Mach Learn (2011) 83: 241–264
DOI 10.1007/s10994-010-5206-7

Constraint-based probabilistic modeling for statistical abduction

Taisuke Sato · Masakazu Ishihata · Katsumi Inoue

Received: 17 September 2009 / Revised: 17 June 2010 / Accepted: 5 July 2010 /
Published online: 19 August 2010

Abstract We introduce a new framework for logic-based probabilistic modeling called *constraint-based probabilistic modeling* which defines *CBPMs (constraint-based probabilistic models)*, i.e. conditional joint distributions $P(\cdot \mid KB)$ over independent propositional variables constrained by a knowledge base *KB* consisting of clauses. We first prove that generative models such as PCFGs and discriminative models such as CRFs have equivalent CBPMs as long as they are discrete. We then prove that CBPMs in infinite domains exist which give existentially closed logical consequences of *KB* probability one. Finally we derive an EM algorithm for the parameter learning of CBPMs and apply it to statistical abduction.

Keywords Probabilistic model · Constraint · Abduction

1 Introduction

Suppose we have i.i.d. data as a bag of ground atoms and wish to build their logic-based probabilistic model (Getoor and Taskar 2007; De Raedt and Kersting 2008). Theoretically there are many ways to do it but current approaches seem classified into two types, *feature-based discriminative approaches* and *rule-based generative approaches*. The former type defines a log-linear model $P(x) = Z^{-1} \exp(\sum_i w_i f_i(x))$ where the f_i's are "features", i.e. real-valued functions returning in the case of Boolean ones 1 (true) or 0 (false),

Editors: Hendrik Blockeel, Karsten Borgwardt, Luc De Raedt, Pedro Domingos, Kristian Kersting, Xifeng Yan.

T. Sato (✉) · M. Ishihata
Tokyo Institute of Technology, 2-12-1, Ookayama, Meguro-ku, Tokyo, Japan
e-mail: sato@mi.cs.titech.ac.jp

M. Ishihata
e-mail: ishihata@mi.cs.titech.ac.jp

K. Inoue
National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, Japan
e-mail: kii@nii.ac.jp

the w_i's weights and Z a normalizing constant. For example CFDs (case-factor diagrams) (McAllester et al. 2004) adopt Boolean features to define a distribution over a "feasible" set of truth assignments x for Boolean variables. MLNs (Markov logic networks) (Richardson and Domingos 2006) use first-order clauses as features that count the number of clauses' ground instances which are true in the Herbrand interpretation x.

Contrastingly the latter type, rule-based approaches such as SLPs (Muggleton 1996), ICL (Poole 1997), PRISM (Sato and Kameya 2001, 2008) and more recently ProbLog (De Raedt et al. 2007), employ logical rules, i.e. definite or general clauses to describe a probabilistic data generation process. They proof-theoretically define a distribution over ground atoms (Muggleton 1996; Poole 1997), or model-theoretically define a probability measure on possible worlds, i.e. the set of Herbrand interpretations (Sato and Kameya 2001; De Raedt et al. 2007). Joint distributions thus defined are a subclass of log-linear models where the normalizing constant is unity but able to represent a variety of probabilistic models from BNs (Bayesian networks) to PCFGs (probabilistic context free grammars).

In this paper[1,2] we introduce *constraint-based probabilistic modeling*, a new modeling framework which deals with the above two types uniformly. It defines *CBPMs (constraint-based probabilistic models)*, i.e. conditional joint distributions, or more generally conditional probability measures $P_c(\cdot \mid KB)$[3] on the Herbrand interpretations for KB such that $P_c(\cdot)$ is a finite or infinite product of Bernoulli distributions and KB is a set of propositional or first-order clauses. It is motivated by an observation that it is difficult for the current rule-based approaches to deal with some kind of logical knowledge including disjunctive knowledge such as $win(rock) \vee win(paper) \vee win(scissors)$, looping rules such as $friend(x, y) \Leftarrow friend(y, x)$ and cyclic causal chains such as chemical reactions among metabolites in a metabolic network (Chen et al. 2008). We wish to logically express those types of knowledge by CBPMs using arbitrary clauses in probabilistic modeling and apply logical inference and statistical inference together.

The basic idea of CBPMs is simple: independent propositional variables (ground atoms) are constrained by a knowledge base KB so that their joint distribution is the intended one. To illustrate it, let us take the simplest example and consider a distribution $P(X = x)$ for a single random variable X that takes on values $\{a, b\}$. We represent $P(X = x)$ as a CBPM as follows. Introduce a propositional variable $\lceil X = a \rceil$ corresponding to the event $X = a$ together with probability $P_c(\lceil X = a \rceil) = \frac{P(X=a)}{1+P(X=a)}$ and similarly for $\lceil X = b \rceil$, where $P_c(\cdot)$ is a joint distribution that makes $\lceil X = a \rceil$ and $\lceil X = b \rceil$ independent. Since either $X = a$ or $X = b$ always happens but they never happen together, we impose a logical constraint, exclusive-or, $KB_X = (\lceil X = a \rceil \vee \lceil X = b \rceil) \wedge \neg(\lceil X = a \rceil \wedge \lceil X = b \rceil)$ on them. Then we see

$$
\begin{aligned}
&P_c(\lceil X = a \rceil \mid KB_X) \\
&\quad = \frac{P_c(\lceil X = a \rceil \wedge KB_X)}{P_c(KB_X)} \\
&\quad = \frac{P_c(\lceil X = a \rceil) P_c(\neg \lceil X = b \rceil)}{P_c(\lceil X = a \rceil) P_c(\neg \lceil X = b \rceil) + P_c(\neg \lceil X = a \rceil) P_c(\lceil X = b \rceil)}
\end{aligned}
$$

[1]This paper is a revised version of a conference paper presented at ILP 2009 with the same title augmented with proofs for theorems and new sections for infinite CBPMs, related work and a new learning experiment.

[2]Distributions are discrete throughout the paper.

[3]A joint distribution $P(X_1 = x_1, \ldots, X_n = x_n)$ is a probability measure on $\mathbf{R}^n$ induced by random variables $X_1, \ldots, X_n$. When n is countably infinite, the distribution is considered as a probability measure on $\mathbf{R}^\infty$. To deal with finite n and infinite n uniformly, when the context is clear, we use "joint distribution" and "probability measure" synonymously.

$$= \frac{P(X=a)}{P(X=a)+P(X=b)}$$
$$= P(X=a).$$

Similarly $P_c(\lceil X=b \rceil \mid KB_X) = P(X=b)$ holds. In this way $P(X=x)$ is representable as a CBPM $P_c(\lceil X=x \rceil \mid KB_X)$ $(x \in \{a,b\})$.

This example is trivial but by developing the idea behind it, we can show that CBPMs are expressive and powerful. That is, they can express both generative models such as PCFGs and discriminative models such as CRFs (conditional random fields) (Lafferty et al. 2001). In addition, we can represent our first-order knowledge in a knowledge base KB and perform logical inference freely. When domains (Herbrand universes) are finite, we can prove that logically equivalent KBs define the same distribution, and hence we may replace one KB with another as long as they are logically equivalent, which would be difficult for feature-based approaches that use logical formulas as features.[4] Also we allow any types of clause in KB be they non-Horn ones or looping ones that might cause an infinite loop in logic programs. Furthermore despite CBPMs' broad coverage of probabilistic models, their probabilities are uniformly computed and learned from data by the *EMC algorithm* described in Sect. 4.

Primary advantages of CBPMs from the viewpoint of knowledge representation are generality covering generative and discriminative models and logical expressiveness due to the use of first-order clauses. The treatment of infinite domains is another advantage: CBPMs are always definable even for infinite domains and for arbitrary KBs. From a machine learning point of view, what is new is the EMC algorithm, an EM algorithm applicable to discrete (binarized) conditional distributions with hidden variables.

In what follows, after introducing CBPMs, we prove basic theorems in Sect. 2. Section 3 defines CBPMs in infinite domains. In Sect. 4, we get back to finite domains and derive the EMC algorithm for the parameter learning of CBPMs. In Sect. 5, we apply CBPMs to constraint-based statistical abduction. Section 6 contains related work and Sect. 7 is conclusion.

2 Constraint-based probabilistic models

2.1 Introducing CBPMs

Let $\mathcal{L}$ be a countable first order language, $\mathcal{U}_\mathcal{H}$ the *Herbrand universe*, i.e. the set of ground terms in $\mathcal{L}$ and $\mathcal{B}_\mathcal{H}$ the *Herbrand base*, i.e. the set of ground atoms in $\mathcal{L}$. We fix an enumeration $A_1, A_2, \ldots$ of ground atoms in $\mathcal{B}_\mathcal{H}$ and identify a 0-1 vector such as $(1,0,\ldots)$ with a *Herbrand interpretation*, i.e. truth assignment for $\mathcal{B}_\mathcal{H}$ such that $A_1 = 1$ (true), $A_2 = 0$ (false) ... If it makes a closed formula φ true, it is called a *Herbrand model* of φ. We use the Cartesian product $\mathcal{I}_\mathcal{H} = \prod_i \{0,1\}_i$ to denote the set of all Herbrand interpretations. We assume each $\{0,1\}_i$ representing the truth values of A_i is a discrete probability space with a probability measure $\mu_i(\cdot)$.

[4] Consider a non-ground unit clause $P(x)$ representing $\forall x\, P(x)$. In the Herbrand universe $\{a,b\}$, it is logically equivalent to $P(a) \wedge P(b)$ or to the clause set $\{P(a), P(b)\}$, but $\{P(x)\}$, $\{P(a) \wedge P(b)\}$ and $\{P(a), P(b)\}$ are different feature sets. Likewise $\{A, A \Rightarrow B\}$ and $\{A, B\}$ are logically equivalent but different Boolean feature sets. When we replace logically equivalent but different feature sets, we need to adjust parameters (weights) to preserve the distribution.

Let $P_c(\cdot)$ be a product probability measure of the μ_i's on $\mathcal{I}_\mathcal{H}$. We consider A_i as a binary random variable from $\mathcal{I}_\mathcal{H}$ to $\{0,1\}$ such that $A_i(\omega) = x_i$ for $\omega = (x_1, x_2, \ldots) \in \mathcal{I}_\mathcal{H}$ with $P_c(A_i = x_i) = \mu_i(\{x_i\})$ $(x_i \in \{0,1\})$. $P_c(\cdot)$ makes all ground atoms independent and every closed formula φ in $\mathcal{L}$ is a binary random variable such that $\varphi(\omega) = 1$ if $\omega \models \varphi$ else 0 for $\omega \in \mathcal{I}_\mathcal{H}$ with the probability $P_c(\varphi = 1) = P_c(\{\omega \in \mathcal{I}_\mathcal{H} \mid \omega \models \varphi\})$. We write $P_c(\varphi)$ (resp. $P_c(\neg\varphi)$) instead of $P_c(\varphi = 1)$ (resp. $P_c(\varphi = 0)$)[5] and $P(x)$ instead of $P(X = x)$ when the context is clear. We use $V(X)$ for the set of values a random variable X takes.

A *CBPM (constraint-based probabilistic model)* is a conditional probability measure $P_c(\cdot \mid KB)$ on the Herbrand interpretations $\mathcal{I}_\mathcal{H}$ where KB is a set of countably many clauses. We assume KB is consistent. Although $P_c(\varphi \mid KB)$, the conditional probability of a closed formula φ, is definable measure-theoretically for any KB, when $P_c(KB) = 0$, we are unable to define it as $\frac{P_c(\varphi \wedge KB)}{P_c(KB)}$. So hereafter, to make probability computation feasible and discussion simple, we assume, unless otherwise stated, that $\mathcal{L}$ has no function symbol, $\mathcal{B}_\mathcal{H}$ is finite and $P_c(KB) > 0$ (see Sect. 3 for the infinite case).

