JMLR: Workshop and Conference Proceedings 20 (2011) 81–96 Asian Conference on Machine Learning

Bayesian inference for statistical abduction using Markov chain Monte Carlo

Masakazu Ishihata ISHIHATA@MI.CS.TITECH.AC.JP
Graduate School of Information Science and Engineering,
Tokyo Institute of Technology
Taisuke Sato SATO@MI.CS.TITECH.AC.JP
Graduate School of Information Science and Engineering,
Tokyo Institute of Technology

Editor: Chun-Nan Hsu and Wee Sun Lee

Abstract

Abduction is one of the basic logical inferences (deduction, induction and abduction) and derives the best explanations for our observation. Statistical abduction attempts to define a probability distribution over explanations and to evaluate them by their probabilities. The framework of statistical abduction is general since many well-known probabilistic models, i.e., BNs, HMMs and PCFGs, are formulated as statistical abduction. *Logic-based probabilistic models* (LBPMs) have been developed as a way to combine probabilities and logic, and it enables us to perform statistical abduction. However, most of existing LBPMs impose restrictions on explanations (logical formulas) to realize efficient probability computation and learning. To relax those restrictions, we propose two MCMC (Markov chain Monte Carlo) methods for Bayesian inference on LBPMs using binary decision diagrams. The main advantage of our methods over existing methods is that it has no restriction on formulas. In the context of statistical abduction with Bayesian inference, whereas our deterministic knowledge can be described by logical formulas as rules and facts, our non-deterministic knowledge like frequency and preference can be reflected in a prior distribution in Bayesian inference. To illustrate our methods, we first formulate LDA (latent Dirichlet allocation) which is a well-known generative probabilistic model for bag-of-words as a form of statistical abduction, and compare the learning result of our methods with that of an MCMC method called collapsed Gibbs sampling specialized for LDA. We also apply our methods to diagnosis for failure in a logic circuit and evaluate explanations using a posterior distribution approximated by our method. The experiment shows Bayesian inference achieves better predicting accuracy than that of Maximum likelihood estimation.

Keywords: statistical abduction, Bayesian inference, Markov chain Monte Carlo, binary decision diagrams

1. Introduction

Abduction is one of the basic logical inferences (deduction, induction and abduction) and derives the best explanation E for our observation O such that E is consistent with *knowledge base* KB and $KB \wedge E \models O$. For example, we observe grass in our garden is wet and have knowledge that grass is wet if it rained or someone watered the grass. Then, we can derive two explanations "it rained" and "someone watered garden" for the observation. The problem here is we do not know which explanation is the best. Statistical abduction

attempts to define a probability distribution over explanations and to evaluate them by their probabilities. For example, if we know that the probabilities of "it rained" and "someone watered garden" are 0.3 and 0.5, respectively, then we can say the second one is the best. Statistical abduction is a general framework since many well-known probabilistic models, i.e., BNs (Bayesian networks), HMMs (hidden Markov models) and PCFGs (probabilistic context-free grammars), are formulated as statistical abduction (Sato and Kameya (2001)). For example, explanations for HMMs correspond to hidden states for a given sequence, and those for PCFGs are parse trees for a given corpse. Recently, statistical abduction has been applied to diagnosis (Poole (1993), Ishihata et al. (2010)), plan recognition (Singla and Mooney (2011), Raghavan and Mooney (2011)), systems biology (Inoue et al. (2009), Synnaeve et al. (2011)) etc.

