \ldots, p_n$, leading to a functionality constraint on $votesFor(\cdot, \cdot)$.

3 Inference and Learning in PSL

The PSL system provides efficient inference methods for the two key tasks of (a) inferring most likely values for a set of propositions given values of the remaining propositions as evidence (most

probable explanation or MPE inference) and (b) computing marginal distributions. The form of PSL programs together with the use of soft truth values ensure that the space of interpretations with non-zero density forms a convex polytope. Inference algorithms for both settings exploit this convexity to achieve efficiency. Additionally, PSL provides methods for learning weights from labeled data. We summarize the main ideas here and refer to the corresponding technical papers for full details.

MPE Inference The first common inference task in PSL is to find the most probable interpretation given evidence, that is, the most likely interpretation extending a given partial interpretation. This means maximizing the density function $f(I)$ in Equation (4), which is equivalent to minimizing the summation in the exponent, subject to both the evidence and the equality and inequality constraints. For instance, in the voting example, given the social network and the true voting behavior of a few persons obtained in a poll, MPE inference derives the most likely voting behavior of all others.

As shown by Broecheler *et al.* [4], this constrained optimization problem can be cast as a second order cone program (SOCP). The SOCP can be solved in time $O(n^{3.5})$, where n is the number of relevant rule groundings, that is, those with non-zero distance to satisfaction. In order to avoid manipulation of non-relevant rules, PSL follows an iterative approach that determines the set of relevant rules based on the truth values of the evidence atoms and the current truth values of non-evidence atoms before constructing the SOCP. Initially, a truth value of 0 is used for non-evidence atoms. After constructing and solving the SOCP, the set of relevant rules is updated based on the current MPE interpretation. This process is repeated until no more rules get activated.

Recently, Bach *et al.* [1] demonstrated that MPE inference based on consensus optimization can achieve linear scalability while being only marginally less accurate than standard cubic time SOCP solvers used in the approach discussed above. Consensus optimization splits the optimization problem into independent, small problems tied together by additional constraints. In PSL, separate subproblems are created for each ground rule. Each such subproblem uses its own local copies of literals, and introduces constraints that equate the truth values of these local copies with those of the corresponding original literal. For instance, for a given person a and party p, all groundings of Rules (1) and (2) are dependent through $votesFor(a, p)$ in the original optimization problem, but are made independent by using different copies of this atom in consensus optimization. Consensus optimization then iterates between (a) optimizing truth values of local copies as a trade-off between minimizing their contribution to the original objective and their agreement with the original atom, and (b) updating truth values of original atoms to the average of their local copies, where all subproblems have closed form solutions.

Computing Marginal Distributions The second common inference task in PSL is to calculate the probability $P(l \leq I(\ell_i) \leq u)$ that an atom ℓ_i takes a truth value from a given interval $[l, u]$. Broecheler and Getoor [3] introduce a sampling algorithm to approximate such marginal distributions, which is a #P-hard problem in the number of ground atoms in general. Intuitively, calculating $P(l \leq I(\ell_i) \leq u)$ corresponds to computing the volume of the corresponding slice of the convex polytope of non-zero density interpretations. In PSL, marginal distributions are approximated by collecting a histogram of sampled points following the hit-and-run Markov chain Monte Carlo scheme. Starting from a MAP state, which can be obtained efficiently as discussed above, the algorithm explores the convex polytope by first sampling a direction uniformly at random, followed by sampling a point on the line segment within the polytope. As the general scheme can get stuck in corners of the polytope, where most directions do not point towards the interior, these cases are detected, and a relaxation method is applied to restrict direction sampling to feasible directions.

Weight Learning The weights of rules can be learned with maximum-likelihood estimation [4]. The gradient of the log-likelihood with respect to a weight λ_i is

$$\frac{\partial}{\partial \lambda_i} \log f(I) = -\sum_{r \in R_i} (d_r(I))^p + \mathbb{E}\left[\sum_{r \in R_i} (d_r(I))^p\right], \tag{5}$$

where R_i is the set of ground rules parameterized with weight λ_i. Computing the expectation $\mathbb{E}\left[\sum_{r \in R_i} (d_r(I))^p\right]$ is intractable, so a common approximation is used: $\sum_{r \in R_i} (d_r(I^\star))^p$ where $I^\star$ is the most probable interpretation given the current weights. Additionally, new weight learning methods for PSL are an active area of research.

4 Related Work

Among the variety of probabilistic relational formalisms developed recently, Markov Logic Networks (MLNs) [9] is perhaps the most closely related to PSL. We therefore briefly summarize the key commonalities and differences here. Both PSL and MLNs use first order logic as a template language to specify undirected graphical models, where ground atoms correspond to random variables and first order formulas encode dependencies among these variables and induce the features of the graphical model. However, there are two important differences. First, PSL relaxes the Boolean truth values of MLNs to continuous, soft truth values in the interval $[0, 1]$. This allows for easy integration of similarity functions into models. Second, PSL restricts the syntax of first order formulas to that of rules with conjunctive bodies. Together, these two characteristics ensure that inference in PSL is a convex optimization problem in continuous space and therefore enable the efficient inference approaches discussed above.

Acknowledgements This work is supported by the National Science Foundation under Grant No. CCF0937094 and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D12PC00337. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/NBA, or the U.S. Government. A. Kimmig is a postdoctoral fellow of the Research Foundation Flanders (FWO Vlaanderen).

References

[1] Stephen H. Bach, Matthias Broecheler, Lise Getoor, and Dianne P. O'Leary. Scaling MPE inference for constrained continuous Markov random fields with consensus optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[2] Stephen H. Bach, Matthias Broecheler, Stanley Kok, and Lise Getoor. Decision-driven models with probabilistic soft logic. In *NIPS Workshop on Predictive Models in Personalized Medicine*, 2010.

[3] Matthias Broecheler and Lise Getoor. Computing marginal distributions over continuous Markov networks for statistical relational learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[4] Matthias Broecheler, Lilyana Mihalkova, and Lise Getoor. Probabilistic similarity logic. In *Conference on Uncertainty in Artificial Intelligence*, 2010.

[5] L. De Raedt, P. Frasconi, K. Kersting, and S. Muggleton, editors. *Probabilistic Inductive Logic Programming - Theory and Applications*, volume 4911 of *Lecture Notes in Computer Science*. Springer, 2008.

[6] Lise Getoor and Benjamin Taskar. *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

[7] Bert Huang, Angelika Kimmig, Lise Getoor, and Jennifer Golbeck. Probabilistic soft logic for trust analysis in social networks. In *International Workshop on Statistical Relational Artificial Intelligence (StaRAI 2012)*, 2012.

[8] Alex Memory, Angelika Kimmig, Stephen H. Bach, Louiqa Raschid, and Lise Getoor. Graph summarization in annotated data using probabilistic soft logic. In *Proceedings of the International Workshop on Uncertainty Reasoning for the Semantic Web (URSW)*, 2012.

[9] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.