Consider a joint distribution $P(X_1 = x_1, \ldots, X_N = x_N)$ and a CBPM $P_c(\lceil X_1 = x_1 \rceil, \ldots, \lceil X_N = x_N \rceil \mid KB)$ where the $\lceil X_i = x_i \rceil$'s are arbitrary propositional variables (ground atoms)[6] such that $\lceil X_i = x_i \rceil$ uniquely corresponds to the event $X_i = x_i$ $(x_i \in V(X_i), 1 \le i \le N)$. When $P_c(\lceil X_1 = x_1 \rceil, \ldots, \lceil X_N = x_N \rceil \mid KB) = P(X_1 = x_1, \ldots, X_N = x_N)$ holds for every x_i $(x_i \in V(X_i), 1 \le i \le N)$,[7] we say that $P_c(\lceil X_1 = x_1 \rceil, \ldots, \lceil X_N = x_N \rceil \mid KB)$ *is equivalent to* $P(X_1 = x_1, \ldots, X_N = x_N)$. We prove a basic theorem on CBPMs.

Theorem 1 *Every joint distribution has an equivalent CBPM.*

Proof Let $P(X_1 = x_1, \ldots, X_N = x_N)$ be a joint distribution. We correspondingly introduce *state variables* $\lceil X_i = x_i \rceil$ and *parameter variables* $\theta_{x_1,\ldots,x_N}$ $(x_i \in V(X_i), 1 \le i \le N)$, and define KB as follows.

$$XOR(X_i) = \left(\bigvee_{x_i \in V(X_i)} \lceil X_i = x_i \rceil \right) \wedge \bigwedge_{x_i \neq x_i'} \neg \left(\lceil X_i = x_i \rceil \wedge \lceil X_i = x_i' \rceil \right),$$

$$XOR = \bigwedge_{i=1}^{N} XOR(X_i),$$

$$EQU = \bigwedge_{x_1,\ldots,x_N} \left(\bigwedge_{i=1}^{N} \lceil X_i = x_i \rceil \Leftrightarrow \theta_{x_1,\ldots,x_N} \right),$$

$$KB = XOR \wedge EQU.$$

[5] This applies to a set of formulas as well.

[6] Propositional variables and ground atoms are synonymous in this paper.

[7] We sometimes use vector notation such as $P(\boldsymbol{X} = \boldsymbol{x})$ when non-ambiguous which denotes $P(X_1 = x_1, \ldots, X_N = x_N)$ where $\boldsymbol{X} = (X_1, \ldots, X_N)$ and $\boldsymbol{x} = (x_1, \ldots, x_N)$.

$XOR(X_i)$ says that X_i exclusively takes on one of the values in $V(X_i)$. KB is equivalent to the following DNF formula where disjuncts are mutually exclusive.

$$KB \Leftrightarrow \bigvee_{x_1,\ldots,x_N} \left\{ \bigwedge_{i=1}^{N} \left(\lceil X_i = x_i \rceil \wedge \bigwedge_{x_i' \neq x_i} \neg \lceil X_i = x_i' \rceil \right) \wedge \left(\theta_{x_1,\ldots,x_N} \wedge \bigwedge_{(x_1,\ldots,x_N) \neq (x_1',\ldots,x_N')} \neg \theta_{x_1',\ldots,x_N'} \right) \right\}.$$

Next introduce a joint distribution $P_c(\cdot)$ which makes all the variables hitherto introduced independent such that

$$P_c(\lceil X_1 = x_1 \rceil, \ldots, \lceil X_N = x_N \rceil, \theta_{x_1,\ldots,x_N}) = \left(\prod_{i=1}^{N} P_c(\lceil X_i = x_i \rceil) \right) P_c(\theta_{x_1,\ldots,x_N}),$$

$$P_c(\lceil X_i = x_i \rceil) = 1/2 \quad \text{for } \forall i, x_i \in V(X_i),$$

$$P_c(\theta_{x_1,\ldots,x_N}) = \frac{P(X_1 = x_1, \ldots, X_N = x_N)}{1 + P(X_1 = x_1, \ldots, X_N = x_N)}.$$

It holds that $0 \leq P_c(\theta_{x_1,\ldots,x_N}) \leq 1/2$ and $P(X_1 = x_1, \ldots, X_N = x_N) = \frac{P_c(\theta_{x_1,\ldots,x_N})}{P_c(\neg\theta_{x_1,\ldots,x_N})}$. We see, by calculation,

$$\begin{aligned} P_c(KB) &= \alpha \sum_{x_1,\ldots,x_N} \prod_{i=1}^{N} \left(\frac{P_c(\lceil X_i = x_i \rceil)}{P_c(\neg \lceil X_i = x_i \rceil)} \right) \left(\frac{P_c(\theta_{x_1,\ldots,x_N})}{P_c(\neg\theta_{x_1,\ldots,x_N})} \right) \\ &= \alpha \sum_{x_1,\ldots,x_N} \left(\frac{P_c(\theta_{x_1,\ldots,x_N})}{P_c(\neg\theta_{x_1,\ldots,x_N})} \right) = \alpha \sum_{x_1,\ldots,x_N} P(X_1 = x_1, \ldots, X_N = x_N) \\ &= \alpha \end{aligned}$$

where

$$\alpha = \left(\prod_{i=1}^{N} \prod_{x_i} P_c(\neg \lceil X_i = x_i \rceil) \right) \prod_{x_1,\ldots,x_N} P_c(\neg\theta_{x_1,\ldots,x_N}) > 0.$$

Hence we conclude

$$\begin{aligned} &P_c(\lceil X_1 = x_1 \rceil, \ldots, \lceil X_N = x_N \rceil \mid KB) \\ &= \frac{\left(\alpha \frac{P_c(\theta_{x_1,\ldots,x_N})}{P_c(\neg\theta_{x_1,\ldots,x_N})} \right)}{\alpha} = \frac{P_c(\theta_{x_1,\ldots,x_N})}{P_c(\neg\theta_{x_1,\ldots,x_N})} \\ &= P(X_1 = x_1, \ldots, X_N = x_N). \end{aligned}$$

□

2.2 CBPMs for BNs

Theorem 1 is general and applicable to BNs. However if applied to a BN defining a joint distribution $P(X_1 = x_1, \ldots, X_N = x_N) = \prod_{i=1}^{N} P(X_i = x_i \mid \Pi_i = \pi_i)$ where Π_i is the random vector consisting of parent variables of X_i and π_i is its value, we need as many as

$|V(X_1) \times \cdots \times V(X_N)|$ parameter variables, which is exponential in N. A better way to construct an equivalent but more compact CBPM is to respect the factorized structure of the BN and to encode CPTs (conditional probability tables) $P(X_i = x_i \mid \Pi_i = \pi_i)$ individually following (Chavira and Darwiche 2005). They introduce Boolean formulas EQU_i and $XOR(X_i)$[8] $(1 \leq i \leq N)$ where

$$EQU_i = \bigwedge_{x_i, \pi_i} \left(\lceil X_i = x_i \rceil \wedge \lceil \Pi_i = \pi_i \rceil \Leftrightarrow \theta_{x_i|\pi_i} \right),$$

$$KB_{\mathrm{BN}} = \bigwedge_{i=1}^{N} XOR(X_i) \wedge EQU_i$$

and compile the conjunction, KB_{BN}, to an AC (arithmetic circuit) that can compute arbitrary marginal probabilities of the original BN. Since no joint distribution is given to the Boolean variables in their approach, they are not random variables. However, using the above encoding, we can prove the equivalence as follows. First put $P_c(\lceil X_i = x_i \rceil) = 1/2$ and $P_c(\theta_{x_i|\pi_i}) = \frac{P(X_i = x_i \mid \Pi_i = \pi_i)}{1 + P(X_i = x_i \mid \Pi_i = \pi_i)}$ for $\forall i$, x_i and π_i similarly to Theorem 1. We have

$$\begin{aligned} P_c(KB_{\mathrm{BN}}) &= \alpha_{\mathrm{BN}} \sum_{x_1, \ldots, x_N} \prod_{i=1}^{N} \left(\frac{P_c(\theta_{x_i|\pi_i})}{P_c(\neg\theta_{x_i|\pi_i})} \right) \\ &= \alpha_{\mathrm{BN}} \sum_{x_1, \ldots, x_N} \prod_{i=1}^{N} P(X_i = x_i \mid \Pi_i = \pi_i) \\ \text{where } \alpha_{\mathrm{BN}} &= \prod_{i=1}^{N} \left(\prod_{x_i} P_c(\neg\lceil X_i = x_i \rceil) \right) \left(\prod_{x_i, \pi_i} P_c(\neg\theta_{x_i|\pi_i}) \right) \end{aligned}$$

and consequently

$$\begin{aligned} &P_c(\lceil X_1 = x_1 \rceil, \ldots, \lceil X_N = x_N \rceil \mid KB_{\mathrm{BN}}) \\ &= \frac{\prod_{i=1}^{N} P(X_i = x_i \mid \Pi_i = \pi_i)}{\sum_{x_1, \ldots, x_N} \prod_{i=1}^{N} P(X_i = x_i \mid \Pi_i = \pi_i)} \\ &= P(X_1 = x_1, \ldots, X_N = x_N). \end{aligned} \tag{1}$$

Thus the number of parameter variables $\theta_{x_i|\pi_i}$ required by this construction is the same as the number of parameters in the original BN.

We have two comments on KB_{BN}. First even if a BN has a cyclic directed path, we can construct KB_{BN} and define $P_c(\lceil X_1 = x_1 \rceil, \ldots, \lceil X_N = x_N \rceil \mid KB_{\mathrm{BN}})$ as long as CPTs are assigned to all nodes. So it can define a joint distribution for cyclic BNs as well as for acyclic BNs. In the cyclic case, the equation (1) tells us that the defined distribution is obtained by normalizing the product of CPTs.

Second, one might imagine BN probability computation by way of KB_{BN} is inefficient, but Chavira and Darwiche showed the opposite (Chavira and Darwiche 2005). They demonstrated that this logical encoding outperforms the standard junction tree algorithm by a wide

[8] $XOR(X_i)$ is given in the proof of Theorem 1.

margin when optimizations exploiting "local structure"[9] (and subsequent processing to the AC formula) are performed. CBPMs thus can be computationally efficient, if the model class and a computation mechanism are appropriate.

2.3 CBPMs for log-linear models

Having seen CBPMs are a general scheme for representing (discrete) joint distributions, we consider two special cases. The first one is log-linear models where a joint distribution $P(\boldsymbol{X}=\boldsymbol{x})$ is given as a product of potential functions $P(\boldsymbol{X}=\boldsymbol{x}) = Z^{-1}\prod_{i=1}^{M} F_i(\boldsymbol{x}_i)$ ($\bigcup_{i=1}^{M}\boldsymbol{x}_i = \boldsymbol{x}$) such that $F_i(\boldsymbol{x}_i)$ is non-negative and not identically zero.[10] Here $\boldsymbol{X}$, $\boldsymbol{x}$ and $\boldsymbol{x}_i (\subseteq \boldsymbol{x})$ are vectors and Z is a normalizing constant. In this subsection we show that log-linear models have an equivalent CBPM with the same factorization. The basic strategy is to syntactically separate potential functions and let them generate outputs independently but filter out incompatible ones by *equality constraint*s defined below.