In the past two decades, a number of formalisms to combine probability and logic have been proposed in the area of statistical relational learning (SRL) (Getoor and Taskar (2007)). Well-known examples include SLPs (stochastic logic programs) (Muggleton (1996)), ICL (independent choice logic) (Poole (1997)), PRISM (Sato and Kameya (2001)), BLPs (Bayesian logic programs) (Kersting and De Raedt (2001)), MLNs (Markov logic networks) (Richardson and Domingos (2006)) and ProbLog (De Raedt et al. (2007)). In recent years, statistical abduction systems based on BLPs and MLNs have been developed and the former is called BALPs (Bayesian abductive logic programs) (Kate and Mooney (2009)) and the latter is abductive Markov logic (Raghavan and Mooney (2010)). Whereas most of statistical abduction systems including those employ MLE (maximum likelihood estimation) to obtain probabilities of explanations, PRISM which is based on Prolog supplies various learning methods other than MLE such as MAP (maximum a posterior) inference and recently Bayesian inference (Sato et al. (2009), Sato (2011)). The introduction of Bayesian inference for statistical abduction gives the following benefits. The first one is the enlarged range of usable models. For example, LDA (latent Dirichlet allocation) which is a well-known probabilistic generative model for bag-of-words can be formulated as statistical abduction with Bayesian inference. The second benefit is that our non-deterministic knowledge can be explicitly reflected in evaluation of explanations as a prior distribution of Bayesian inference. In statistical abduction, our deterministic knowledge such as rules and facts can be described as logical formulas but our non-deterministic knowledge such as frequency and preference seems difficult to describe by logic. Bayesian inference allows us to represent such knowledge as a prior distribution and explicitly reflects it in evaluation of explanations. However, PRISM has a problem that it assumes the exclusiveness condition on explanations, that is, disjuncts must be probabilistically exclusive. Although most of statistical abduction systems have such restrictions to realize efficient probability computation and learning, but, they prevent us from enjoying the full expressive power of logic.

In this paper, we propose two MCMC methods for Bayesian inference in statistical abduction, which have no restriction over explanations. They are applicable to any explanations as long as they are described as boolean formulas and can relax restrictions of existing statistical abduction systems including PRISM.

The remainder of this paper is organized as follows. We first formulate Bayesian inference for statistical abduction. Then, we propose two MCMC methods to perform Bayesian inference for statistical abduction, one is a Gibbs sampling and the other is a component-

wise Metropolis-Hasting sampling. Next, we apply our methods to finding topics of LDA and to diagnosing stochastic errors in logic circuits. Finally, we discuss related work and future work, followed by conclusion.

2. Preliminary

2.1. Statistical abduction on PBPMs

We here formulate statistical abduction as inference on probabilistic models called *propositional logic-based probabilistic models* (PBPMs). Suppose we have an observation O to be explained and knowledge base KB consisting of first-order clauses. Then, the task of *logical* abduction is to search for an explanation E such that $KB \wedge E \models O$ and $KB \wedge E$ is consistent. In usual, the search space of explanations are limited such as conjunctions of a set of atoms called *abducibles*. In statistical abduction, we introduce a probability distribution over abducibles and define probabilities of explanations. The task of statistical abduction is to infer the best explanation which has the highest probability.

The above framework of statistical abduction can be formulated as inference on probabilistic models. Let $\theta_j \equiv \{\theta_{jv}\}_{v=1}^{M_j}$ $(0 \leq \theta_{jv} \leq 1, \sum_{v=1}^{M_j} \theta_{jv} = 1)$ be a parameter vector of a *categorical distribution* $\mathrm{Cat}\,(\theta_j)$ corresponding to an M_j-sided dice, and also let $x_i \equiv \{x_{iv}\}_{v=1}^{N_i}$ $(x_{iv} \in \{0,1\}, \sum_{v=1}^{N_i} x_{iv} = 1)$ be a value vector drawn from $\mathrm{Cat}\,(\theta_{j_i})$, where j_i is the index of the categorical distribution which generates x_i. (So, $N_i = M_{j_i}$ holds for each i). We use v_i $(1 \leq v_i \leq N_i)$ to denote v such that $x_{iv} = 1$. Then a probability $p(x_i \mid \theta_{j_i})$ is equal to $\theta_{j_i v_i}$. Let θ and x be $\{\theta_j\}_{j=1}^{M}$ and $\{x_i\}_{i=1}^{N}$, respectively. Then, a joint distribution $p(x \mid \theta)$ is computed as follows:

$$p(x \mid \theta) = \prod_{j=1}^{M} \prod_{v=1}^{M_j} \theta_{jv}^{\sigma_{jv}(x)}, \qquad \sigma_{jv}(x) \equiv \sum_{i:j_i=j} x_{iv}. \tag{1}$$

Now, we introduce $f(x)$ which is a function of x such that $f(x) \in \{0,1\}$, and use f (resp. $\neg f$) to denote the value of $f(x)$ is 1 (resp. 0). Then, the probability $p(f \mid \theta)$ is defined as follows:

$$p(f \mid x) \equiv f(x), \qquad p(f \mid \theta) = \sum_{x} p(f, x \mid \theta) = \sum_{x} f(x) p(x \mid \theta).$$

