Toward Markov Logic with Conditional Probabilities

Jens Fisseler
Department of Computer Science
FernUniversität in Hagen
58084 Hagen
Germany

Abstract

Combining probability and first-order logic has been the subject of intensive research during the last ten years. The most well-known formalisms combining probability and some subset of first-order logic are probabilistic relational models (PRMs), Bayesian logic programs (BLPs) and Markov logic networks (MLNs). Of these three formalisms, MLN is the currently most actively researched. While the subset of first-order logic used by Markov logic networks is more expressive than that of the other two formalisms, its probabilistic semantics is given by weights assigned to formulas, which limits the comprehensibility of MLNs. Based on a knowledge representation formalism developed for propositional probabilistic models, we propose an alternative way to specify Markov logic networks, which allows the specification of probabilities for the formulas of a MLN. This results in better comprehensibility, and might open the way for using background knowledge when learning MLNs or even for the use of MLNs for probabilistic expert systems.

Introduction

Representing and reasoning with (uncertain) knowledge is one of the main concerns of artificial intelligence. One way to represent and process uncertain knowledge is to use probabilistic methods, which, with the introduction of *probabilistic graphical models*, have seen increasing research interest during the last two decades. *Markov* and *Bayesian networks* are two well-known classes of probabilistic graphical models (Pearl 1988; Cowell *et al.* 1999). Bayesian networks (*BNs*) have been investigated more thoroughly than Markov networks (*MNs*), because they factorize as a product of marginal and conditional distributions, and can be used to express causal relationships. The specification of BNs by an expert, however, can be laborious or even impossible, because all of its constituting distributions must be specified, and some of these parameters may be unknown to the expert. Another way for specifying Bayesian networks is learning them from data (Buntine 1996; Jordan 1998).

There has also been some work on learning Markov networks and their subset of decomposable models from data (Bromberg, Margaritis, & Honavar 2006; Karger & Srebro 2001; Malvestuto 1991), but specifying them directly is much more difficult than for BNs, because MNs factorize as a product of potential functions, not distributions.

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Markov and Bayesian networks are propositional probabilistic models and thus have restricted expressiveness. Therefore, several approaches for combining probability and some subset of first-order logic have been developed, both within the knowledge representation and machine learning communities, see (De Raedt & Kersting 2003; Cussens 2007) for an overview. Of the approaches developed for machine learning, the best known formalisms are *probabilistic relational models* (*PRMs*), *Bayesian logic programs* (*BLPs*) and *Markov logic networks* (*MLNs*), see corresponding chapters in (Getoor & Taskar 2007). Although their models are specified in first-order logic, for inference, PRMs, BLPs and MLNs all work at a propositional level. PRMs and BLPs induce a Bayesian network, whereas MLNs induce a Markov net. The formulas of PRMs and BLPs are parametrized with (conditional) distributions, whereas a MLNs consist of weighted formulas. Therefore, MLNs are not as easily comprehensible as PRMs and BLPs, and it is also difficult to specify them by hand, e.g. as background knowledge for learning.

In this paper, we propose a way to directly specify probabilities for the formulas of a Markov logic network, and also add the ability to model conditionals with MLNs. Thus we hope to make MLNs more comprehensible, thereby making it easier for the user to use background knowledge when learning MLNs. Better comprehensibility might also facilitate the use of MLNs as a knowledge representation formalism for expert systems.

In the following section, we introduce a formalism for specifying propositional graphical models. Markov logic networks are introduced in the third section, after which we describe our novel formalism and give several examples, which demonstrate certain properties of our formalism. We close with some conclusions and aspects of further work.

Propositional Probabilistic Graphical Models with Constraints

In the introduction, we have outlined two ways of building graphical models: either learning them from data, or specify them directly, which is only feasible for Bayesian networks. Yet another way to define a graphical model is to specify

probabilistic constraints on it, using *probabilistic constraint logic* (*PCL*).

Let $V = \{1, \ldots, k\}$ denote the index set of a finite set $X_V = \{X_v \mid v \in V\}$ of random variables, each with a finite domain $\mathcal{X}_v$. Let further P denote a joint probability distribution for X_V, with state space $\Omega := \prod_{i=1}^{k} \mathcal{X}_i$.

We want to be able to specify *equality constraints* on the joint distributions for X_V, i.e. constrain the *expected value* $\mathbb{E}_P(f) := \sum_{\omega \in \Omega} f(\omega)P(\omega)$ of so-called *feature functions* $f : \Omega \to [-1, 1]$. The feature functions f are not arbitrarily defined, but induced by propositional formulas, as described below. Therefore, the feature functions represent properties of the desired joint distribution, e.g. relationships between the random variables in X_V.

