Constraint-based Probabilistic Modeling for Statistical Abduction

Taisuke Sato[1,2], Masakazu Ishihata[1], and Katsumi Inoue[2,1]

[1] Graduate School of Information Science and Engineering,
Tokyo institute of Technology
{sato,ishihata}@mi.cs.titech.ac.jp
[2] Principles of Informatics Research Division
National Institute of Informatics
kii@nii.ac.jp

1 Introduction

Suppose we have i.i.d. data as a bag of ground atoms and wish to build their logic-based probabilistic model [1, 2]. 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 typically defines a log-linear model $p(x) = Z^{-1} \exp(\sum_i w_i f_i(x))$ where the f_i's are boolean features taking 1 (true) or 0 (false), the w_i's weights and Z a normalizing constant. For example MLNs (Markov logic networks) [3] use first-order clauses as templates whose ground instantiations work as boolean features.

The latter type, rule-based approaches such as SLPs [4], ICL [5], PRISM [6, 7] and more recently ProbLog [8], employs definite or general clauses to describe a generative process of output. They proof-theoretically define a distribution over ground atoms [4, 5], or model-theoretically define a probability measure over possible worlds, i.e. the set of Herbrand interpretations [6, 8]. Joint distributions thus defined are a subclass of log-linear models where the normalizing constant is unity but able to cover a variety of probabilistic models from BNs (Bayesian networks) to PCFGs (probabilistic context free grammars).

In this paper[3]we introduce *constraint-based probabilistic modeling*, a new modeling framework that uniformly covers the above two types. It defines *CBPMs (constraint-based probabilistic models)*, i.e. conditional distributions $P_c(\cdot \mid KB)$ such that $P_c(\cdot)$ is a product of Bernoulli distributions and KB is a set of clauses. It is motivated by an observation that abductive reasoning for metabolic networks [9] requires a flexible framework capable of describing cyclic dependencies caused by positive/negative feedback among metabolites.

The basic idea of CBPMs is simple; independent atoms are constrained by a knowledge base KB. $P_c(\cdot \mid KB)$ is a conditional distribution over the Herbrand interpretations that satisfy KB. Yet they are expressive enough statistically and logically. Statistically they can define both generative models such as PCFGs and discriminative models such as CRFs (conditional random fields). Logically

[3] Distributions are discrete throughout the paper.

Presented at the 19th International Conference on Inductive Logic Programming (ILP'09).

we can directly reflect our knowledge in the first-order KB and perform logical deduction freely. Despite broad coverage of probabilistic models by CBPMs, probabilities are uniformly learned from data by the BDD-EMC algorithm developed for CBPMs efficiently using dynamic programming.

2 Constraint-based probabilistic models

In this section we define CBPMs and state theorems about them. Let $\mathcal{L}$ be a countable first order language, $\mathcal{H}_B$ 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 $\mathcal{H}_B$ and identify a 0-1 vector $(1, 0, \ldots)$ with a Herbrand interpretation making A_1 true (1), A_2 false (0) .... Let P_c be a (-n infinite) product distribution $P_c(A_1 = x_1, A_2 = x_2, \ldots) = \prod_{i=1}^{\infty} P_c(A_i = x_i)$ $(x_i \in \{0,1\})$ for the A_i's and identify it with a probability measure over the Herbrand interpretations for $\mathcal{L}$. So all ground atoms are independent and every closed formula φ in $\mathcal{L}$ is a random variable taking a value $\in \{1, 0\}$ w.r.t. P_c. We write $P_c(\varphi)$ (resp. $P_c(\neg\varphi)$) instead of $P_c(\varphi = 1)$ (resp. $P_c(\varphi = 0)$) and also $P(x)$ instead of $P(X = x)$ when the context is clear.

A *CBPM (constraint-based probabilistic model)* is a conditional probability measure $P_c(\cdot \mid KB)$ on the set of Herbrand interpretations conditioned on a set KB of countably many clauses. Although it always exists 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 that $\mathcal{L}$ has no function symbol, $\mathcal{H}_B$ is finite and $P_c(KB) > 0$ (KB is consistent)[4].