Let us consider a joint distribution $P(X=a, Y=b, Z=c) \propto F_1(a,b)F_2(b,c)$. To find an equivalent CBPM, we introduce a *factor distribution* $Q^{(1)}(X=a, Y_1=b) = \frac{F_1(a,b)}{\sum_{a,b} F_1(a,b)}$ and its equivalent CBPM $P_c^{(1)}(\lceil X=a\rceil, \lceil Y_1=b\rceil \mid KB_1)$ as in Theorem 1. Likewise introduce another factor distribution $Q^{(2)}(Y_2=b, Z=c)$ by normalizing $F_2(b,c)$ and an equivalent CBPM $P_c^{(2)}(\lceil Y_2=b\rceil, \lceil Z=c\rceil \mid KB_2) = Q^{(2)}(Y_2=b, Z=c) = \frac{F_2(b,c)}{\sum_{b,c} F_2(b,c)}$. Y_1 and Y_2 are syntactic variants of Y in the original distribution $P(X=a, Y=b, Z=c)$. So $V(Y) = V(Y_1) = V(Y_2)$ holds. They are called *connection variable*s for Y. Next construct a product distribution $P_c = P_c^{(1)} \times P_c^{(2)}$ such that $P_c^{(1)}$ and $P_c^{(2)}$ are marginal distributions of P_c. By simple calculation, we find that

$$
\begin{aligned}
&P(X=a, Y=b, Z=c)\\
&\quad= \frac{F_1(a,b)F_2(b,c)}{\sum_{a,b,c} F_1(a,b)F_2(b,c)} = \frac{Q^{(1)}(X=a, Y_1=b)Q^{(2)}(Y_2=b, Z=c)}{\sum_{a,b,c} Q^1(X=a, Y_1=b)Q^{(2)}(Y_2=b, Z=c)}\\
&\quad= \frac{P_c^{(1)}(\lceil X=a\rceil, \lceil Y_1=b\rceil \mid KB_1)P_c^{(2)}(\lceil Y_2=b\rceil, \lceil Z=c\rceil \mid KB_2)}{\sum_{a,b,c} P_c^{(1)}(\lceil X=a\rceil, \lceil Y_1=b\rceil \mid KB_1)P_c^{(2)}(\lceil Y_2=b\rceil, \lceil Z=c\rceil \mid KB_2)}\\
&\qquad KB_1 \text{ and } KB_2 \text{ have no variable in common and independent w.r.t. } P_c\\
&\quad= \frac{P_c(\lceil X=a\rceil \wedge \lceil Y_1=b\rceil \wedge \lceil Y_2=b\rceil \wedge \lceil Z=c\rceil \wedge KB_1 \wedge KB_2)}{P_c\left(\bigvee_{a,b,c}(\lceil X=a\rceil \wedge \lceil Y_1=b\rceil \wedge \lceil Y_2=b\rceil \wedge \lceil Z=c\rceil \wedge KB_1 \wedge KB_2)\right)}\\
&\quad= \frac{P_c(\lceil X=a\rceil \wedge \lceil Y_1=b\rceil \wedge \lceil Y_2=b\rceil \wedge \lceil Z=c\rceil \wedge KB_1 \wedge KB_2)}{P_c\left(\bigvee_a \lceil X=a\rceil \wedge \bigvee_b(\lceil Y_1=b\rceil \wedge \lceil Y_2=b\rceil) \wedge \bigvee_c \lceil Z=c\rceil \wedge KB_1 \wedge KB_2\right)}\\
&\qquad KB_1 \Rightarrow \bigvee_a \lceil X=a\rceil,\ KB_2 \Rightarrow \bigvee_c \lceil Z=c\rceil \text{ and } \lceil Y_1=b\rceil \wedge \lceil Y_2=b\rceil \Rightarrow \lceil Y_1=Y_2\rceil
\end{aligned}
$$

[9] One is determinism, meaning some CPT entries have zero probability. Another is CSI (context-specific independence (Boutilier et al. 1996)) which is value-wise conditional independence.

[10] Usually log-linear models are written as $P(\boldsymbol{x}) = Z^{-1}\exp(\sum_{i=1}^{M} w_i f_i(\boldsymbol{x}_i))$ which is always positive. We however prefer a more general form $P(\boldsymbol{x}) = Z^{-1}\prod_{i=1}^{M}\exp(w_i f_i(\boldsymbol{x}_i)) = \prod_{i=1}^{M} F_i(\boldsymbol{x}_i)$ and allow $F_i(\boldsymbol{x}_i)$ to be zero at some $\boldsymbol{x}_i$.

are tautologies where

$$\begin{aligned}\lceil Y_1 = Y_2 \rceil &= \bigvee_b (\lceil Y_1 = b \rceil \wedge \lceil Y_2 = b \rceil) \\ &= P_c(\lceil X = a \rceil, \lceil Y_1 = b \rceil, \lceil Y_2 = b \rceil, \lceil Z = c \rceil \mid \lceil Y_1 = Y_2 \rceil \wedge KB_1 \wedge KB_2) \\ &= P_c(\lceil X = a \rceil, \lceil Y_1 = b \rceil, \lceil Z = c \rceil \mid \lceil Y_1 = Y_2 \rceil \wedge KB_1 \wedge KB_2).\end{aligned}$$

Thus $P(X = a, Y = b, Z = c) \propto F_1(a, b)F_2(b, c)$ is expressed as a CBPM. The point of this transformation is the introduction of connection variables Y_1 and Y_2 and the addition of an *equality constraint* $\lceil Y_1 = Y_2 \rceil$ on them to ensure that they take a compatible value b appearing in $F_1(a, b)$ and $F_2(b, c)$.

Generalizing this example is straightforward. To state Theorem 2 below, we make our terminology precise. Let $P(\boldsymbol{X} = \boldsymbol{x}) \propto \prod_{i=1}^{M} F_i(\boldsymbol{x}_i)$ be a given distribution. If z, a value of a random variable $Z \in \boldsymbol{X}$, is shared by two or more potential functions $F_{i_1}(\boldsymbol{x}_{i_1}), \ldots, F_{i_k}(\boldsymbol{x}_{i_k})$ $(k > 1)$, Z is said to be *shared*. For such Z we introduce new variables $Z'_{i_1}, \ldots, Z'_{i_k}$ called *connection variable*s for Z together with a Boolean formula $\bigvee_{z \in V(Z)} \lceil Z'_{i_1} = z \rceil \wedge \cdots \wedge \lceil Z'_{i_k} = z \rceil$ called the *equality constraint associated with* Z. Now we have (proof omitted)

Theorem 2 *Suppose* $P(\boldsymbol{X} = \boldsymbol{x}) = Z^{-1} \prod_{i=1}^{M} F_i(\boldsymbol{x}_i)$ $(\bigcup_{i=1}^{M} \boldsymbol{x}_i = \boldsymbol{x})$ *is given. Then* $P(\boldsymbol{X} = \boldsymbol{x})$ *has an equivalent CBPM* $P_c(\lceil \boldsymbol{X}'_1 = \boldsymbol{x}_1 \rceil, \ldots, \lceil \boldsymbol{X}'_M = \boldsymbol{x}_M \rceil \mid C \wedge KB)$ *with the same factorization as* P:

$$\begin{aligned}P(\boldsymbol{X} = \boldsymbol{x}) &= P_c(\lceil \boldsymbol{X}'_1 = \boldsymbol{x}_1 \rceil, \ldots, \lceil \boldsymbol{X}'_M = \boldsymbol{x}_M \rceil \mid C \wedge KB) \\ &= \frac{\prod_{i=1}^{M} P_c(\lceil \boldsymbol{X}'_i = \boldsymbol{x}_i \rceil \mid KB_i)}{\sum_{\boldsymbol{x}} \prod_{i=1}^{M} P_c(\lceil \boldsymbol{X}'_i = \boldsymbol{x}_i \rceil \mid KB_i)} \qquad (2)\end{aligned}$$

where C *is a conjunction of the equality constraints associated with shared variables in* $\boldsymbol{X}$, $P_c(\lceil \boldsymbol{X}'_i = \boldsymbol{x}_i \rceil \mid KB_i)$ *is a CBPM equivalent to the factor distribution* $Q^{(i)}(\boldsymbol{X}'_i = \boldsymbol{x}_i) = \frac{F_i(\boldsymbol{x}_i)}{\sum_{\boldsymbol{x}_i} F_i(\boldsymbol{x}_i)}$ $(1 \le i \le M)$ *and* $KB = \bigwedge_{i=1}^{M} KB_i$.

Here $\lceil \cdot \rceil$ is extended to vector equations in an obvious way. The denominator in (2) is the probability that independent sampling from factor distributions $Q^{(i)}(\boldsymbol{X}'_i = \boldsymbol{x}_i)$ $(1 \le i \le M)$ returns "compatible" values $\boldsymbol{x}_i$ such that $\bigcup_{i=1}^{M} \boldsymbol{x}_i = \boldsymbol{x}$ for some $\boldsymbol{x}$.

Theorem 2 gives us a way of viewing log-linear models such as CRFs and MLNs as conditional distributions, which leads to a new parameter learning algorithm presented in Sect. 4.

2.4 Generative models and CBPMs

We next consider rule-based generative models such as PCFGs. To show that CBPMs can deal with them, we use PRISM which is a symbolic-statistical modeling language based on Prolog extended with a built-in predicate `msw/3` representing probabilistic choices (Sato and Kameya 2001, 2008). PRISM covers generative models in general and PCFGs in particular as exemplified in (Sato and Kameya 2001). We first review PRISM's semantics (*distribution semantics*) for self-containedness.

A PRISM program $DB = R \cup F$ consists of a set R of definite clauses and a set F of ground `msw` atoms. No clause in R contains the `msw` predicate in the head. We give a *base*

distribution $P_{\mathtt{msw}}(\cdot)$ over the Herbrand interpretations of F in such a way that $\mathtt{msw}(i,t,v)$ represents a probabilistic choice i from exclusive alternatives V_i at trial t returning a value v ($\in V_i$) where i, t and v are arbitrary ground terms. So when $\{\mathtt{msw}(i,t,v) \mid v \in V_i\}$ are drawn from $P_{\mathtt{msw}}(\cdot)$, exactly one of them becomes true for each i,t.

$P_{\mathtt{msw}}(\cdot)$ is extended to $P_{DB}(\cdot)$, a probability measure on the set of possible Herbrand interpretations for DB, by way of the least model semantics (Lloyd 1984) combined with Kolmogorov's extension theorem as in (Sato and Kameya 2001). $P_{DB}(\cdot)$ turns every closed formula φ into a binary random variable having a probability $P_{DB}(\varphi)$. What is important in the distribution semantics is that $\mathrm{iff}(R)$, the if-and-only-if completion of R (Lloyd 1984), has probability one, i.e. $P_{DB}(\mathrm{iff}(R)) = 1$.

When $\mathcal{B}_{\mathcal{H}}$ is finite as assumed here, $\mathrm{iff}(R)$ is given as a Boolean formula $\mathrm{iff}^g(R) \stackrel{\mathrm{def}}{=} \bigwedge_H \mathrm{iff}^g(H)$ where $\mathrm{iff}^g(H) = H \Leftrightarrow B_1 \vee \cdots \vee B_M$ ($M \geq 0$) is a formula such that H is a ground atom which is not an $\mathtt{msw}$ atom and $\{H \Leftarrow B_1, \ldots, H \Leftarrow B_M\}$ is the set of ground clauses from R having H in the head. When $M = 0$, $\mathrm{iff}^g(H) = \neg H$. Suppose G is a non-$\mathtt{msw}$ ground atom. An *explanation* for G is a conjunction E of $\mathtt{msw}$ atoms in F such that $E \wedge R \vdash G$. We say G has a *disjunctive explanation* $E_1 \vee \cdots \vee E_k$ if there are explanations $E_1, \ldots, E_k$ for G satisfying $\mathrm{iff}^g(R) \vdash G \Leftrightarrow E_1 \vee \cdots \vee E_k$.[11] Since $P_{DB}(G) = P_{\mathtt{msw}}(E_1 \vee \cdots \vee E_k)$ holds thanks to $P_{DB}(\mathrm{iff}^g(R)) = 1$ by the distribution semantics, we compute $P_{DB}(G)$ by logically reducing G to its disjunctive explanation using $\mathrm{iff}^g(R)$. It is however generally hard to tell when G has a disjunctive explanation. To state a sufficient condition, we introduce a binary relation "$\succ$" over $\mathcal{B}_{\mathcal{H}}$ by $A \succ B$ if-and-only-if B appears in the body W of some ground clause $A \Leftarrow W$ from DB. DB is said to be *cycle-free* if there is no looping chain $A_1 \succ A_2 \succ \cdots \succ A_1$. Then it is rather easy to see that if DB is cycle-free and $\mathcal{B}_{\mathcal{H}}$ is finite, G has a disjunctive explanation such that $\mathrm{iff}^g(R) \vdash G \Leftrightarrow E_1 \vee \cdots \vee E_k$.

Now let $A_1, \ldots, A_N$ be an enumeration of non-$\mathtt{msw}$ atoms in $\mathcal{B}_{\mathcal{H}}$ ($\mathcal{B}_{\mathcal{H}}$ is finite). Construct a product P_1 of Bernoulli distributions over $A_1, \ldots, A_N$ such that $P_1(A_1) = \cdots = P_1(A_N) = 1/2$. Then construct a CBPM P_2 for $\mathtt{msw}$ atoms following the proof of Theorem 1 such that $P_2(\cdot \mid XOR_{\mathtt{msw}}) = P_{\mathtt{msw}}(\cdot)$ holds where $XOR_{\mathtt{msw}}$ is a Boolean formula expressing the mutual exclusiveness and exhaustiveness of $\mathtt{msw}$ atoms. Finally define P_c as the product distribution of P_1 and P_2. Note that the A_i's and the $\mathtt{msw}$ atoms are independent w.r.t. P_c.