We call a joint distribution $p(f, x \mid \theta)$ a *base model* and its graphical representation is shown at the left upper in Fig. 1. Suppose the value and the definition of $f(x)$ are given as an observation O and knowledge base KB, respectively. Then, computing the most probable x given them on the base models is almost equivalent to performing statistical abduction, that is, finding the explanation E which has the highest probability from all possible x (a search space). However, it slightly differs from the *logical* statistical abduction in that KB is described by logic in logical abduction. Now, we *propositionalize* the base model $p(f, x \mid \theta)$ to describe f as a boolean formula in *independent* boolean random variables. Let "$x_i = v$" be a boolean random variable taking 1 (true) if $x_{iv} = 1$. Then, f can be represented as a boolean formula as follows:

$$f = \bigvee_{x:f(x)=1} f_x, \qquad f_x = \bigwedge_{x_i \in x} \text{“}x_i = v_i\text{”},$$

Here, note that "$x_i = v$" and "$x_i = v'$" $(v \neq v')$ depend on each other. However, they can be described as a boolean formula in *independent* boolean random variables $b \equiv \{b_{iv} \mid b_{iv} \equiv$ "$x_i \leq v \mid x_i \geq v$", $1 \leq i \leq N, 1 \leq v < N_i\}$ as follows:

$$\text{“}x_i = v\text{”} \equiv \begin{cases} b_{iv} \wedge \bigwedge_{v'=1}^{v-1} \neg b_{iv'} & 1 \leq v < N_i \\ \bigwedge_{v'=1}^{v-1} \neg b_{iv'} & v = N_i \end{cases},$$

Thus, f_x can be also described as boolean formulas in b, and $p(f_x \mid \theta) \equiv \sum_b f_x(b) p(b \mid \theta)$ equals to $p(x \mid \theta)$ if the probability of b_{iv} is defined as follows (Ishihata et al. (2010)):

$$p(b_{iv} = 1 \mid \theta) \equiv \frac{\theta_{j_i v}}{\phi_{j_i v}}, \qquad p(b_{iv} = 0 \mid \theta) \equiv \frac{\phi_{j_i, v+1}}{\phi_{j_i v}}, \qquad \phi_{jv} \equiv \sum_{v'=v}^{M_j} \theta_{jv'}.$$

For example, the probability of "$x_i = v$" $(1 \leq v < N_i)$ can be computed using b as follows:

$$\begin{aligned} p(\text{“}x_i = v\text{”} \mid \theta) &= p(b_{iv} = 1 \mid \theta) \prod_{v'=1}^{v-1} p(b_{iv'} = 0 \mid \theta) \\ &= \frac{\theta_{j_i v}}{\phi_{j_i v}} \prod_{v'=1}^{v-1} \frac{\phi_{j_i, v'+1}}{\phi_{j_i v'}} \\ &= \theta_{j_i v}. \end{aligned}$$

In the same way, a probabilistic event f can be described as boolean formulas in independent boolean random variables b, and its probability are computed by $p(b \mid \theta)$ as follows:

$$p(f \mid \theta) = \sum_b p(f, b \mid \theta) = \sum_b f(b) p(b \mid \theta),$$

where $f(b)$ is the value of the boolean formula of f given an assignment b. We call the joint distribution $p(f, b \mid \theta)$ a *propositional logic-based probabilistic model* (PBPM) for a base model $p(f, x \mid \theta)$ and its graphical representation is shown at the left lower in Fig. 1. Consequently, statistical abduction is formulated as a problem to infer the most probable x given f and its boolean formula in b. Here, PBPMs have no restriction on a boolean formula of f and define probabilities over any boolean formulas in b.