For $X_i \in X_V$ and $x_i \in \mathcal{X}_i$, the expression $X_i = x_i$ is an *(atomic) formula*, and if ϕ, ψ and ρ are formulas, then $(\phi \wedge \psi)$, $(\phi \vee \psi)$ and $\neg\rho$ are formulas. The set of *interpretations* for these propositional formulas is Ω, the state space of the joint distributions for X_V. An interpretation $\omega \in \Omega$ is a *model* for an atomic formula $X_i = x_i$, iff $\omega_i = x_i$, i.e. the i-th element of ω equals x_i. This definition of a model is extended to more complex formulas as usual, and for any formula ρ, we denote the set of its models by $\mathrm{Mod}_{PL}(\rho) \subseteq \Omega$. Based on this, we can compute the probability of any formula ρ with respect to any joint distribution P for X_V as $P(\rho) = \sum_{\omega \in \mathrm{Mod}_{PL}(\rho)} P(\omega)$.

The sentences expressible with PCL are probabilistic rules and facts. Let ϕ, ψ and ρ be propositional formulas, and let ξ and ζ be real numbers in $[0, 1]$. Then $(\phi \mid \psi)[\xi]$ is a *probabilistic rule*, with premise ψ, conclusion ϕ and probability ξ. Probabilistic rules with a tautological premise are *probabilistic facts*, written $(\rho)[\zeta] \equiv (\rho \mid \top)[\zeta]$. A probability distribution P is a model of a probabilistic rule $R = (\phi \mid \psi)[\xi]$, written $P \models R$, iff $P(\phi \wedge \psi) = \xi \cdot P(\psi)$, and it is a model of a set $\mathcal{R}$ of rules, written $P \models \mathcal{R}$, iff it is a model of each rule $R_i \in \mathcal{R}$. $\mathrm{Mod}_{PCL}(\mathcal{R})$ denotes the set of all models for $\mathcal{R}$.

As to semantics, models for a set $\mathcal{R}$ of probabilistic rules can be equivalently described by the expected values of certain feature functions. Given a single probabilistic rule $R = (\phi \mid \psi)[\xi]$, we define a feature function $f : \Omega \to [-1, 1]$ representing R as

$$f(\omega) := \begin{cases} 1 - \xi & \text{if} \quad \omega \in \mathrm{Mod}_{PL}(\phi \wedge \psi), \\ -\xi & \text{if} \quad \omega \in \mathrm{Mod}_{PL}(\neg\phi \wedge \psi), \\ 0 & \text{if} \quad \omega \in \mathrm{Mod}_{PL}(\neg\psi). \end{cases} \tag{1}$$

A probability distribution P is a model of R, iff $\mathbb{E}_P(f) = 0$:

$$\begin{aligned} & P \in \mathrm{Mod}_{PCL}(R) \\ :\Leftrightarrow \quad & P(\phi \wedge \psi) = \xi \cdot P(\psi) \\ \Leftrightarrow \quad & \sum_{\omega \in \mathrm{Mod}_{PL}(\phi \wedge \psi)} P(\omega) = \xi \cdot \sum_{\omega \in \mathrm{Mod}_{PL}(\psi)} P(\omega) \\ \Leftrightarrow \quad & \sum_{\omega \in \mathrm{Mod}_{PL}(\phi \wedge \psi)} P(\omega) \\ & \quad - \xi \cdot \sum_{\omega \in \mathrm{Mod}_{PL}((\phi \wedge \psi) \vee (\neg\phi \wedge \psi))} P(\omega) = 0 \\ \Leftrightarrow \quad & \sum_{\omega \in \mathrm{Mod}_{PL}(\phi \wedge \psi)} P(\omega) - \xi \cdot \sum_{\omega \in \mathrm{Mod}_{PL}(\phi \wedge \psi)} P(\omega) \\ & \quad - \xi \cdot \sum_{\omega \in \mathrm{Mod}_{PL}(\neg\phi \wedge \psi)} P(\omega) = 0 \\ \Leftrightarrow \quad & (1 - \xi \cdot) \sum_{\omega \in \mathrm{Mod}_{PL}(\phi \wedge \psi)} P(\omega) \\ & \quad - \xi \cdot \sum_{\omega \in \mathrm{Mod}_{PL}(\neg\phi \wedge \psi)} P(\omega) = 0 \\ \Leftrightarrow \quad & \sum_{\omega \in \Omega} f(\omega)P(\omega) = \mathbb{E}_P[f] = 0. \end{aligned}$$

This is canonically extended to a set $\mathcal{R} = \{R_1, \ldots, R_m\}$ of probabilistic rules with corresponding feature functions f_i, $1 \le i \le m$.