Lemma 1 *Suppose $\mathcal{B}_{\mathcal{H}}$ is finite and DB is cycle-free. Let Δ be a Boolean formula consisting of* $\mathtt{msw}$ *atoms. We have $P_c(\mathit{iff}^g(R) \wedge \Delta) = (1/2)^N P_2(\Delta)$ for some N.*

Proof Write $\mathrm{iff}^g(R) = \bigwedge_{i=1}^{N}(A_i \Leftrightarrow W_i)$. Take the transitive closure of $\succ$ and extend it to a total ordering "$\succ^*$" (possible because $\mathcal{B}_{\mathcal{H}}$ is finite and DB is cycle-free). Without loss of generality, we may assume $A_1 \succ^* \cdots \succ^* A_N$. Note that A_1 does not occur in W_1 or in any other $A_i \Leftrightarrow W_i$ ($i \geq 2$) because A_1 is the highest atom in the "$\succ^*$" ordering. So A_1 is independent of W_1, $A_i \Leftrightarrow W_i$ ($i \geq 2$) and Δ w.r.t. P_c. Put $\Delta' = (\bigwedge_{i=2}^{N} A_i \Leftrightarrow W_i) \wedge \Delta$. We have

$$\begin{aligned} P_c(\mathrm{iff}^g(R) \wedge \Delta) &= P_c((A_1 \Leftrightarrow W_1) \wedge \Delta') \\ &= P_c(A_1 \wedge W_1 \wedge \Delta') + P_c(\neg A_1 \wedge \neg W_1 \wedge \Delta') \\ &= 1/2 P_c(W_1 \wedge \Delta') + 1/2 P_c(\neg W_1 \wedge \Delta') \end{aligned}$$

[11] Intuitively G is a sentence, explanations $E_1, \ldots, E_k$ are sentence derivations and $\mathrm{iff}^g(R)$ is the set of CFG rules used to parse the sentence.

$$= 1/2P_c((W_1 \vee \neg W_1) \wedge \Delta') = 1/2P_c(\Delta')$$
$$= \cdots = (1/2)^N P_c(\Delta) = (1/2)^N P_2(\Delta). \qquad \square$$

Theorem 3 *Suppose $\mathcal{B}_\mathcal{H}$ is finite and a PRISM program DB is cycle-free. Then DB has an equivalent CBPM such that for a non-*msw *ground atom G, $P_{DB}(G) = P_c(G \mid \mathit{iff}^g(R) \wedge XOR_{\mathtt{msw}})$ where $P_{DB}(G)$ is the probability of G defined by DB.*

Proof Let $E_1 \vee \cdots \vee E_k$ be a disjunctive explanation for G. $P_{DB}(G) = P_{\mathtt{msw}}(E_1 \vee \cdots \vee E_k)$ holds. Since $\mathrm{iff}^g(R) \vdash G \Leftrightarrow E_1 \vee \cdots \vee E_k$, we have, applying Lemma 1,

$$\begin{aligned}
P_c(G \mid \mathrm{iff}^g(R) \wedge XOR_{\mathtt{msw}}) &= P_c(E_1 \vee \cdots \vee E_k \mid \mathrm{iff}^g(R) \wedge XOR_{\mathtt{msw}}) \\
&= \frac{P_c((E_1 \vee \cdots \vee E_k) \wedge \mathrm{iff}^g(R) \wedge XOR_{\mathtt{msw}})}{P_c(\mathrm{iff}^g(R) \wedge XOR_{\mathtt{msw}})} \\
&= \frac{(1/2)^N P_2((E_1 \vee \cdots \vee E_k) \wedge XOR_{\mathtt{msw}})}{(1/2)^N P_2(XOR_{\mathtt{msw}})} \\
&= P_2(E_1 \vee \cdots \vee E_k \mid XOR_{\mathtt{msw}}) = P_{\mathtt{msw}}(E_1 \vee \cdots \vee E_k) \\
&= P_{DB}(G). \qquad \square
\end{aligned}$$

Theorem 3 assumes $\mathcal{B}_\mathcal{H}$ is finite, or equally the number of propositional variables (ground atoms) is finite. As a result Theorem 3 is applicable to BNs but not to PCFGs which require infinitely many random variables. This is theoretically correct but we do not use infinitely many random variables in practice. Think of a PCFG in Chomsky normal form. The backbone CFG rules expressed as a PRISM program take the form $A(i, j) \Leftarrow B(i, j) \wedge C(j, k)$ and $A(i, i+1) \Leftarrow \mathit{word}(i, i+1)$ where i, j, k are position indexes (Sato and Kameya 2001). When we parse a corpus by the PCFG, only finitely many rules are used because the corpus is finite. Accordingly Theorem 3 is applicable to this finite fragment of the original PCFG, which is practically enough to compute and learn probabilities from the corpus.

Theorem 1 shows that CBPMs are general. Theorem 2 and Theorem 3 show that they are not only able to represent but to "simulate"[12] a large class of discrete probabilistic models despite their conceptual simplicity. In addition, probabilities can be efficiently computed based on BDDs (binary decision diagrams) as explained in Sect. 4. In Sect. 5, we exploit the generality of CBPMs to build probabilistic models for abductive reasoning. We conclude this section with logical properties of CBPMs in finite domains.

Proposition 1 *Let $P_c(\cdot \mid KB)$ be a CBPM conditioned on a Boolean formula KB. Suppose $P_c(\cdot)$ gives every Herbrand model of KB a positive probability.*[13] *Then for any Boolean formula φ, it holds that $P_c(\varphi \mid KB) = 1$ if-and-only-if $KB \vdash \varphi$. It also holds that $P_c(\varphi \mid KB) = P_c(\varphi \mid KB')$ if $\vdash KB \Leftrightarrow KB'$.*

Proof Since $KB \vdash \varphi$ implies $P_c(\varphi \mid KB) = 1$ is evident, we prove the other way around. Suppose $KB \nvdash \varphi$. So $KB \cup \{\neg\varphi\}$ has a Herbrand model ω_0 such that $\omega_0 \models KB \wedge \neg\varphi$.

[12] By "simulate" we mean the preservation of factorization in the case of log-linear models shown in Theorem 2 and goal-subgoal simulation by $\mathrm{iff}^g(R)$ of SLD derivation for logic programs in the case of rule-based approaches shown in Theorem 3.

[13] For example, $0 < P_c(A) < 1$ for every ground atom A is enough.

By assumption $P_c(\{\omega_0\}) > 0$. It follows that $P_c(KB \wedge \neg\varphi) = P_c(\{\omega \mid \omega \models KB \wedge \neg\varphi\}) \geq P_c(\{\omega_0\}) > 0$. Hence $P_c(\varphi \mid KB) = 1 - P_c(\neg\varphi \mid KB) < 1$. Also suppose $\vdash KB \Leftrightarrow KB'$. Then

$$P_c(\varphi \mid KB) = \frac{P_c(\varphi \wedge KB)}{P_c(KB)} = \frac{P_c(\varphi \wedge KB')}{P_c(KB')} = P_c(\varphi \mid KB'). \qquad \square$$

Thus in finite domains, we may replace one KB with another when they are logically equivalent. Also $P_c(\varphi \mid KB) = 1$ coincides with the notion of logical consequence $KB \models \varphi$. However in infinite domains, the latter is false as we see next.

3 CBPMs in infinite domains

3.1 Conditional probability

So far we have only been dealing with CBPMs constructed from finitely many Boolean variables (ground atoms) whereas infinite domains such as natural numbers are excluded. In this section, we investigate the differences between CBPMs in finite domains and those in infinite domains. We assume $\mathcal{L}$, the first-order language we use, has function symbols. Then the Herbrand universe $\mathcal{U}_{\mathcal{H}}$ and the Herbrand base $\mathcal{B}_{\mathcal{H}}$ are countably infinite, whereas $\mathcal{I}_{\mathcal{H}}$, the set of Herbrand interpretations for $\mathcal{B}_{\mathcal{H}}$, has as many elements as real numbers and the probability of each interpretation is infinitesimal or zero, numerically speaking. Also universally quantified formulas are likely to have probability zero when $\mathcal{U}_{\mathcal{H}}$ is infinite.[14] Hence if KB contains non-ground clauses, it is impossible in general, as $P_c(KB) = 0$, to compute the conditional probability of a closed formula φ by $P_c(\varphi \mid KB) = \frac{P_c(\varphi \wedge KB)}{P_c(KB)}$.

Nonetheless, $P_c(\varphi = y \mid KB = x)$ $(x, y \in \{0, 1\})$ is measure-theoretically definable as the Radon-Nikodym derivative (Feller 1971). The problem is that this measure-theoretic $P_c(\varphi = y \mid KB = x)$ is not unique at $KB = 1$ because $P_c(KB = 1) = 0$. In other words, we have to choose, or have to construct an appropriate probability measure as $P_c(\cdot \mid KB = 1)$, hopefully as an extension of the finite case. In particular we require $P_c(\varphi = 1 \mid KB = 1) = 1$ hold if $KB \vdash \varphi$. In what follows, we construct such $P_c(\cdot \mid KB = 1)$ by considering an infinite sequence of joint distributions $P_c(X_1 = x_1, \ldots, X_k = x_k \mid \phi_1, \ldots, \phi_n)$ $(n > 0, k > 0)$ and their limit, where $\phi_1, \ldots, \phi_n$ are ground clauses from KB. Before going into details, we look at an example to get a feeling for the infinite case.

Suppose $\mathcal{L}$ contains a unary predicate symbol $q/1$, a constant symbol 0 and a unary function symbol $s/1$. Then the Herbrand base $\mathcal{B}_{\mathcal{H}}{}'$ is $\{q(0), q(s(0)), \ldots\}$. For the sake of brevity, we write $\overbrace{s(\cdots s}^{i}(0)\cdots)$ as i and the successor of i $(i \geq 0)$ as $i+1$. As stated in Sect. 2, a Herbrand interpretation for $\mathcal{B}_{\mathcal{H}}{}'$ is identified with an infinite 0-1 vector $(x_0, x_1, \ldots)$ which specifies the truth value of $q(i)$ as x_i $(i \geq 0,\ x_i \in \{0, 1\})$. Let ${P'}_c(\cdot)$ be a product probability measure satisfying (by abuse of notation) ${P'}_c(q(0) = x_0, q(1) = x_1, \ldots) = \prod_{i=0}^{\infty} {P'}_c(q(i) = x_i)$ where ${P'}_c(q(i) = x_i) = 1/2$ $(i \geq 0, x_i \in \{0, 1\})$.

Put $\varphi \stackrel{\text{def}}{=} \forall x\,(q(x) \Rightarrow q(x+1))$ and define φ_n $(n \geq 0)$ by

$$\varphi_n \stackrel{\text{def}}{=} \forall x < n\,(q(x) \Rightarrow q(x+1))$$

[14] If for example $P_c(q(n)) < \alpha < 1$ for every natural number n, $P_c(\forall x\, q(x)) = \lim_k \alpha^k = 0$.

$$= (q(0) \Rightarrow q(1)) \wedge \cdots \wedge (q(n-1) \Rightarrow q(n)).$$

We find, by calculation,

$$P'_c(\varphi_n) = \left(\frac{1}{2}\right)^{n+1}(n+2),$$

$$P'_c((q(0) \vee \cdots \vee q(k)) \wedge \varphi_n) = \left(\frac{1}{2}\right)^{n+1}(k+1) \quad (n \geq k+1),$$

$$P'_c(\exists x\, q(x) \wedge \varphi_n) = \left(\frac{1}{2}\right)^{n+1}(n+2) \quad \text{(because } P'_c(\exists x\, q(x)) = 1).$$

Accordingly we have

$$\begin{aligned} P'_c(\varphi) &= P'_c(\forall x\,(q(x) \Rightarrow q(x+1))) \\ &= \lim_{n\to\infty} P'_c(\varphi_n) \\ &= \lim_{n\to\infty}\left(\frac{1}{2}\right)^{n+1}(n+2) = 0, \\ P'_c(q(0) \vee \cdots \vee q(k) \mid \varphi) &\stackrel{\text{def}}{=} \lim_{n\to\infty} P'_c(q(0) \vee \cdots \vee q(k) \mid \varphi_n) \\ &= \lim_{n\to\infty} \frac{k+1}{n+2} = 0, \\ P'_c(\exists x\, q(x) \mid \varphi) &\stackrel{\text{def}}{=} \lim_{n\to\infty} P'_c(\exists x\, q(x) \mid \varphi_n) \\ &= 1. \end{aligned}$$

We note that although $P'_c(\varphi) = 0$, conditional probabilities $P'_c(q(0) \vee \cdots \vee q(k) \mid \varphi)$ and $P'_c(\exists x\, q(x) \mid \varphi)$ have definite values, 0 and 1 respectively, if they are computed as limits of $P'_c(\cdot \mid \varphi_n)$ using successive approximations φ_n to φ. However we also note that $1 = P'_c(\exists x\, q(x) \mid \varphi) \neq \lim_{k\to\infty} P'_c(q(0) \vee \cdots \vee q(k) \mid \varphi) = 0$. This means our limiting procedure which assigns probabilities to all formulas independently all at once does not yield a countably additive probability measure. So we next construct a conditional probability measure by repeating a limiting procedure and defining increasingly larger joint distributions to which Kolmogorov's extension theorem applies.