2.2. Bayesian inference for statistical abduction on PBPMs

Given an observation f and its boolean formula in b, we here perform Bayesian inference to infer the most probable x. In Bayesian inference, we assume a parameter θ as a random variable and introduce a *prior distribution* $p(\theta \mid \alpha)$ $(\alpha \equiv \{\alpha_k\}_{k=1}^{L})$ defined as

$$p(\theta \mid \alpha) = \prod_{j=1}^{M} p(\theta_j \mid \alpha_{k_j}), \quad p(\theta_j \mid \alpha_{k_j}) = \frac{1}{\mathrm{Z}(\alpha_{k_j})} \prod_{v=1}^{M_j} \theta_{jv}^{\alpha_{k_j v} - 1}, \quad \mathrm{Z}(\alpha_k) \equiv \frac{\prod_{v=1}^{L_k} \Gamma(\alpha_{kv})}{\Gamma\left(\sum_{v=1}^{L_k} \alpha_{kv}\right)},$$

where $\alpha_k \equiv \{\alpha_{kv}\}_{v=1}^{L_k}$ $(\alpha_{kv} > 0)$ is a parameter of a *Dirichlet distribution* $\mathrm{Dir}(\alpha_k)$ and k_j denotes the index of the Dirichlet distribution which generates θ_j. The introduction of the

prior $p(\theta \mid \alpha)$ modifies graphical representations of base models and PBPMs to those in the right side in Fig. 1. Since Dirichlet distributions are *conjugate* to categorical distributions, the *posterior distribution* $p(\theta \mid x, \alpha)$, which is the modified distribution of θ by a given x, is also a product of Dirichlet distributions as follows:

$$p(\theta \mid x, \alpha) = \frac{p(x \mid \theta)p(\theta \mid \alpha)}{p(x \mid \alpha)}, \qquad p(x \mid \theta)p(\theta \mid \alpha) = \prod_{j=1}^{M} \frac{1}{\mathrm{Z}\left(\alpha_{k_j}\right)} \prod_{v=1}^{M_j} \theta_{jv}^{\alpha_{k_j v}+\sigma_{jv}(x)-1},$$

where $p(x \mid \alpha)$ is computed as follows:

$$p(x \mid \alpha) = \int p(x \mid \theta)p(\theta \mid \alpha)d\theta = \prod_{j=1}^{M} \frac{\mathrm{Z}\left(\alpha_{k_j} + \sigma_j(x)\right)}{\mathrm{Z}\left(\alpha_{k_j}\right)}, \qquad \sigma_j(x) \equiv \{\sigma_{jv}(x)\}_{v=1}^{M_j}. \tag{2}$$

We here define the most probable x given f as one that maximizes $p(x \mid f, \alpha)$ computed as

$$p(x \mid f, \alpha) = \frac{f(x)p(x \mid \alpha)}{p(f \mid \alpha)}, \qquad p(f \mid \alpha) = \sum_x f(x)p(x \mid \alpha),$$

where $p(f \mid \alpha)$ is called *marginal likelihood*. Unfortunately, computing $p(f \mid \alpha)$ and $\mathrm{argmax}_x\, p(x \mid f, \alpha)$ involve evaluating $p(x \mid \alpha)$ on the large discrete search space. To the best of our knowledge, there is no efficient algorithm for computing $p(f \mid \alpha)$, let alone that for $\mathrm{argmax}_x\, p(x \mid f, \alpha)$.

We avoid this difficulty by switching from computing $\mathrm{argmax}_x\, p(x \mid f, \alpha)$ to sampling x from $p(x \mid f, \alpha)$. Suppose we have K samples $\{x_{(k)}\}_{k=1}^{K}$ taken from $p(x \mid f, \alpha)$. Then, one which maximize $p(x_{(k)} \mid \alpha)$ is the most probable explanation in the sample. More generally, suppose we are given N boolean formulas (explanations) $f_1, \ldots, f_N$ and would like to choose the most probable f_i given f. Then, we can approximate $p(f_i \mid f, \alpha)$ using the samples $\{x_{(k)}\}_{k=1}^{K}$ by

$$p(f_i \mid f, \alpha) = \sum_x f_i(x)p(x \mid f, \alpha) \approx \sum_{k=1}^{K} \frac{f_i(x_{(k)})}{K},$$

and choose the most probable f_i using the approximated probabilities. In addition, we can also approximate the marginal likelihood $p(f \mid \alpha)$ using the samples and a particular $\hat{\theta}$ as follows:

$$p(f \mid \alpha) = \frac{p(\hat{\theta} \mid \alpha)p(f \mid \hat{\theta})}{\sum_x p(\hat{\theta} \mid x, \alpha)p(x \mid f, \alpha)}, \qquad \sum_x p(\hat{\theta} \mid x, \alpha)p(x \mid f, \alpha) \approx \frac{1}{K}\sum_{k=1}^{K} p(\hat{\theta} \mid x_{(k)}, \alpha),$$

where $p(\hat{\theta} \mid \alpha)$ is easy to compute and $p(f \mid \hat{\theta})$ can be computed by a BDD-based probability computation algorithm proposed by Ishihata et al. (2010).