The set $\mathrm{Mod}_{PCL}(\mathcal{R})$ is either empty (if the constraints are inconsistent), contains exactly one model, or contains infinitely many models, as it is a convex subset of the set of all probability distributions for X_V. Thus, if we want to obtain point probabilities for queries, as with BNs and MNs, we have to select a single model from $\mathrm{Mod}_{PCL}(\mathcal{R})$. One way to make this choice is based on the *Principle of Maximum Entropy* (Kapur & Kesavan 1992), which proposes to choose from $\mathrm{Mod}_{PCL}(\mathcal{R})$ the probability distribution with maximum entropy:

$$P_{ME} := \operatorname*{argmax}_{P \in \mathrm{Mod}_{PCL}(\mathcal{R})} \underbrace{\left(- \sum_{\omega \in \Omega} P(\omega) \log_2 P(\omega) \right)}_{=: H(P)}. \tag{2}$$

The reason for using maximum entropy as the selection criterion is that the probabilistic constraints represent all that we know, and we want a model that represents this knowledge, but is as unbiased as possible with respect to what we don't know (Shore & Johnson 1980).

As the entropy function $H(P)$ is concave, and $\mathrm{Mod}_{PCL}(\mathcal{R})$ is a convex set, (2) gives rise to a convex optimization problem, which can be solved with the *generalized iterative scaling* (*GIS*) procedure (Darroch & Ratcliff 1972). The resulting joint probability distribution P_{ME} has the form

$$P_{ME}(\omega) = \frac{1}{Z} \exp\left(\sum_{i=1}^{m} \lambda_i f_i(\omega) \right), \tag{3}$$

therefore P_{ME} is a *log-linear model* of a Markov network. $Z := \sum_{\omega \in \Omega} \exp(\sum_{i=1}^{m} \lambda_i f_i(\omega))$ is the usual normalization constant, the f_i are the feature functions defined for each rule $R_i \in \mathcal{R}$ according to (1), and $\{\lambda_1, \ldots, \lambda_m\}$ are the parameters calculated with the GIS algorithm.

The probabilistic constraint logic introduced in this section has been used to implement the expert system shell SPIRIT (Rödder, Reucher, & Kulmann 2006), and a data mining algorithm for learning probabilistic rules has also been implemented (Fisseler, Kern-Isberner, & Beierle 2007). It is a viable and theoretically well-founded approach for representing uncertain knowledge (Kern-Isberner 1998a; Shore & Johnson 1980).

Markov Logic Networks

Markov logic (Domingos & Richardson 2007) is a formalism for specifying a probability distribution over the possible worlds of a first-order knowledge base, i.e. over its interpretations (Halpern 1990). As this set is in principle infinite, certain provisions and restrictions have to be made to ensure its finiteness (see below). Although Markov logic has been recently extended to infinite domains (Singla & Domingos 2007), in this paper we are only concerned with finite domains.

A *Markov logic network* (*MLN*) L consists of a set $\{(F_1, w_1), \ldots, (F_m, w_m)\}$ of pairs (F_i, w_i), where each F_i is a formula of first-order logic and w_i is a real number, a *weight* which is attached to F_i. Given a set of constants $C = \{c_1, \ldots, c_{|C|}\}$, L induces a (ground) Markov network $M_{L,C}$. Therefore, a MLN can be seen as a template for generating ground Markov networks.

Given a Markov logic network L and a set C of constants, we can associate the first-order signature $\Sigma_{L,C} = (\mathrm{pred}(L), \mathrm{func}(L), C)$ with (L, C), where $\mathrm{pred}(L)$ (resp. $\mathrm{func}(L)$) denotes the set of predicate (resp. function) symbols occurring in the formulas F_i. To ensure there is only a finite set of interpretations for $\Sigma_{L,C}$, (Domingos & Richardson 2007) make the following assumptions:

Unique names Different constants denote different objects.

Domain closure The only objects in the domain are those representable using the constant and function symbols in $\Sigma_{L,C}$.

Known functions For each function in $\Sigma_{L,C}$, the value of this function applied to every tuple of arguments is known and in C.

These assumptions ensure that the set of random variables of the ground Markov network $M_{L,C}$ induced by L and C is finite. This set consists of the ground atoms formed of the predicate symbols in $\mathrm{pred}(L)$ and the constants in C, i.e. it is the Herbrand base of $(\mathrm{pred}(L), \emptyset, C)$:

$$X_{L,C} := \{p(t_1, \ldots, t_k) \mid p \in \mathrm{pred}(L), t_1, \ldots, t_k \in C\}.$$

The range of the random variables in $X_{L,C}$ is $\{true, false\}$.