3.2 Infinite domains

Suppose $\mathcal{L}$ has function symbols. So the Herbrand universe $\mathcal{U}_\mathcal{H}$ and the Herbrand base $\mathcal{B}_\mathcal{H}$ are countably infinite. Let $A_1, A_2, \ldots$ be an enumeration of atoms in $\mathcal{B}_\mathcal{H}$. $\mathcal{I}_\mathcal{H}$, the set of all Herbrand interpretations for $\mathcal{B}_\mathcal{H}$, is written as $\mathcal{I}_\mathcal{H} = \prod_{i=0}^{\infty}\{0,1\}_i$ where $\{0,1\}_i$ represents the truth values of A_i. We consider each $\{0,1\}_i$ as a probability space with the discrete topology and give the product topology to $\mathcal{I}_\mathcal{H}$. We construct a probability space $(\mathcal{I}_\mathcal{H}, \mathcal{F}, P_c)$ where $P_c(\cdot)$ is an infinite product measure on $\mathcal{F}$ which is the smallest σ-algebra containing all open sets of $\mathcal{I}_\mathcal{H}$ (Feller 1971). $P_c(A_1 = x_1, A_2 = x_2, \ldots, A_n = x_n) = \prod_{i=1}^{n} P_c(A_i = x_i)$ holds for every $n > 0$. Let $\phi_1, \phi_2, \ldots$ be an enumeration of ground clauses from KB. In the Herbrand universe, KB is logically equivalent to $\bigwedge_{i>0}\phi_i$.

Define an infinite sequence $(n_{0,i})_{i>0}$ by

$$n_{0,i} = i \ \ (i > 0).$$

Now we are going to inductively construct a subsequence $(n_{k,i})_{i>0}$ of $(n_{0,i})_{i>0}$ for each k ($k \geq 0$) so that eventually $(n_{i,i})_{i>0}$, a subsequence of every subsequence, makes a distribution sequence $P_c(A_1 = x_1, \ldots, A_k = x_k \mid \phi_1, \ldots, \phi_{n_{i,i}})$ $(i = 1, 2, \ldots)$ convergent for all $k > 0$ and the x_j's.

Suppose $(n_{k-1,i})_{i>0}$, a subsequence of $(n_{0,i})_{i>0}$, has been defined for k $(k > 0)$ and $\lim_{i\to\infty} P_c(A_1 = x_1, \ldots, A_{k-1} = x_{k-1} \mid \phi_1, \ldots, \phi_{n_{k-1,i}})$[15] exists for every $x_1, \ldots, x_{k-1} \in \{0, 1\}$ if $k > 1$. Choose a subsequence $(n_{k,i})_{i>0}$ of $(n_{k-1,i})_{i>0}$ such that

$$P_{k,\infty}(A_1 = x_1, \ldots, A_k = x_k) \stackrel{\text{def}}{=} \lim_{i\to\infty} P_c(A_1 = x_1, \ldots, A_k = x_k \mid \phi_1, \ldots, \phi_{n_{k,i}})$$

exists for every $x_1, \ldots, x_k \in \{0, 1\}$.[16] Since $(n_{k,i})_{i>0}$ is a subsequence of $(n_{k-1,i})_{i>0}$, we have

$$\begin{aligned}
&\sum_{x_k} \lim_{i\to\infty} P_c(A_1 = x_1, \ldots, A_k = x_k \mid \phi_1, \ldots, \phi_{n_{k,i}}) \\
&\quad = \sum_{x_k} \lim_{i\to\infty} P_c(A_1 = x_1, \ldots, A_k = x_k \mid \phi_1, \ldots, \phi_{n_{k-1,i}}) \\
&\quad = \lim_{i\to\infty} \sum_{x_k} P_c(A_1 = x_1, \ldots, A_k = x_k \mid \phi_1, \ldots, \phi_{n_{k-1,i}}) \\
&\quad = \lim_{i\to\infty} P_c(A_1 = x_1, \ldots, A_{k-1} = x_{k-1} \mid \phi_1, \ldots, \phi_{n_{k-1,i}}) \\
&\quad = P_{k-1,\infty}(A_1 = x_1, \ldots, A_{k-1} = x_{k-1}). \qquad (3)
\end{aligned}$$

Repeat this process and define $(n_{k-1,i})_{i>0}$ for all k $(k > 0)$. Then consider $(n_{i,i})_{i>0}$. $(n_{i,i})_{i\geq k}$, the sequence with the initial $k-1$ elements removed from $(n_{i,i})_{i>0}$, is a subsequence of $(n_{k,i})_{i>0}$ for every k $(k > 0)$. Therefore,

$$\begin{aligned}
&\lim_{i\to\infty} P_c(A_1 = x_1, \ldots, A_k = x_k \mid \phi_1, \ldots, \phi_{n_{i,i}}) \\
&\quad = \lim_{i\to\infty} P_c(A_1 = x_1, \ldots, A_k = x_k \mid \phi_1, \ldots, \phi_{n_{k,i}}) \\
&\quad = P_{k,\infty}(A_1 = x_1, \ldots, A_k = x_k) \qquad (4)
\end{aligned}$$

holds for every $k > 0$ and $x_1, \ldots, x_k \in \{0, 1\}$. Also by construction, it holds that

$$\begin{aligned}
&\sum_{x_k} P_{k,\infty}(A_1 = x_1, \ldots, A_k = x_k) \\
&\quad = \sum_{x_k} \lim_{i\to\infty} P_c(A_1 = x_1, \ldots, A_k = x_k \mid \phi_1, \ldots, \phi_{n_{k,i}})
\end{aligned}$$

[15] In this section, a formula ϕ on the conditioning part always means $\phi = 1$. That is, $P_c(\cdot \mid \phi)$ means $P_c(\cdot \mid \phi = 1)$.

[16] We can always choose such a convergent subsequence as there are only finitely many $(x_1, \ldots, x_{k-1})$s and a countably infinite set $\{P_c(A_1 = x_1, \ldots, A_{k-1} = x_{k-1} \mid \phi_1, \ldots, \phi_{n_{k-1,i}}) \mid x_j \in \{1, 0\}, i > 0\}$ has a cluster point in $[0, 1]$.

$$= P_{k-1,\infty}(A_1 = x_1, \ldots, A_{k-1} = x_{k-1}) \quad \text{by (3)}.$$

So we can apply Kolmogorov's extension theorem (Chow and Teicher 1997) to the set of "consistent" joint distributions $\{P_{k,\infty}(A_1 = x_1, \ldots, A_k = x_k) \mid k > 0\}$ and conclude that a probability measure denoted by $P_c^\infty(\cdot \mid \bigwedge_{i>0} \phi_i)$ on $\mathcal{I}_\mathcal{H}$ having $P_{k,\infty}(A_1 = x_1, \ldots, A_k = x_k)$s as marginal distributions exists:

$$\begin{aligned} & P_c^\infty\left(A_1 = x_1, \ldots, A_k = x_k \,\middle|\, \bigwedge_{i>0} \phi_i\right) \\ &= P_{k,\infty}(A_1 = x_1, \ldots, A_k = x_k) \\ &= \lim_{i\to\infty} P_c(A_1 = x_1, \ldots, A_k = x_k \mid \phi_1, \ldots, \phi_{n_{i,i}}) \quad \text{by (4)}. \end{aligned}$$

We summarize the argument so far as

Theorem 4 *Suppose KB is a set of countably many clauses in a first-order language $\mathcal{L}$ which may contain function symbols. Let $\mathcal{I}_\mathcal{H}$ be the Herbrand interpretations for $\mathcal{L}$ and $P_c(\cdot)$ a probability measure on $\mathcal{I}_\mathcal{H}$ such that $P_c(A_1 = x_1, \ldots, A_k = x_k) = \prod_{i=1}^k P_c(A_i = x_i)$ $(x_i \in \{0, 1\})$ for every k $(k > 0)$ where $A_1, A_2, \ldots$ is an enumeration of ground atoms in $\mathcal{L}$. Let $\phi_1, \phi_2, \ldots$ be an enumeration of ground clauses from KB. We assume $P_c(\phi_1 \wedge \cdots \wedge \phi_m) > 0$ for every $m > 0$.*[17]

Then there exist an infinite sequence $(n_{i,i})_{i>0}$ and a probability measure on $\mathcal{I}_\mathcal{H}$ denoted by $P_c^\infty(\cdot \mid \bigwedge_{i>0} \phi_i)$ or equivalently by $P_c^\infty(\cdot \mid KB)$ such that $\lim_{i\to\infty} P_c(A_1 = x_1, \ldots, A_k = x_k \mid \phi_1, \ldots, \phi_{n_{i,i}})$ converges to a joint distribution $P_c^\infty(A_1 = x_1, \ldots, A_k = x_k \mid KB)$ for every $k > 0$. Moreover, if $KB \vdash \phi$ holds for an existentially closed formula *ϕ,*[18] *we have $P_c^\infty(\phi \mid KB) = 1$.*

Proof We have only to prove the latter part. Suppose $KB \vdash \phi$ and ϕ is existentially closed. We consider the universally quantified formula $\neg\phi$ as a set of clauses.

Since $KB \cup \{\neg\phi\}$ is inconsistent, it follows from Herbrand's theorem that there are ground clauses $\{\phi_1, \ldots, \phi_N\}$ from KB and those $\{\psi_1, \ldots, \psi_M\}$ from $\neg\phi$ such that $\phi_1, \ldots, \phi_N \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_M)$. So $\phi_1, \ldots, \phi_{n_{i,i}} \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_M)$ holds for every i such that $\{\phi_1, \ldots, \phi_{n_{i,i}}\} \supseteq \{\phi_1, \ldots, \phi_N\}$, and hence we have $P_c(\neg(\psi_1 \wedge \cdots \wedge \psi_M) \mid \phi_1, \ldots, \phi_{n_{i,i}}) = 1$ for large i. Therefore, by taking the limit, we conclude $P_c^\infty(\neg(\psi_1 \wedge \cdots \wedge \psi_M) \mid KB) = 1$. On the other hand, since $\neg\phi \cup \{\neg(\psi_1 \wedge \cdots \wedge \psi_N)\}$ has no Herbrand model, we have $\neg(\psi_1 \wedge \cdots \wedge \psi_N) \vdash \phi$. It follows from $P_c^\infty(\neg(\psi_1 \wedge \cdots \wedge \psi_M) \mid KB) = 1$ that $P_c^\infty(\phi \mid KB) = 1$ as well. □

We remark that $P_c^\infty(\cdot \mid KB)$ depends on $(n_{i,i})_{i>0}$ and may not be unique. A simple uniqueness condition is as follows. Let $\phi_1, \phi_2, \ldots$ be an enumeration of ground clauses from KB. Suppose $P_c(KB) = \lim_{i\to\infty} P_c(\phi_1 \wedge \cdots \wedge \phi_i) > 0$ exists.[19] Then $P_c^\infty(\cdot \mid KB)$ is unique and coincides with $P_c(\cdot \mid KB)$. This is because $P_c(\varphi \mid KB) = \lim_{i\to\infty} P_c(\varphi \mid \phi_1 \wedge \cdots \wedge \phi_i) = \lim_{i\to\infty} P_c(\varphi \mid \phi_1 \wedge \cdots \wedge \phi_{n_{i,i}}) = P_c^\infty(\varphi \mid KB)$[20] holds for any Boolean formula φ. When

[17] This is satisfied if $0 < P_c(A_i) < 1$ for all i $(i > 0)$.