In this paper, we propose two MCMC methods to take a sample of x from $p(x \mid f, \alpha)$. The first one is a Gibbs sampling and described in Section 3. The second one is a component-wise Metropolis-Hastings sampling and proposed in Section 4.

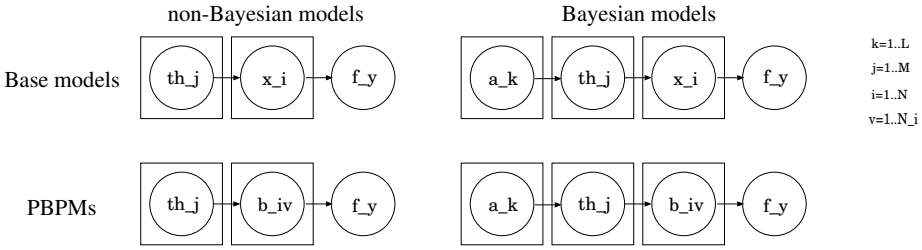

Figure 1: Graphical representations of base models and PBPMs

3. Gibbs sampling for PBPMs

3.1. Gibbs sampling

Markov chain Monte Carlo (MCMC) methods are a class of general sampling algorithms based on Markov chains. Let $\pi(s)$ be a target distribution that we would like to get samples. An MCMC method constructs a Markov chain with a transition probability distribution $P(\tilde{s} \mid s)$ which is easy to sample from and has the target distribution $\pi(s)$ as its *equilibrium distribution*, that is, $P_i(s)$ converges to $\pi(s)$ as $i \to \infty$ where $P_i(s)$ is the probability of s after i state changes.

The Gibbs sampling is an example of MCMC methods and generates a state s by sampling each component (a variable or a subset of variables in s) from a conditional distribution given the current values of the other variables. A Gibbs sampling for $p(x \mid f, \alpha)$ is naively constructed with N components x_i $(1 \leq i \leq N)$ if a conditional distribution $p(x_i \mid x_{-i}, f, \alpha)$ is computable, where $x_{-i} \equiv x \backslash \{x_i\}$. Fortunately, a conditional probability $p(x_i = v \mid x_{-i}, f, \alpha)$ is easily computed as follows:

$$p(x_i = v \mid x_{-i}, f, \alpha) \propto f(\{x_i = v, x_{-i}\}) \frac{\alpha_{k_i v} + \sigma_{j_i v}(x_{-i})}{\sum_{v'=1}^{N_i} \alpha_{k_i v} + \sigma_{j_i v}(x_{-i})},$$

where $x_i = v$ denotes $x_{iv} = 1$ and k_i is a shorthand of k_{j_i}. This Gibbs sampling is a kind of a generalization of collapsed Gibbs sampling for LDA (latent Dirichlet allocation) (Griffiths et al. (2004)), however, actually this naive Gibbs sampling is usually useless since it might have unreachable states. For instance, suppose x consists of two values x_1 and x_2, and $f(x)$ takes 1 if $v_1 = v_2$ and 0 otherwise. Then, $p(x_1 = v \mid x_2, f, \alpha)$ equals to 1 if $v = v_2$ and 0 otherwise. So, in this case, a state change never happens.

A solution for the above problem is switching the target distribution from $p(x \mid f, \alpha)$ to $p(x, \theta \mid f, \alpha)$ and constructing a Gibbs sampling with two components x and θ. So, we alternately take samples of x and θ from the following conditional distributions:

$$p(\theta \mid x, f, \alpha) = p(\theta \mid x, \alpha) \qquad p(x \mid \theta, f, \alpha) = p(x \mid f, \theta).$$

A posterior distribution $p(\theta \mid x, \alpha)$ is a product of Dirichlet distributions as shown in 2.2, and a sampling algorithm from Dirichlet distributions has proposed (Gentle (2003)).