The state space of the joint probability distribution defined by (L, C) is $\Omega_{L,C} := \mathcal{P}(X_{L,C})$[1], i.e. the set of all Herbrand interpretations of $(\mathrm{pred}(L), \emptyset, C)$, and the distribution is given by

$$P_{L,C}(\omega) := \frac{1}{Z} \exp\left(\sum_{i=1}^{m} w_i n_i(\omega)\right), \tag{4}$$

where $Z := \sum_{\omega \in \Omega_{L,C}} \exp\left(\sum_{i=1}^{m} w_i n_i(\omega)\right)$ is the usual normalization constant and $n_i(\omega)$ denotes the number of ground instances of F_i true in ω:

$$n_i(\omega) := |\{g \mid g \in \mathrm{gnd}(F_i, C), \omega \models g\}|.$$

$\mathrm{gnd}(F_i, C)$ is the set of all ground instances of formula F_i, generated by substituting every variable of F_i by every constant in C and completely evaluating all occurring ground function terms. $\omega \models g$ means that the Herbrand interpretation ω is a model of the ground formula g.

If we define a feature function $f_i : \Omega_{L,C} \times \mathrm{gnd}(F_i, C) \to \{0, 1\}$,

$$f_i(\omega, g) := \begin{cases} 1 & \text{if} \quad \omega \models g, \\ 0 & \text{if} \quad \omega \not\models g, \end{cases} \tag{5}$$

for every formula F_i, (4) can be rewritten as

$$P_{L,C}(\omega) = \frac{1}{Z} \exp\left(\sum_{i=1}^{m} w_i \sum_{g \in \mathrm{gnd}(F_i, C)} f_i(\omega, g)\right). \tag{6}$$

This clearly shows that $M_{L,C}$ is represented as a log-linear model, where the weight w_i is shared by all ground feature functions of formula F_i.

Knowledge Representation with Markov Logic

Assume we want to formalize the following knowledge in Markov logic:

$$\begin{array}{lll} \text{R7.1:} & \textit{Common-Cold}(a) & [0.01] \\ \text{R7.2:} & \textbf{if } \textit{Susceptible}(a) & \\ & \textbf{then } \textit{Common-Cold}(a) & [0.1] \\ \text{R7.3:} & \textbf{if } \textit{Contact}(a, b) & \\ & \textbf{and } \textit{Common-Cold}(b) & \\ & \textbf{then } \textit{Common-Cold}(a) & [0.6] \end{array} \tag{7}$$

Note that we have adopted the convention of (Domingos & Richardson 2007) and are writing predicate names and constants with the first letter in upper-case letter, whereas the first letter of variable names is in lower-case.

(R7.1) states that one normally does not have a common cold (only with probability 0.01). Rule (R7.2) states that a person catches a common cold with probability 0.1 if this person is susceptible to it, and (R7.3) expresses the knowledge that person a, which was in contact with another person b that had the common cold, also gets a common cold with probability 0.6.

A plain (but inadequate, as will be shown) way to encode these *if-then*-rules with Markov logic is to represent them using material implication:

$$\begin{array}{lll} \text{R8.1:} & \textit{Common-Cold}(a). & [w_{8.1}] \\ \text{R8.2:} & \textit{Common-Cold}(a) \leftarrow \textit{Susceptible}(a). & [w_{8.2}] \\ \text{R8.3:} & \textit{Common-Cold}(a) \leftarrow \textit{Contact}(a, b) & \\ & \qquad \wedge \textit{Common-Cold}(b). & [w_{8.3}] \end{array} \tag{8}$$

Having defined the formulas, the next step is to specify their weights, but currently there is no way of directly computing the weights from prescribed probabilities. As (Domingos & Richardson 2007) gives an interpretation for the weight of a formula F_i as the log-odds between a world where F_i is true and a world where it is false, other things being equal, one might choose $w_{8.1} := \ln \frac{1}{100}$, $w_{8.2} := \ln \frac{1}{10}$ and $w_{8.3} := \ln \frac{6}{10}$. But as the ground instances of the formulas (R8.1)–(R8.3) share some of their ground atoms, they influence each other, and therefore this naive approach for computing the weights is not feasible. For example,

[1] Given a set S, $\mathcal{P}(S)$ denotes its *power set*.

with $C = \{U, V\}$, the probability (cf. (4)) of U having a common cold is $P(\textit{Common-Cold}(U)) = 0.066416$, which is significantly higher than the desired probability 0.01, which has been used to specify $w_{8.1}$. On the contrary, $P(\textit{Common-Cold}(U) \mid \textit{Contact}(U, V) \wedge \textit{Common-Cold}(V)) = 0.031946$, which is much lower than the prescribed probability 0.6.