Let X be a random variable and $V(X)$ the set of values X takes. We denote by $\lceil X = x \rceil$ a propositional random variable which takes on 1 when the event $X = x$ $(x \in V(X))$ happens and 0 otherwise. For a given joint distribution $P(X_1 = x_1, \ldots, X_N = x_N)$, consider a CBPM $P_c(\lceil X_1' = x_1 \rceil, \ldots, \lceil X_N' = x_N \rceil \mid KB)$ where the X_i''s may or may not be identical to the X_i's. If $P_c(\lceil X_1' = x_1 \rceil, \ldots, \lceil X_N' = x_N \rceil \mid KB) = P(X_1 = x_1, \ldots, X_N = x_N)$ holds for every possible x_i $(1 \leq i \leq N)$, we say that $P(X_1 = x_1, \ldots, X_N = x_N)$ *is equivalent to* $P_c(\lceil X_1' = x_1 \rceil, \ldots, \lceil X_N' = x_N \rceil \mid KB)$. These notations are extended to vectors $\boldsymbol{X}, \boldsymbol{x}$.

We state two theorems without proofs. The first one deals with log-linear (discriminative) models where a joint distribution $P(\boldsymbol{x})$ is given as a product of potential functions $P(\boldsymbol{X} = \boldsymbol{x}) = Z^{-1} \prod_{i=1}^{M} F_i(\boldsymbol{x}_i)$. Here $\boldsymbol{X}$, $\boldsymbol{x}$ and $\boldsymbol{x}_i (\subseteq \boldsymbol{x})$ are vectors and Z is a normalizing constant.

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

$$
\begin{aligned}
P(\boldsymbol{X} = \boldsymbol{x}) &= P_c(\lceil \boldsymbol{X}_1' = \boldsymbol{x}_1 \rceil, \ldots, \lceil \boldsymbol{X}_M' = \boldsymbol{x}_M \rceil \mid C \wedge \textstyle\bigwedge_i KB_i) \\
&= \frac{\prod_i P_c^{(i)}(\lceil \boldsymbol{X}_i' = \boldsymbol{x}_i \rceil \mid KB_i)}{\sum_{\boldsymbol{x}} \prod_i P_c^{(i)}(\lceil \boldsymbol{X}_i' = \boldsymbol{x}_i \rceil \mid KB_i)}
\end{aligned}
$$

[4] The infinite case will be treated in a longer version of this paper.

where C and the KB_i's are some boolean formulas and $P_c^{(i)}(\lceil \boldsymbol{X}_i^{'} = \boldsymbol{x}_i \rceil \mid KB_i)$ ($1 \leq i \leq M$) is a CBPM defined by KB_i equivalent to a factor joint distribution $Q^{(i)}(\boldsymbol{X}_i = \boldsymbol{x}_i)$ such that $P_c^{(i)}(\lceil \boldsymbol{X}_i^{'} = \boldsymbol{x}_i \rceil \mid KB_i) = Q^{(i)}(\boldsymbol{X}_i = \boldsymbol{x}_i) = \frac{F_i(\boldsymbol{x}_i)}{\sum_{\boldsymbol{x}_i} F_i(\boldsymbol{x}_i)}$.

We next consider rule-based generative models such as PCFGs. We use PRISM which is a symbolic-statistical modeling language based on Prolog extended with a built-in predicate `msw/3` representing probabilistic choices [6, 7]. PRISM programs cover generative models in general and PCFGs in particular.

To state the theorem below which says CBPMs can simulate PRISM, we introduce a binary relation "$\succ$" over $\mathcal{H}_B$ 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 $A_1 \succ A_2 \succ \cdots \succ A_1$.

Theorem 2. *Suppose a PRISM program DB is cycle-free. DB has an equivalent CBPM such that for a non-*`msw` *ground atom G, we have $P_{DB}(G) = P_c(G \mid KB)$ where $P_{DB}(G)$ is the probability of G defined by DB and KB is a set of certain clauses related to DB.*

3 Constraint-based statistical abduction

In this section we apply CBPMs to statistical abduction.

Abduction is one of logical inferences (deduction, induction, abduction) which infers the best explanation E for our 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 [6].

Suppose we have i.i.d. observations $O_1, \ldots, O_T$, ground literals, and a knowledge base *KB* that may contain non-Horn clauses as well as cyclic rules such as $friend(X, Y) \Leftarrow friend(Y, X)$. For each O_t ($1 \leq t \leq T$), we search for an explanation E_t in the search space $\mathcal{E}$ of possible explanations such that $KB \wedge E_t \vdash O_t$ and $KB \wedge E_t$ is consistent. We assume $\mathcal{E}$ is specified beforehand as a set of conjunctions of abducibles or a set of clauses having at most three literals etc. Each O_t can have multiple explanations $E_1^{(t)}, \ldots, E_{k_t}^{(t)}$ and we call the disjunction $E^{(t)} = E_1^{(t)} \vee \cdots \vee E_{k_t}^{(t)}$ *disjunctive explanation* for O_t. We then construct a CBPM $P_c(\cdot \mid KB, \boldsymbol{\theta})$ that specifies a distribution on Herbrand interpretations for $\mathcal{H}_B$. Here $\boldsymbol{\theta}$ collectively stands for parameters, i.e. the probabilities of atoms in $\mathcal{H}_B$ being true. We estimate $\boldsymbol{\theta}$ by MLE (maximum likelihood estimation) as the maximizer of the likelihood function $\mathbf{L}(\boldsymbol{\theta}) = \prod_{t=1}^{T} P_c(E^{(t)} \mid KB, \boldsymbol{\theta})$.