[18] A formula is existentially closed if it is closed and takes the form $\exists x_1, \ldots, x_n \varphi$ where φ has no quantifier.

[19] Apparently the limit does not depend on the choice of enumeration.

[20] Any convergent subsequence of a convergent sequence has the same limit.

$P_c(KB) = 0$ however, the uniqueness condition is still an open question. We need to check the uniqueness of $P_c^\infty(\cdot \mid KB)$ for each case at the moment.

Unlike Proposition 1, the inverse of Theorem 4, i.e. $P_c^\infty(\phi \mid KB) = 1$ implies $KB \vdash \phi$, does not necessarily hold in infinite domains. Here is a counter example. Let $\varphi = \forall x\,(q(x) \Rightarrow q(x+1))$, $\varphi_n = (q(0) \Rightarrow q(1)) \wedge \cdots \wedge (q(n-1) \Rightarrow q(n))$ and ${P'}_c(\cdot)$ be as in the previous subsection. We see that $\lim_{n\to\infty} {P'}_c(q(0) = x_0, \ldots, q(k) = x_k \mid \varphi_n)$ always converges for any $x_0, \ldots, x_k$ and the resulting ${P'}_c^\infty(\cdot \mid \varphi)$ is unique regardless of the choice of $(n_{i,i})_{i>0}$. Thus $(\varphi_n)_{n\geq 0}$ defines an infinite CBPM ${P'}_c^\infty(\cdot \mid \varphi)$. We have ${P'}_c^\infty(\neg q(0) \mid \varphi) = 1$ because ${P'}_c^\infty(q(0) \mid \varphi) = \lim_{n\to\infty} {P'}_c(q(0) \mid \varphi_n) = 0$. We on the other hand have $\varphi \nvdash \neg q(0)$ because φ has a Herbrand model that makes every ground atom true. So ${P'}_c^\infty(\neg q(0) \mid \varphi) = 1$ but not $\varphi \nvdash \neg q(0)$.

From the next section on, we get back to finite domains and consider probability computation and parameter learning in finite domains.

4 Parameter learning

In this section we assume KB is a Boolean formula (finite set of ground clauses) and derive the *EMC algorithm*, an EM algorithm for CBPMs which statistically estimates parameters from data. It is a descendent of the FAM algorithm for SLPs (Cussens 2001) such that conditioning by a "success" proposition in FAM is generalized to conditioning by a knowledge base KB.

4.1 Probability computation by BDDs

Before deriving the EMC algorithm, we describe how probabilities of probabilistic Boolean formulas are computed by BDDs (binary decision diagrams) (Akers 1978; Bryant 1986).[21] The description is sketchy due to space limitations.

Suppose a probabilistic Boolean formula φ consisting of independent Boolean variables is given. To compute $P_c(\varphi)$, we convert φ to a BDD which is a directed acyclic graph representing a disjunctive normal form of φ such that disjuncts are mutually exclusive. In the graph, each non-terminal node N is labeled by a Boolean variable, A, and has two outgoing edges to its children, 1-edge and 0-edge, respectively representing the assignments $A =$ true and $A =$ false. There are two types of terminal nodes, 1-terminal and 0-terminal. Each path from the root node to a 1-terminal represents an assignment for the Boolean variables in φ making φ true, thereby corresponding to a disjunct in a disjunctive normal form $\bigvee_{i=1}^{n}(L_{i,1} \wedge \cdots \wedge L_{i,k_i})$ for φ. Let BDD_φ be a BDD for φ. BDD_φ is unique as a graph once a variable ordering in φ is fixed. The size of BDD_φ heavily depends on the variable ordering and finding the best ordering is NP-hard.

$P_c(\varphi)$ is naively computed as

$$P_c\left(\bigvee_{i=1}^{n}\bigwedge_{j=1}^{k_i} L_{i,j}\right) = \sum_{i=1}^{n}\prod_{j=1}^{k_i} P_c(L_{i,j})$$

[21] Similar probability computation is possible by ZDDs (zero-suppressed BDDs) (Minato 2001). For the full details of probability computation by BDDs and ZDDs, see (Ishihata et al. 2008), downloadable at `http://www.cs.titech.ac.jp/~tr/reports/2008/TR08-0004.pdf`.

where for a literal L ($L = A$ or $L = \neg A$ for some atom A), $P_c(L) = P_c(A)$ if $L = A$ else $1 - P_c(A)$. It is possible however to perform the same computation much more efficiently based on BDD_φ by taking advantage of the fact that subgraphs in BDD_φ are shared and hence probability computations for subgraphs can also be shared. So we compute *backward probabilities* for all nodes in BDD_φ by dynamic programming, starting from terminal nodes then upward in which the backward probability for a non-terminal node is computed as the sum of those for its child nodes. $P_c(\varphi)$ is obtained as the backward probability for the root node. It is computed in time proportional to the number of edges in BDD_φ.

Applying the BDD probability computation to parameter estimation by the EM algorithm (Dempster et al. 1977) is more complicated. The backward probability computation described above is just one of the four types of probability computation done in the *BDD-EM algorithm* which is a BDD-based EM algorithm for probabilistic Boolean formulas (Ishihata et al. 2008; Inoue et al. 2009). It uses "decomposed BDDs", hierarchically organized BDDs, and computes by dynamic programming forward probability, inside-probability and outside-probability in addition to backward probability. Using BDDs is not the only one choice though. We may use ZDDs instead of BDDs. Then for example, parameter learning for PCFGs is done in $O(N^3)$ per iteration by the ZDD-EM algorithm where N is the sentence length (Ishihata et al. 2008), exactly the same order as the standard Inside-Outside algorithm for PCFGs. Although details are skipped, the EMC algorithm introduced next employs the same computation techniques as the BDD(ZDD)-EM algorithm to compute required probabilities and expectations.

4.2 Deriving the EMC algorithm

Suppose we have a CBPM $P_c(\cdot \mid KB, \boldsymbol{\theta})$ and let $O_1, \ldots, O_T$ ($T > 0$) be Boolean formulas representing i.i.d. observations. We estimate parameters $\boldsymbol{\theta}$, i.e. probabilities of atoms being true, from $O_1, \ldots, O_T$ by MLE (maximum likelihood estimation) using the following likelihood function.

$$\mathbf{L}(\boldsymbol{\theta}) = \prod_{t=1}^{T} P_c(O_t \mid KB, \boldsymbol{\theta}). \tag{5}$$

Since atoms appearing in KB other than those in $O_1, \ldots, O_T$ are not observed, we use the EM algorithm (Dempster et al. 1977) to find $\boldsymbol{\theta}$ that maximizes $\mathbf{L}(\boldsymbol{\theta})$. Let x be a Herbrand interpretation considered as a 0-1 vector as before and $P_c(x \mid \boldsymbol{\theta})$ the underlying joint distribution that makes atoms in $\mathcal{B}_\mathcal{H}$ independent.

To deal with the existence of i.i.d. atoms in probabilistic models, we assume $\mathcal{B}_\mathcal{H}$ is partitioned into a set $\mathcal{C}$ of sets s of i.i.d. atoms. Every atom $A \in s$ ($s \in \mathcal{C}$) has the same parameters $\theta_{s,1} = P_c(A)$ and $\theta_{s,0} = P_c(\neg A)$. We use $\sigma_{s,v}(x)$ ($s \in \mathcal{C}, v \in \{1, 0\}$) for the count of atoms in s that take on v in the Herbrand interpretation x. We can write $\boldsymbol{\theta} = \{\theta_{s,1}, \theta_{s,0} \mid s \in \mathcal{C}\}$.

Let $P_c(x_t, O_t \mid KB, \boldsymbol{\theta})$ be a conditional joint distribution over Herbrand interpretations x_t and the t-th observation O_t whose marginal distribution is $P_c(O_t \mid KB, \boldsymbol{\theta})$. Since it is not just a joint distribution but a conditional one, we follow (Cussens 2001) and derive the Q function as follows. First introduce a new joint distribution $\widetilde{P}(\cdot)$ below:

$$\begin{aligned}&\widetilde{P}(\langle u^{(1)}, \ldots, u^{(k-1)} \rangle, x_t, O_t \mid KB, \boldsymbol{\theta}) \\ &\quad \stackrel{\text{def}}{=} P_c(u^{(1)}, \neg KB \mid \boldsymbol{\theta}) \cdots P_c(u^{(k-1)}, \neg KB \mid \boldsymbol{\theta}) P_c(x_t, O_t, KB \mid \boldsymbol{\theta})\end{aligned}$$

Step 1:	Initialize parameters $\boldsymbol{\theta}^{(0)}$
Step 2:	Iterate $\boldsymbol{\theta}^{(k)} = \mathrm{argmax}_{\boldsymbol{\theta}}\, Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k-1)})$ until convergence of $\mathbf{L}(\boldsymbol{\theta}^{(k)})$
Step 3:	Return $\boldsymbol{\theta}^{(\infty)} = \boldsymbol{\theta}^{(k)}$ at convergence as estimated values

Fig. 1 The abstract EM algorithm

where the $u^{(j)}$'s are independent Herbrand interpretations. It holds that

$$\sum_{k=1}^{\infty} \sum_{\langle u^{(1)},\ldots,u^{(k-1)}\rangle} \widetilde{P}(\langle u^{(1)},\ldots,u^{(k-1)}\rangle, x_t, O_t \mid KB, \boldsymbol{\theta})$$
$$= \sum_{k=1}^{\infty} \left(\sum_{u} P_c(u, \neg KB \mid \boldsymbol{\theta}) \right)^{k-1} P_c(x_t, O_t, KB \mid \boldsymbol{\theta}) = P_c(x_t, O_t \mid KB, \boldsymbol{\theta}).$$

Accordingly $P_c(O_t \mid KB, \boldsymbol{\theta})$ is a marginal distribution of $\widetilde{P}(\langle u^{(1)},\ldots,u^{(k-1)}\rangle, x_t, O_t \mid KB, \boldsymbol{\theta})$. Hence putting $\overline{u}_t^{k-1} = \langle u^{(1)},\ldots,u^{(k-1)}\rangle$ and assuming $\overline{u}_t^{k-1}, x_t$ as hidden variables in $\widetilde{P}(\overline{u}_t^{k-1}, x_t, O_t \mid KB, \boldsymbol{\theta}')$, we introduce $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ by

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}') \stackrel{\mathrm{def}}{=} \sum_{t=1}^{T} \sum_{k=1}^{\infty} \sum_{x_t} \sum_{\overline{u}_t^{k-1}} \widetilde{P}(\overline{u}_t^{k-1}, x_t \mid O_t, KB, \boldsymbol{\theta}') \ln \widetilde{P}(\overline{u}_t^{k-1}, x_t, O_t \mid KB, \boldsymbol{\theta})$$
$$= \frac{T}{P_c(KB \mid \boldsymbol{\theta}')} \left(\sum_{u:u\models\neg KB} P_c(u \mid \boldsymbol{\theta}') \ln P_c(u \mid \boldsymbol{\theta}) \right)$$
$$+ \sum_{t=1}^{T} \frac{1}{P_c(O_t \wedge KB \mid \boldsymbol{\theta}')} \sum_{x_t : x_t \models O_t \wedge KB} P_c(x_t \mid \boldsymbol{\theta}') \ln P_c(x_t \mid \boldsymbol{\theta}). \tag{6}$$

Using this $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$, the EM algorithm is abstractly described in Fig. 1.