Since, as we have seen, there is no direct way of interpreting the weights by probabilities, a pragmatic way of specifying the weights for the formulas of a MLN is to learn them from data. This is unproblematic when doing data mining, but for knowledge representation, one generally has no appropriate data set available and therefore has to generate one which represents the given formulas with the desired probabilities. This is not straightforward for complex knowledge bases. Therefore, we generated a data set representing only the desired probabilities of (R8.1) and (R8.2), containing information about 100 individuals $\{C_1, \ldots, C_{100}\}$. To represent (R8.1), for one of these individuals, w.l.o.g. C_1, $\textit{Common-Cold}(C_1)$ must be true. Representing (R8.2) is not that unequivocal. If the empirical probability of the material implication (R8.2) shall be 0.1, then $\textit{Susceptible}(a)$ must be true for $\{C_{11}, \ldots, C_{100}\}$, whereas is must be false for $\{C_1, \ldots, C_{10}\}$, because then (R8.2) is true for $\{C_1, \ldots, C_{10}\}$ only. Lets call this data set the *material data set*. On the other hand, a data set generated based on the *intended* interpretation of (R7.2) (see above) would contain $\textit{Common-Cold}(C_1)$ and $\textit{Susceptible}(C_1), \ldots, \textit{Susceptible}(C_{10})$, because only one in ten persons that are susceptible to common cold actually has a common cold. Lets call this data the *conditional data set*.

Learning weights from these two data sets gives quite different results. For any constant U, in both models the probability $P(\textit{Common-Cold}(U))$ is close to the intended probability 0.01: 0.010528 for the material data set model, 0.010762 for the conditional data set model. But the models differ significantly with respect to the probability of the conditional query $P(\textit{Common-Cold}(U) \mid \textit{Susceptible}(U))$: the conditional data set model yields 0.099655, which is very close to the intended probability 0.1, whereas the material data set model yields 0.011059. This is because the material data set model models the material implication (R8.2) (for which we get $P(\textit{Common-Cold}(U) \leftarrow \textit{Susceptible}(U)) = 0.100298$), not the intended conditional probability of (R7.2).

Please note that the difference in the models computed from the material and conditional data set does not stem from the fact that we have learned the weights from data. It would be the same if there was a direct way for computing the weights from the prescribed probabilities. The real problem is that material implication is inadequate for representing the meaning of *if-then*-rules (Gabbay & Guenthner 2001). Therefore, we introduce a formalism that allows for the adequate representation of *if-then*-rules with conditionals, and also enables us to directly attach probabilities to the formulas representing our knowledge.

Markov Logic with Probabilistic Constraints

We now introduce our proposal for Markov logic with probabilistic constraints, which is essentially a combination of Markov logic and probabilistic conditional logic (cf. section "Propositional Probabilistic Graphical Models with Constraints"). A similar formalism has been introduced within the probabilistic logic programming research community (Kern-Isberner & Lukasiewicz 2004).

Assume that instead of pairs (F_i, w_i) of first-order formulas F_i and weights w_i our knowledge base $\mathcal{R}$ consists of *(first-order) probabilistic constraints* $(\phi_i \mid \psi_i)[\xi_i]$, $1 \leq i \leq m$, where ϕ_i and ψ_i are formulas from a subset of first-order logic, and ξ_i are real numbers in $[0, 1]$. If the universally quantified formula $\forall \psi_i$ is a tautology, $(\phi_i \mid \psi_i)[\xi_i]$ can also be written as $(\phi_i)[\xi_i]$.

We currently assume that the formulas don't contain function symbols and quantifiers. Quantifiers are disallowed because the probabilistic constraints are assumed to represent general knowledge, and therefore are implicitly universally quantified (see below). The reasons for omitting function symbols are discussed in the "Examples" subsection below.

Given a set of probabilistic constraints $\mathcal{R} = \{(\phi_i \mid \psi_i)[\xi_i]\}$ and a set of constants $C = \{c_1, \ldots, c_{|C|}\}$, we can proceed as with "classical" Markov logic and associate the first-order signature $\Sigma_{\mathcal{R},C} = (\text{pred}(\mathcal{R}), \emptyset, C)$ with $(\mathcal{R}, C)$. Using the unique names and domain closure assumptions (see section on "Markov Logic Networks"), every interpretation of $\Sigma_{\mathcal{R},C}$ uniquely corresponds to an element of $\Omega_{\mathcal{R},C} := \mathcal{P}(X_{\mathcal{R},C})$, the set of all Herbrand interpretations of $(\text{pred}(\mathcal{R}), \emptyset, C)$, with $X_{\mathcal{R},C} := \{p(t_1, \ldots, t_k) \mid p \in \text{pred}(\mathcal{R}), t_1, \ldots, t_k \in C\}$.