The reason for our choice of this likelihood function is as follows. First note that O_t and *KB* are logically independent (o.w. *KB* would explain O_t) and they are connected solely through the $E_i^{(t)}$'s. So simply maximizing $P_c(O_t \mid KB, \boldsymbol{\theta})$ will not work. Also note we wish our explanation is true but we do not know which one is true. So we instead wish their disjunction, $E^{(t)}$, is true. Hence we maximize $P_c(O_t \wedge E^{(t)} \mid KB, \boldsymbol{\theta})$. Since $KB \models E^{(t)} \Rightarrow O_t$, we replace $P_c(O_t \wedge E^{(t)} \mid KB, \boldsymbol{\theta})$ with $P_c(E^{(t)} \mid KB, \boldsymbol{\theta})$, reaching our $\mathbf{L}(\boldsymbol{\theta})$.

After learning $\boldsymbol{\theta}$, we determine the most likely explanation for O_t as the one having the highest probability in $\{P_c(E^{(t)}_{k_j} \mid KB, \boldsymbol{\theta}) \mid 1 \leq j \leq k_t\}$. We learn $\boldsymbol{\theta}$ by an EM algorithm, *the BDD-EMC algorithm* which is derived for CBPMs. Regrettably we have to entirety omit details of the BDD-EMC algorithm for space limitations. We just remark that it is a generalization of the FAM algorithm [10] and the BDD-EM algorithm [11] which is implemented on BDDs and applicable to log-linear models with hidden variables.

4 Learning example

We present here a small learning example. It is often observed that smart people are rich and rich people know each other. The following KB_{rich} formalizes this observation.

$$KB_{rich} = \begin{cases} friend(a,b) \quad friend(b,c) \\ friend(X,Y) \Leftarrow friend(Y,X) \\ rich(X) \Leftrightarrow smart(X) \vee \\ \qquad\qquad \exists Y\,(friend(X,Y) \wedge rich(Y) \Leftarrow \neg noise(Y,X)) \end{cases}$$

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). 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 X is smart or has a rich friend, the latter being valid only if $\neg noise(X,Y)$, i.e. no noise occurs and vice versa. Friendship is cyclic and being rich is also (probabilistically) cyclic here.

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 $c(n/m)$. Also suppose we wish to estimate the probability of $rich(b)$ from observations $a(n/m)$ and $c(n'/m')$. As the explanation for $rich(a)$, we choose the right hand side of $rich(a)$, i.e. $smart(a) \vee \exists Y (friend(a,Y) \wedge rich(Y) \Leftarrow \neg noise(Y,a))$ with Y instantiated to b and c, and dually, its negation as the one for $\neg rich(a)$. Similarly for $rich(c)$ and $\neg rich(c)$.

Under this abductive setting we conducted a learning experiment varying $a(n/m)$ and $c(n'/m')$ with the probability of $noise(X,Y)$ fixed to 0.1. Figure 1 plots the log-likelihood of the disjunctive explanations for the observations $(a(2/1)c(1/2))$. One can see a sharp rise of the log-likelihood at early iterations of the BDD-EMC algorithm.

Table 1 summarizes learned probabilities $P_c(A \mid KB_{rich}, \boldsymbol{\theta})$ for various atoms A. It shows that $P_c(rich(b) \mid KB_{rich})$ is the highest (0.9998) when a and c, b's friends, are observed to be rich three times $(a(3/0)c(3/0))$ while it decreases to one third (0.343) when they are sometimes observed to be not rich $(a(2/1)c(1/2))$. When a and c are never observed to be rich $(a(0/3)c(0/3))$, the probability drops to 0.001.