Since $Q(\boldsymbol{\theta}, \boldsymbol{\theta}') \geq Q(\boldsymbol{\theta}', \boldsymbol{\theta}') \Rightarrow \mathbf{L}(\boldsymbol{\theta}) \geq \mathbf{L}(\boldsymbol{\theta}')$ is provable, $\mathbf{L}(\boldsymbol{\theta}^{(k)})$ is guaranteed to increase at every iteration. Note that $\boldsymbol{\theta}^{(\infty)}$ only gives a local maximum of $\mathbf{L}(\boldsymbol{\theta})$, not necessarily the global maximum. By substituting (6) for $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ in Fig. 1, we obtain the *EMC algorithm* (EM algorithm for constraint-based probabilistic models) in Fig. 2.

As explained previously, we adopted BDD(ZDD)-based probability computation when we implemented the EMC algorithm. We call the resulting algorithm the *BDD-EMC algorithm*. Compared to the BDD-EM algorithm (Ishihata et al. 2008) it additionally computes the first term in (7) which is the average number of i.i.d. atoms in s that take on the value v in a Herbrand interpretation falsifying KB. Although BDD(ZDD)-based probability computation techniques may help us but computing $P_c(KB \mid \boldsymbol{\theta})$ is still infeasible when KB is large. It remains as a future research topic for CBPMs.

5 Constraint-based statistical abduction

In this section we apply CBPMs to statistical abduction. We assume domains are finite.

Put $\begin{cases} \mathcal{W}_1 = \{u \mid u \models \neg KB\} \\ \mathcal{W}_2^t = \{x_t \mid x_t \models O_t \wedge KB\} \ (1 \le t \le T) \end{cases}$
where $\mathcal{W}_1$ (resp. $\mathcal{W}_2^t$) is the set of truth assignments for $\mathcal{B}_\mathcal{H}$ which make KB false (resp. $O_t \wedge KB$ true), and repeat the **E-step** and the **M-step** below alternately until convergence.

E-step: Compute the *conditional expectation* $\eta_\theta^v[s]$ of $\sigma_{s,v}$ for $s \in \mathcal{C}$, $v \in \{0, 1\}$ by

$$\eta_\theta^v[s] = \frac{T}{P_c(KB)} \sum_{u \in \mathcal{W}_1} \sigma_{s,v}(u) \prod_{s' \in \mathcal{C}} \prod_{v' \in \{1,0\}} \theta_{s',v'}^{\sigma_{s',v'}(u)} + \sum_{t=1}^{T} \frac{1}{P_c(O_t \wedge KB)} \sum_{x_t \in \mathcal{W}_2^t} \sigma_{s,v}(x_t) \prod_{s' \in \mathcal{C}} \prod_{v' \in \{1,0\}} \theta_{s',v'}^{\sigma_{s',v'}(x_t)} \tag{7}$$

where

$$P_c(\neg KB \mid \boldsymbol{\theta}) = \sum_{u \in \mathcal{W}_1} \prod_{s \in \mathcal{C}} \prod_{v \in \{1,0\}} \theta_{s,v}^{\sigma_{s,v}(u)},$$

$$P_c(KB \mid \boldsymbol{\theta}) = 1 - P_c(\neg KB \mid \boldsymbol{\theta}),$$

$$P_c(O_t \wedge KB \mid \boldsymbol{\theta}) = \sum_{x_t \in \mathcal{W}_2^t} \prod_{s \in \mathcal{C}} \prod_{v \in \{1,0\}} \theta_{s,v}^{\sigma_{s,v}(x_t)}.$$

M-step: Update $\boldsymbol{\theta}$ to $\hat{\boldsymbol{\theta}}$ by

$$\hat{\theta}_{s,v} = \frac{\eta_\theta^v[s]}{\eta_\theta^1[s] + \eta_\theta^0[s]} \quad \text{for every } s \in \mathcal{C}, v \in \{0, 1\}.$$

Fig. 2 The EMC algorithm

5.1 Statistical abduction

Abduction is one of three forms of logical inference (deduction, induction, abduction) that infers the best explanation E for an observation O such that $KB \wedge E \vdash O$ and $KB \wedge E$ is consistent. *Statistical abduction* in addition attempts to quantify explanations with probabilities and select the best explanation as the one having the highest probability, realizing robust abduction applicable to noisy data. The framework of statistical abduction is general. Many known probabilistic models from BNs to PCFGs are understood as performing statistical abduction (Sato and Kameya 2001). There are already a couple of systems for statistical abduction (Poole 1993, 1997; Sato and Kameya 2002). They are common in that O is an atom representing our observation, E is a conjunction made up of particular (probabilistic) atoms called *abducible*s, and KB is a logic program of one kind or another that describes the process of how O is deduced from $KB \wedge E$. One problem with these systems is that to ensure this procedural nature, KB is restricted to definite clause programs (Poole 1993;

Sato and Kameya 2002) or to acyclic logic programs (Poole 1997)[22] that prevents the use of disjunctions and cyclic rules. Observations restricted to atoms is another problem. We may observe negative results $\neg happy(Bill)$ or complex events such as $rich(Bill) \Rightarrow happy(Bill)$. Although these restrictions simplify inference and probability computation, they are unnecessarily restrictive from the viewpoint of knowledge representation.

To solve these problems, we propose *constraint-based statistical abduction* which applies CBPMs to statistical abduction. In the constraint-based statistical abduction, we have a knowledge base KB which is a set of arbitrary clauses, not restricted to Horn clauses, and i.i.d. observations $O_1, \ldots, O_T$ which we assume to be arbitrary Boolean formulas made up of ground atoms. For each O_t $(1 \leq t \leq T)$, we search for an explanation E in the search space $\mathcal{E}$ of possible explanations such that $KB \wedge E \vdash O_t$ and $KB \wedge E$ is consistent. $\mathcal{E}$ is specified beforehand, for instance as a set of conjunctions of abducibles as in PRISM. Let $\{E_1^{(t)}, \ldots, E_{k_t}^{(t)}\}$ be a set of explanations we obtain for O_t.[23] The disjunction $E^{(t)} = E_1^{(t)} \vee \cdots \vee E_{k_t}^{(t)}$ is called a *disjunctive explanation* for O_t following PRISM. We then construct a CBPM $P_c(\cdot \mid KB, \boldsymbol{\theta})$ that specifies a distribution over Herbrand interpretations. Here $\boldsymbol{\theta}$ collectively stands for parameters, i.e. the probabilities of atoms in $\mathcal{B}_{\mathcal{H}}$ being true. We estimate $\boldsymbol{\theta}$ as the maximizer of the following likelihood function $\mathbf{L}^{abd}(\boldsymbol{\theta})$.

$$\mathbf{L}^{abd}(\boldsymbol{\theta}) = \prod_{t=1}^{T} P_c(E^{(t)} \mid KB, \boldsymbol{\theta}). \tag{7}$$

The reason for the choice of this likelihood function is as follows. First as a probabilistic model applied to i.i.d. data $O_1, \ldots, O_T$, we should maximize $\mathbf{L}(\boldsymbol{\theta}) = \prod_{t=1}^{T} P_c(O_t \mid KB, \boldsymbol{\theta})$. On the other hand as we are seeking for true explanations for the O_t's in statistical abduction, the probability of (at least) one of the O_t's explanations $\{E_1^{(t)}, \ldots, E_{k_t}^{(t)}\}$, or equivalently their disjunction $E^{(t)}$ being true, should be high. In other words, we should maximize $\mathbf{L}^{abd}(\boldsymbol{\theta}) = \prod_{t=1}^{T} P_c(E^{(t)} \mid KB, \boldsymbol{\theta})$ as well. Unfortunately parameters that maximize $\mathbf{L}^{abd}(\boldsymbol{\theta})$ may differ from those that maximize $\mathbf{L}(\boldsymbol{\theta})$. We therefore maximize $\prod_{t=1}^{T} P_c(O_t \wedge E^{(t)} \mid KB, \boldsymbol{\theta})$ as a compromise between the two. However in view of $KB \models E^{(t)} \Rightarrow O_t$, $P_c(O_t \wedge E^{(t)} \mid KB, \boldsymbol{\theta})$ is equal to $P_c(E^{(t)} \mid KB, \boldsymbol{\theta})$, thus reaching $\mathbf{L}^{abd}(\boldsymbol{\theta})$ in (7). Moreover the EMC algorithm in Fig. 2 works for any O_ts as long as they are Boolean formulas. So we can use it simply by replacing O_t with $E^{(t)}$ in Fig. 2 to maximize $\mathbf{L}^{abd}(\boldsymbol{\theta})$.

5.2 Learning example

We present here a small learning example for the understanding of constraint-based statistical abduction. It is often observed that smart people are rich and rich people are friends with rich people who are generous. The following KB_{rich} formalizes this observation (free variables are implicitly universally quantified).

$$KB_{rich} = \begin{cases} friend(a, b), \\ friend(b, c), \\ generous(b), \\ friend(x, y) \Leftarrow friend(y, x), \\ rich(x) \Leftrightarrow smart(x) \vee \exists y(friend(x, y) \wedge rich(y) \wedge generous(y)). \end{cases}$$

[22] Later the condition is relaxed to "contingently acyclic logic programs" (Poole 2000).

[23] Note that there may be infinitely many explanations but we assume they are finite.

Table 1 Learned probabilities $P_r(F)$ for ground formulas F

F	Observations		
	$a[30/0]c[30/0]$	$a[20/10]c[10/20]$	$a[0/30]c[0/30]$
$friend(a,b)$	1.00000	1.00000	1.00000
$friend(b,c)$	1.00000	1.00000	1.00000
$friend(c,a)$	0.42478	0.51797	0.49984
$generous(a)$	0.68836	0.28644	0.45345
$generous(b)$	1.00000	1.00000	1.00000
$generous(c)$	0.89564	0.15686	0.57080
$smart(a)$	0.59494	0.58865	0.00000
$smart(b)$	1.00000	0.03550	0.00000
$smart(c)$	0.24699	0.12255	0.00000
$rich(a)$	1.00000	0.66666	0.00000
$rich(b)$	1.00000	0.28147	0.00000
$rich(c)$	1.00000	0.33334	0.00000
$rich(a) \wedge \neg rich(b)$	0.00000	0.38520	0.00000

The above KB_{rich} is non-Horn. It says that there live three people a, b and c in the world where a and b are friends and so are b and c (but it is unknown whether or not a and c are friends). b is known to be generous. We are sure that if y is a friend of x, symmetrically, x is a friend of y. Also it holds that x is rich if-and-only-if x is smart or has a friend who is rich and generous. Friendship is cyclic and being rich is also cyclic.

Suppose we have observed the state of a and c several times. If we observe $rich(a)$ n times while $\neg rich(a)$ m times, we denote the observations by $a[n/m]$. Similarly for $c[n'/m']$. We estimate the probability of $rich(b)$ from observations $a[n/m]$ and $c[n'/m']$. As the set of possible explanations for $rich(a)$ for example, we take $\{smart(a) \vee (friend(a,y) \wedge rich(y) \wedge generous(y)) \mid y \in \{a,b,c\}\}$, i.e. ground instantiations of the r.h.s. of the equivalence formula for $rich(x)$, and dually the negation of the r.h.s. as the ones for $\neg rich(a)$. Similarly for $rich(c)$ and $\neg rich(c)$. Under this abductive setting, we learned parameters $\boldsymbol{\theta}$ in $P_c(\cdot \mid KB_{rich}, \boldsymbol{\theta})$ from observations $a[n/m]$ and $c[n'/m']$ by the EMC algorithm varying n, m, n' and m'.

Table 1 summarizes probabilities $P_r(F) \stackrel{\text{def}}{=} P_c(F \mid KB_{rich}, \boldsymbol{\theta})$ specified by the learned $\boldsymbol{\theta}$ for various ground formulas F.

Columns correspond to each observation set. So for example $friend(c,a)$ is true with probability 0.51797 when parameters are learned from observations $a[20/10]c[10/20]$, i.e. $rich(a)$ observed 20 times and $\neg rich(a)$ 10 times etc.[24]

In the table, facts in KB_{rich} such as $friend(a,b)$ and $generous(b)$ receive probability one. Also we can confirm $rich(a) \Leftarrow rich(b)$, a logical consequence of KB_{rich}, has probability one by computing $P_r(rich(a) \Leftarrow rich(b)) = P_r(rich(a)) + P_r(\neg rich(b)) - P_r(rich(a) \wedge \neg rich(b))$ from the table. Learned probabilities seem to support our intuition that the chance of being rich is affected by friends. For example look at b. When both a and c are always observed to be rich ($a[30/0]c[30/0]$), $P_r(rich(b))$ hits the highest value (1.0) while it decreases to less than one third (0.28147) when b's friends are sometimes observed to be not

[24]For each observation set, we repeated parameter learning 100 times with random start and selected the parameter set that gave the highest likelihood. This applies to Table 2 as well.