Unlike a Markov logic network, which defines a unique probability distribution, $\mathcal{R}$ only *constraints* the set of probability distributions for $X_{\mathcal{R},C}$. A probability distribution $P_{\mathcal{R},C}$ for $X_{\mathcal{R},C}$ is a model of a probabilistic constraint $R = (\phi \mid \psi)[\xi]$, written $P_{\mathcal{R},C} \models R$, iff for all ground instances $(g_\phi \mid g_\psi) \in \text{gnd}((\phi \mid \psi), C)$ of R, $P_{\mathcal{R},C}(g_\phi \wedge g_\psi) = \xi \cdot P_{\mathcal{R},C}(g_\psi)$. $P_{\mathcal{R},C}$ is a model of a set of probabilistic constraints $\mathcal{R}$, written $P_{\mathcal{R},C} \models \mathcal{R}$, iff it is a model of each probabilistic constraint $R_i \in \mathcal{R}$. $\text{Mod}_{\textit{FO-PCL}}(\mathcal{R})$ denotes the set of all models of $\mathcal{R}$.

As our proposal is a template framework for generating ground Markov networks, we can use probabilistic constraint logic to model these Markov networks. Therefore, we define a feature function $f_{(g_{\phi_i} \mid g_{\psi_i})} : \Omega_{\mathcal{R},C} \to [-1, 1]$,

$$f_{(g_{\phi_i} \mid g_{\psi_i})}(\omega) := \begin{cases} 1 - \xi & \text{if} \quad \omega \models (g_{\phi_i} \wedge g_{\psi_i}), \\ -\xi & \text{if} \quad \omega \models (\neg g_{\phi_i} \wedge g_{\psi_i}), \\ 0 & \text{if} \quad \omega \not\models g_{\psi_i}. \end{cases} \tag{9}$$

for every ground instance $(g_{\phi_i} \mid g_{\psi_i}) \in \text{gnd}((\phi_i \mid \psi_i), C)$ of every probabilistic constraint $(\psi_i \mid \phi_i)[\xi_i] \in \mathcal{R}$ (cf. (1)).

Because we have mapped first-order probabilistic constraints to PCL, the set $\text{Mod}_{\textit{FO-PCL}}(\mathcal{R})$ of models is either empty, contains exactly one model or infinitely many. Using the same rationale as with PCL, choosing from $\text{Mod}_{\textit{FO-PCL}}(\mathcal{R})$ the model with maximum entropy yields the least biased model, which is given by

$$P_{\mathcal{R},C}(\omega) = \frac{1}{Z} \exp\left(\sum_{i=1}^{m} \sum_{(g_{\phi_i} \mid g_{\psi_i})} \lambda_{(g_{\phi_i} \mid g_{\psi_i})} f_{(g_{\phi_i} \mid g_{\psi_i})}(\omega) \right). \tag{10}$$

Comparing (10) to (6), one can see that our proposal introduces a parameter $\lambda_{(g_{\phi_i} \mid g_{\psi_i})}$ for *every* ground instance

$\|C\|$	Optimal parameters
2	$\lambda_{(\text{R11.1})} = 0.169043101\times10^{-10}$ $\lambda_{(\text{R11.2})} = 21.47483647\times10^{8}$ $\lambda_{(\text{R11.3})} = 37.19700113$
3	$\lambda_{(\text{R11.1})} = 0.241946210\times10^{-10}$ $\lambda_{(\text{R11.2})} = 21.47483647\times10^{8}$ $\lambda_{(\text{R11.3})} = 22.29916591$
4	$\lambda_{(\text{R11.1})} = 0.322136648\times10^{-10}$ $\lambda_{(\text{R11.2})} = 21.47483647\times10^{8}$ $\lambda_{(\text{R11.3})} = 13.3572512009$

Table 1: Maximum entropy parameters for the ground instances of the probabilistic constraints depicted in (11).

$(g_{\phi_i} \,|\, g_{\psi_i})$ of a probabilistic constraint $(\phi_i \,|\, \psi_i)[\xi_i]$, whereas the corresponding parameter of a Markov logic network, weight w_i, is shared by all ground instances of a formula F_i. This *parameter sharing* is an intrinsic property of all formalisms for combining probability and first-order logic, and desirable from a computational complexity viewpoint. But the following examples show that, although parameter sharing does occur within our formalisms, it is not always appropriate and therefore none of its intrinsic properties.

Examples

Assume we want to model the *if-then*-rules from (7) as first-order probabilistic constraints:

$$\begin{aligned}&\text{R11.1: } (\textit{Common-Cold}(a))[0.01],\\ &\text{R11.2: } (\textit{Common-Cold}(a) \,|\, \textit{Susceptible}(a))[0.1],\\ &\text{R11.3: } (\textit{Common-Cold}(a) \,|\, \textit{Contact}(a,b)\\ &\qquad\qquad \wedge \textit{Common-Cold}(b))[0.6].\end{aligned} \tag{11}$$

Note that, although constraints of the form $(\phi_i)[\xi_i]$ allow us to specify probabilities for formulas with material implication, e.g. as $(\textit{Common-Cold}(a) \leftarrow \textit{Susceptible}(a))[0.1]$, conditional constraints of the form $(\phi_i \,|\, \psi_i)[\xi_i]$ are more adequate for modeling uncertain *if-then*-rules (Halpern 1990; Gabbay & Guenthner 2001), which is why we have used them for the examples in this subsection.