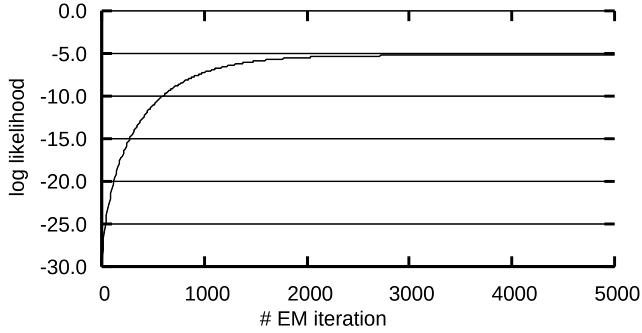

Fig. 1. Learning curve

Table 1. Learned probabilities

Atoms	Observations		
	a(3/0)c(3/0)	a(2/1)c(1/2)	a(0/3)c(0/3)
$friend\,(a, b)$	1.0000	1.0000	1.0000
$friend\,(a, c)$	0.3551	0.0524	0.6113
$friend\,(b, a)$	1.0000	1.0000	1.0000
$friend\,(b, c)$	1.0000	1.0000	1.0000
$friend\,(c, a)$	0.3551	0.0524	0.6110
$friend\,(c, b)$	1.0000	1.0000	1.0000
$smart\,(a)$	0.7799	0.5635	0.0018
$smart\,(b)$	0.9967	0.0953	0.0003
$smart\,(c)$	0.8444	0.0546	0.0047
$rich\,(a)$	0.9998	0.6440	0.0023
$rich\,(b)$	0.9998	0.3430	0.0010
$rich\,(c)$	0.9994	0.3207	0.0059

5 Concluding remarks

To our knowledge, constraint-based probabilistic modeling is the first logic-based framework applicable to both logically defined log-linear models [12, 3] and rule-based generative models [4–6, 8]. CFDs (case factor diagrams) define log-linear models at propositional level [12] whereas CBPMs use first-order clauses and we can make logical inference at first-order level like $P_c(\varphi \mid KB) = 1$ if $KB \vdash \varphi$. MLNs [3] use first-order clauses to define log-linear models like CBPMs. What CBPMs differ most from MLNs is the role of clauses. In CBPMs, unlike MLNs, clauses logically exclude some Herbrand interpretations, giving them probability 0, and define (not necessarily uniform) distributions on the remaining interpretations. Also they allow us to simulate generative models (see Theorem 2) such as PCFGs and to compute probabilities of sentences in the given PCFG.

The BDD-EMC algorithm for CBPMs offers, though not always, an alternative parameter learning algorithm to the IM (iterative maximization) algorithm [13]. 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 BDD-EMC algorithm is a single-loop algorithm and simple to implement.

We are planning to apply constraint-based statistical abduction to the analysis of bio-sequences as shown in [9].

References

1. Getoor, L., Taskar, B., eds.: Introduction to Statistical Relational Learning. MIT Press, Cambridge, MA (2007)
2. De Raedt, L., Kersting, K.: Probabilistic inductive logic programming. In De Raedt, L., Frasconi, P., Kersting, K., Muggleton, S., eds.: Probabilistic Inductive Logic Programming - Theory and Applications. Lecture Notes in Computer Science. Springer (2008) 1–27
3. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning **62** (2006) 107–136
4. Muggleton, S.: Stochastic logic programs. In De Raedt, L., ed.: Advances in Inductive Logic Programming. IOS Press (1996) 254–264
5. Poole, D.: The independent choice logic for modeling multiple agents under uncertainty. Artificial Intelligence **94**(1-2) (1997) 7–56
6. Sato, T., Kameya, Y.: Parameter learning of logic programs for symbolic-statistical modeling. Journal of Artificial Intelligence Research **15** (2001) 391–454
7. Sato, T., Kameya, Y.: New Advances in Logid-Based Probabilistic Modeling by PRISM. In De Raedt, L., Frasconi, P., Kersting, K., Muggleton, S., eds.: Probabilistic Inductive Logic Programming. LNAI 4911, Springer (2008) 118–155
8. De Raedt, L., Kimmig, A., Toivonen, H.: ProbLog: A probabilistic Prolog and its application in link discoverry. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07). (2007) 2468–2473
9. Chen, J., Muggleton, S., Santos, J.: Learning probabilistic logic models from probabilistic examples. Machine Learning **73** (2008) 55–85
10. Cussens, J.: Parameter estimation in stochastic logic programs. Machine Learning **44**(3) (Sept. 2001) 245–271
11. Inoue, K., Sato, T., Ishihata, M., Kameya, Y., Nabeshima, H.: Evaluating abductive hypotheses using an em algorithm on bdds. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09). (2009) to appear.
12. McAllester, D., Collins, M., Pereira, F.: Case-factor diagrams for structured probabilistic modeling. In: Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI'04), Arlington, Virginia, AUAI Press (2004) 382–391
13. Riezler, S.: Probabilistic Constraint Logic Programming. PhD thesis, Universität Tübingen (1998)