Table 2 MAP-learned probabilities with $a[0/30]c[0/30]$

$\alpha = \beta$	$P_r(rich(a))$	$P_r(rich(b))$	$P_r(rich(c))$
1.01	0.00046	0.00013	0.00019
1.1	0.00451	0.00170	0.00434
2	0.02687	0.01338	0.01916
10	0.13469	0.07649	0.12029
100	0.52631	0.41646	0.52583
1000	0.81027	0.72530	0.81030

rich ($a[20/10]c[10/20]$). When they are never observed to be rich ($a[0/30]c[0/30]$), it drops to 0.0.

We notice that the last behavior, i.e. no observation means probability zero, is typical with MLE. If, however, one wishes to avoid this, it is possible to incorporate MAP (maximum a posteriori) estimation into the EMC algorithm though we do not detail it here. Actually when we applied a beta distribution prior $\propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$ uniformly to the probabilities of ground atoms and learned them with $a[0/30]c[0/30]$, changing $\alpha = \beta$ to 1.01, 1.1, 2, 10, 100 and 1000 respectively, $P_r(rich(x))$ ($x \in \{a, b, c\}$) always remained non-zero as shown in Table 2.

It is also interesting to see an interplay between logical inference and probabilistic inference. By logical inference we know $KB_{rich} \vdash rich(a) \Leftarrow rich(b)$. So $P_r(rich(a) \Leftarrow rich(b)) = 1$ holds by the property of CBPMs. Consequently we have $P_r(rich(a)) \geq P_r(rich(b))$, irrespective of learned parameters. Since the same holds for c, $rich(b)$ must satisfy two inequalities, $P_r(rich(a)) \geq P_r(rich(b))$ and $P_r(rich(c)) \geq P_r(rich(b))$, which certainly hold in Table 1 and in Table 2 for all learning cases.

6 Related work

To our knowledge, constraint-based probabilistic modeling is the first probabilistic modeling framework that uniformly deals with (discrete) log-linear models and rule-based probabilistic models. There is a large body of related work but we mention only some of them for reasons of space.

CFDs (case factor diagrams) (McAllester et al. 2004) define log-linear models at propositional level. A set of "feasible" truth assignments on (essentially) finitely many propositional variables are constructed by a CFD combining case statements and factor statements. A log-linear model is then defined considering each feasible assignment as a vector of Boolean features. CFDs and CBPMs are similar in that both define distributions over truth assignments. However CBPMs use first-order clauses instead of CFDs and can define a joint distribution over infinitely many propositional variables (ground atoms).

MLNs (Richardson and Domingos 2006) use typically first-order clauses like CBPMs but as feature functions to define log-linear models. What CBPMs differ most from (clausal) MLNs is the role of clauses. In CBPMs, unlike MLNs, clauses in a knowledge base KB logically exclude Herbrand interpretations that falsify KB, giving them probability zero, and define a (possibly) non-uniform distribution over the remaining interpretations. In MLNs, the same effect is obtained by giving clauses equal weights and taking the infinite limit, but the resulting distribution is always uniform. Also we can simulate rule-based generative models such as PCFGs procedurally by CBPMs using $\text{iff}^s(R)$ as left-to-right rewriting rules

(see Lemma 1 and Theorem 3) and calculate the probability of an observation from rule selection probabilities.

One of the salient features of CBPMs is the unconditional existence of probabilistic models in infinite domains (Theorem 4). Existing approaches are more or less conditional as far as we know. Probability measures (Gibbs measures) for infinite MLNs exist (Singla and Domingos 2007). Their existence however is guaranteed only when clauses are locally finite, in particular when they are σ-determinate, i.e. literals in a clause share the same set of variables over infinite domains. So clauses such as $even(x) \Leftarrow plus(x, y, y)\ (x = y + y)$ in the natural number domain are prohibited. Similarly in the case of infinite BNs, there is some restriction (Pfeffer and Koller 2000; Kersting and De Raedt 2001; Milch et al. 2005; Laskey 2006). For example nodes cannot have infinite parents in recursive probability models (Pfeffer and Koller 2000). Bayesian logic programs (Kersting and De Raedt 2001) use Bayesian clauses of the form $A_0|A_1, \ldots, A_n$ to define a BN such that nodes are ground atoms in the least Herbrand model of the program and A'_0 has incoming edges from $A'_1, \ldots, A'_n$ (A'_i is a ground instance of A_i). An infinite BN is definable under the condition that every node has finite ancestors and there is no cyclic path in the dependency graph derived from the program. In contingent BNs (Milch et al. 2005) where edges are labeled by events such as $X = 0$, nodes can have infinite parents and cyclic paths are allowed. However the set of labels labeling a cyclic path, an infinite upward chain $X_1 \leftarrow X_2 \leftarrow \cdots$, and infinite incoming edges to a node must be inconsistent. MEBN (Laskey 2006) is a first-order probabilistic language based on MFrags which are schematic specifications of local BNs. For an infinite BN to be definable in MEBN, it is required that for each instantiated MFrag for a node, when parent nodes allowed by the MFrag are added indefinitely, the CPT stops changing at some point such that from that point on, no further addition of relevant parent nodes does not affect the CPT.

The EMC algorithm in Sect. 4 offers, though not always, an alternative parameter learning algorithm to the IM (iterative maximization) algorithm (Riezler 1998). The IM algorithm is applicable to log-linear models with incomplete data but since it solves numerical equations at every iteration say by Newton's method, it is a double loop algorithm. By comparison the EMC algorithm is a single-loop algorithm and simple to implement.

7 Conclusion

We have proposed constraint-based probabilistic modeling and proved that discrete probabilistic models (log-linear, rule-based) have an equivalent CBPM (constraint-based probabilistic model) $P_c(\cdot \mid KB)$ which is a joint distribution conditioned on a clausal set KB. We also proved the existence of CBPMs in infinite domains giving probability one to existentially closed logical consequences of KB. We then derived a new EM algorithm named the EMC algorithm applicable to log-linear models with hidden variables for the parameter learning of CBPMs. Finally we applied CBPMs to statistical abduction and proposed constraint-based statistical abduction that allows a knowledge base KB to include arbitrary clauses unlike existing approaches.

Although we provided a theoretical basis for CBPMs in this paper, we have a long list of future research topics. They include Bayesian inference, approximate probability computation, computational complexity of CBPMs and applications to real data. Also the uniqueness condition of CBPMs in infinite domains and the treatment of graphical models with a mixture of directed and undirected edges are interesting future research topics.

Acknowledgements We thank Dr. Yoshitaka Kameya for helpful discussion and careful reading of the manuscript. We also thank anonymous referees for their useful comments and suggestions.

References

Akers, S. (1978). Binary decision diagrams. *IEEE Transactions on Computers*, *27*(6), 509–516.

Boutilier, C., Friedman, N., Goldszmidt, M., & Koller, D. (1996). Context-specific independence in Bayesian networks. In *Proceedings of the 12th conference on uncertainty in artificial intelligence (UAI'96)* (pp. 115–123).

Bryant, R. (1986). Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, *35*(8), 677–691.

Chavira, M., & Darwiche, A. (2005). Compiling Bayesian networks with local structure. In *Proceedings of the 19th international joint conference on artificial intelligence (IJCAI'05)* (pp. 1306–1312).

Chen, J., Muggleton, S., & Santos, J. (2008). Learning probabilistic logic models from probabilistic examples. *Machine Learning*, *73*, 55–85.

Chow, Y., & Teicher, H. (1997). *Probability theory* (3rd ed.). Berlin: Springer.

Cussens, J. (2001). Parameter estimation in stochastic logic programs. *Machine Learning*, *44*(3), 245–271.

De Raedt, L., & Kersting, K. (2008). Probabilistic inductive logic programming. In L. De Raedt, P. Frasconi, K. Kersting, & S. Muggleton (Eds.), *Probabilistic inductive logic programming—theory and applications*. *Lecture notes in computer science* (pp. 1–27). Berlin: Springer.

De Raedt, L., Kimmig, A., & Toivonen, H. (2007). ProbLog: a probabilistic Prolog and its application in link discovery. In *Proceedings of the 20th international joint conference on artificial intelligence (IJCAI'07)* (pp. 2468–2473).

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, *39*(1), 1–38.

Feller, W. (1971). *An introduction to probability theory and its applications*. New York: Wiley.

Getoor, L., & Taskar, B. (Eds.) (2007). *Introduction to statistical relational learning*. Cambridge: MIT Press.

Inoue, K., Sato, T., Ishihata, M., Kameya, Y., & Nabeshima, H. (2009). Evaluating abductive hypotheses using an EM algorithm on BDDs. In *Proceedings of the 21st international joint conference on artificial intelligence (IJCAI'09)* (pp. 810–815).

Ishihata, M., Kameya, Y., Sato, T., & Minato, S. (2008). *Propositionalizing the EM algorithm by BDDs* (Technical Report TR08-0004). Dept. of CS, Tokyo Institute of Technology.

Kersting, K., & De Raedt, L. (2001). Adaptive Bayesian logic programs. In *Proceedings of the 11th international conference on inductive logic programming (ILP'01)* (pp. 104–117).

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th international conference on machine learning (ICML'01)* (pp. 282–289).

Laskey, K. (2006). *MEBN: a logic for open-world probabilistic reasoning* (C4I Center Technical Report C4I06-01). George Mason University Department of Systems Engineering and Operations Research.

Lloyd, J. W. (1984). *Foundations of logic programming*. Berlin: Springer.

McAllester, D., Collins, M., & Pereira, F. (2004). Case-factor diagrams for structured probabilistic modeling. In *Proceedings of the 20th annual conference on uncertainty in artificial intelligence (UAI'04)* (pp. 382–391). Arlington: AUAI.

Milch, B., Marthi, B., Sontag, D., Russell, S., Ong, D., & Kolobov, A. (2005). Approximate inference for infinite contingent Bayesian networks. In *Proceedings of the 10th international workshop on artificial intelligence and statistics (AISTATS'05)* (pp. 1352–1359).

Minato, S. (2001). Zero-suppressed BDDs and their applications. *International Journal on Software Tools for Technology Transfer*, *3*(2), 156–170.

Muggleton, S. (1996). Stochastic logic programs. In L. De Raedt (Ed.), *Advances in inductive logic programming* (pp. 254–264). Amsterdam: IOS.

Pfeffer, A., & Koller, D. (2000). Semantics and inference for recursive probability models. In *Proceedings of the 7th national conference on artificial intelligence (AAAI'00)* (pp. 538–544).

Poole, D. (1993). Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, *64*(1), 81–129.

Poole, D. (1997). The independent choice logic for modeling multiple agents under uncertainty. *Artificial Intelligence*, *94*(1–2), 7–56.

Poole, D. (2000). Abducing through negation as failure: stable models within independent choice logic. *Journal of Logic Programming*, *44*, 5–35.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*, 107–136.

Riezler, S. (1998). *Probabilistic constraint logic programming*. PhD thesis, Universität Tübingen.

Sato, T., & Kameya, Y. (2001). Parameter learning of logic programs for symbolic-statistical modeling. *Journal of Artificial Intelligence Research*, *15*, 391–454.

Sato, T., & Kameya, Y. (2002). Statistical abduction with tabulation. In A. Kakas & F. Sadri (Eds.), *LNAI: Vol. 2408*. *Computational logic: logic programming and beyond* (pp. 567–587). Berlin: Springer.

Sato, T., & Kameya, Y. (2008). New advances in logic-based probabilistic modeling by PRISM. In L. De Raedt, P. Frasconi, K. Kersting, & S. Muggleton (Eds.), *LNAI: Vol. 4911*. *Probabilistic inductive logic programming* (pp. 118–155). Berlin: Springer.

Singla, P., & Domingos, P. (2007). Markov logic in infinite domains. In *Proceedings of the twenty-third conference on uncertainty in artificial intelligence (UAI'07)* (pp. 368–375).