Using the expert system shell SPIRIT (Rödder, Reucher, & Kulmann 2006), we have built different models of example (11), letting the number of constants vary between two and four. Although no parameter sharing assumption was applied, all ground instances of the same probabilistic constraints had the same optimal parameter value, which are shown in table 1. Example (11) therefore exhibits parameter sharing. This is because the knowledge it contains for each ground instance of the same probabilistic constraint is identical, therefore the least biased (i.e. maximum entropy) model should of course yield the same parameter value for each of them.

Note that conditional constraints allow the appropriate modeling of *if-then*-rules. For example, modeling only the first two rules (R11.1) and (R11.2), as we have done when learning the weights for the MLN example, we get $P(\textit{Common-Cold}(U)) = 0.01$ and $P(\textit{Common-Cold}(U) \,|\, \textit{Susceptible}(U)) = 0.1$, again for any constant U. This is the intended meaning of the rules (R7.1) and (R7.2).

Whereas example (11) exhibits parameter sharing, the following example demonstrates that additional knowledge can lead to different optimal parameter values for ground instances of probabilistic constraints. Assume we have more specific knowledge for some ground instance of a probabilistic constraint, as depicted by (R12.2) in example (12).

$$\begin{aligned}&\text{R12.1: } (R(a) \,|\, S(a))[0.6],\\ &\text{R12.2: } (R(U))[0.2].\end{aligned} \tag{12}$$

Generating a ground instance of this knowledge base with the constants $\{U, V, W\}$ results in three ground instances of (R12.1) and one ground instance of (R12.2). As we have the same knowledge for the ground instances of (R12.1) generated for V and W, they share the same maximum entropy parameter value $\lambda_{(R(V) \,|\, S(V))} = \lambda_{(R(W) \,|\, S(W))} \approx 1.5$. But for the ground instance $(R(U) \,|\, S(U)$ we get $\lambda_{(R(U) \,|\, S(U))} \approx 23.75216239$, and for constraint (R12.2), which causes this "asymmetry", we get $\lambda_{(R(U))} \approx 0.06315214$.

"Asymmetry" like the one exhibited by example (12) can also occur if (known) functions were allowed. But more importantly, formulas with function symbols can result in inconsistent knowledge. Suppose we are given the formulas depicted in example (13):

$$\begin{aligned}&\text{R13.1: } (R(a) \,|\, S(a))[0.8],\\ &\text{R13.2: } (R(a) \,|\, S(t(a)))[0.4].\end{aligned} \tag{13}$$

Generating a ground instance of these rules with $C := \{U\}$ and t defined as $t(U) := U$ results in two ground constraints, $(R(U) \,|\, S(U))[0.8]$ and $(R(U) \,|\, S(U))[0.4]$. There is obviously no probability distribution which can model both constraints, because they are inconsistent. Inconsistencies can also occur in knowledge bases without function symbols, and how to handle them is an interesting subject of future work.

Computing P_{ME}

Within PCL, the optimal parameter vector λ for the probability distribution with maximum entropy (cf. (3)) is computed with the GIS procedure. This involves computing the expected value of the feature functions f_i given the current parameter vector, and calculating appropriate adjustments for the different parameters λ_i. These computations can be sped up by using a junction tree to represent the probability distribution in a more compact, factorized form.

Although first-order probabilistic constraints are mapped to propositional probabilistic constraints, the resulting models of our formalism generally are too large for exact inference, which is why we have to resort to approximate inference, both for computing the optimal model parameters $\lambda_{(g_{\phi_i} \,|\, g_{\psi_i})}$, and for answering queries to the resulting model.

Note that, as we have seen in the previous subsection, some model parameters λ_i are shared by all ground instances of a first-order probabilistic constraint $(\phi_i \,|\, \psi_i)[\xi_i]$. Therefore, an essential aspect of an efficient implementation of

our approach is to find sufficient conditions for parameter sharing. As this is closely related to lifted probabilistic inference, the work of (de Salvo Braz, Amir, & Roth 2005; Jaimovich, Meshi, & Friedman 2007) might offer suitable initial results.

Conclusions and Further Work

We have introduced a formalism for specifying probabilities for the formulas of a Markov logic network, and have further extended Markov logic with conditionals, which are more suitable for knowledge representation. The specification of probabilities instead of weights makes the resulting models more comprehensible. As these formula probabilities do not specify a single model, a model selection step is necessary, which results in an optimization problem. Although our formalism does not have the intrinsic property of parameter sharing, which is a central aspect of all formalisms combining probability and first-order logic, we have given examples which demonstrate that our formalism does exhibit this property. Therefore, an important aspect of further work is to find sufficient conditions for parameter sharing and exploit them when solving the optimization problem.

Another important area of future research is learning uncertain first-order formulas from data. As the weights of Markov logic formulas not only depend on the individual formulas, but also on their interaction, formulas with probabilities instead of weights would clearly be a benefit for interpreting models learned from data. Furthermore, these models might be more readily used as the knowledge base of an expert system. For this application domain, we also want to further evaluate the general properties of our formalism from a knowledge representation perspective.

In conclusion, we think we have introduced a promising and viable approach for representing and learning uncertain first-order knowledge.

Acknowledgments The research reported here was partly supported by the DFG – Deutsche Forschungsgemeinschaft (grant BE 1700/7-1).

References

Bromberg, F.; Margaritis, D.; and Honavar, V. 2006. Efficient Markov network structure discovery using independence tests. In Ghosh, J.; Skillicorn, D. B.; and Srivastava, J., eds., *Proceedings of the Sixth SIAM International Conference on Data Mining*, 141–152. SIAM.

Buntine, W. L. 1996. A guide to the literature on learning probabilistic networks from data. *IEEE Transaction on Knowledge and Data Engineering* 8(2):195–210.

Cowell, R. G.; Dawid, A. P.; Lauritzen, S. L.; and Spiegelhalter, D. J. 1999. *Probabilistic Networks and Expert Systems*. Springer.

Cussens, J. 2007. Logic-based formalisms for statistical relational learning. In Getoor and Taskar (2007). 269–290.

Darroch, J. N., and Ratcliff, D. 1972. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics* 43(5):1470–1480.

De Raedt, L., and Kersting, K. 2003. Probabilistic logic learning. *SIGKDD Explorations* 5(1):31–48.

de Salvo Braz, R.; Amir, E.; and Roth, D. 2005. Lifted first-order probabilistic inference. In *Proceedings of the 2005 International Joint Conferences on Artificial Intelligence*, 1319–1325.

Domingos, P., and Richardson, M. 2007. Markov logic: A unifying framework for statistical relational learning. In Getoor and Taskar (2007). 339–371.

Fisseler, J.; Kern-Isberner, G.; and Beierle, C. 2007. Learning uncertain rules with CONDORCKD. In Wilson, D., and Sutcliffe, G., eds., *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference*, 74–79. AAAI Press.

Gabbay, D. M., and Guenthner, F., eds. 2001. *Handbook of Philosophical Logic*, volume 4. Kluwer Academic Publishers, 2nd edition. chapter Conditional Logic, 1–98.

Getoor, L., and Taskar, B., eds. 2007. *Introduction to Statistical Relational Learning*. MIT Press.

Halpern, J. Y. 1990. An analysis of first-order logics of probability. *Artificial Intelligence* 46(3):311–350.

Jaimovich, A.; Meshi, O.; and Friedman, N. 2007. Template based inference in symmetric relational Markov random fields. In *Proceedings of the 23rd Conference in Uncertainty in Artificial Intelligence*. AUAI Press.

Jordan, M. I., ed. 1998. *Learning in Graphical Models*. Kluwer Academic Publishers.

Kapur, J. N., and Kesavan, H. K. 1992. *Entropy Optimization Principles with Applications*. Academic Press, Inc.

Karger, D., and Srebro, N. 2001. Learning Markov networks: Maximum bounded tree-width graphs. In *SODA '01: Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, 392–401. SIAM.

Kern-Isberner, G., and Lukasiewicz, T. 2004. Combining probabilistic logic programming with the power of maximum entropy. *Artificial Intelligence* 157:139–202.

Kern-Isberner, G. 1998a. Characterizing the principle of minimum cross-entropy within a conditional-logical framework. *Artificial Intelligence* 98(1–2):169–208.

Malvestuto, F. M. 1991. Approximating discrete probability distributions with decomposable models. *IEEE Transactions on Systems, Man, and Cybernetics* 21(5):1287–1294.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Rödder, W.; Reucher, E.; and Kulmann, F. 2006. Features of the expert-system-shell SPIRIT. *Logic Journal of IGPL* 14(3):483–500.

Shore, J. E., and Johnson, R. W. 1980. Axiomatic derivation of the principle of maximum entropy and the the principle of minimum cross-entropy. *IEEE Transactions on Information Theory* 26(1):26–37.

Singla, P., and Domingos, P. 2007. Markov logic in infinite domains. In *Proceedings of the 23rd Conference in Uncertainty in Artificial Intelligence*, 368–375. AUAI Press.