On the other hand, the conditional probability distribution $p(x \mid f, \theta)$ seems difficult to sample from since its computation generally requires exponential time. However, Ishihata et al. (2010) proposed an efficient algorithm for computing $p(x \mid f, \theta)$ via its PBPM $p(b \mid f, \theta)$

in a dynamic programming manner on a *binary decision diagram* (BDD) for f. In the same manner, we can efficiently take a sample of b using the BDD and also x by decoding the sampled b to x. The detail of the sampling algorithm for $p(b \mid f, \theta)$ on the BDD is described in the following section.

3.2. BDD-based sampling from $p(b \mid f, \theta)$

We here propose an efficient sampling algorithm from $p(b \mid f, \theta)$ using a BDD for f. First, we introduce a totally-order over b and use b_i to denote the i-th ordered variable. Then, a conditional probability $p(b{=}v \mid f, \theta)$ ($v{=}\{v_i\}_{i=1}^{|b|}$, $v_i{\in}\{0,1\}$) can be factorized into the following product:

$$p(b{=}v \mid f, \theta) = \prod_{i=1}^{|b|} p\left(b_i{=}v_i \mid f \wedge \bigwedge_{i'=1}^{i-1} b_{i'}{=}v_{i'}, \theta\right)$$

Thus, we can sample from $p(b \mid f, \theta)$ if each $p(b_i{=}v_i \mid f \wedge \bigwedge_{i'=1}^{i-1} b_{i'}{=}v_{i'}, \theta)$ is computable. To compute these conditional probabilities efficiently, we introduce a *binary decision diagram* (BDD) for f. A BDD (Akers (1978)) is a directed acyclic graph which compactly represents a boolean function. BDDs consist of two types of nodes, one is variable nodes and the other is terminal nodes. A variable node n is labeled by a boolean variable and has exactly two outgoing edges called 1-edge and 0-edge. We use b_n to denote n's label, and then n's u-edge ($u{\in}\{0,1\}$) represents b_n's assignment being u. So, a path in a BDD represents assignments for variables in the path. BDDs must have two terminal nodes, the 1-terminal t_1 and the 0-terminal t_0. A path from the root node to t_u ($u \in \{0,1\}$) in a BDD for f corresponds to a (partial) assignment for b such that $f(b){=}u$. The main idea of BDDs is based on recursive *Shannon expansion*. Let f_i be a boolean function of $b_i, \ldots, b_{|b|}$. Then, f_i can be factorized by b_i as follows:

$$f_i = (b_i \wedge f_{i|b_i=1}) \vee (\neg b_i \wedge f_{i|b_i=0}),$$

where $f_{i|b_i=1}$ (resp. $f_{i|b_i=0}$) is a positive (resp. negative) Shannon cofactor which is f_i with b_i set to 1 (resp. 0). So, $f_{i|b_i=v_i}$ ($v_i \in \{0,1\}$) is a boolean function of $b_{i+1}, \ldots, b_{|b|}$ and it can also be factorized by b_{i+1} into two Shannon cofactors consisting of $b_{i+2}, \ldots, b_{|b|}$. If BDDs for $f_{i|b_i=1}$ and $f_{i|b_i=0}$ are constructed, a BDD for f_i can be easily constructed by introducing a new root node labeled by b_i and its u-edge ($u \in \{0,1\}$) pointing a BDD for $f_{i|b_n=u}$. Consequently, a BDD for f can be constructed by applying Shannon expansion by b_i ($i = 1, \ldots, |b|$), recursively. Actually, an efficient algorithm for constructing BDDs has proposed (Bryant (1986)).

Let f_n be a function represented by a sub-BDD of which root node is n. Then, its *backward probability* $\mathrm{B}[n]{\equiv}p(f_n \mid \theta)$ can be computed recursively as follows:

$$\mathrm{B}[t_1] \equiv 1, \qquad \mathrm{B}[t_0] \equiv 0, \qquad \mathrm{B}[n] = \sum_{u \in \{0,1\}} p(b_n{=}u \mid \theta)\mathrm{B}[n_u],$$

where n_u is n's child node pointed by its u-edge. Since $p(b_n{=}u \mid \theta)\mathrm{B}[n_u]$ corresponds to a joint probability $p(b_n{=}u, f_n \mid \theta)$, a conditional probability $p(b_n{=}u \mid f_n, \theta)$ can be computed as $p(b_n{=}u \mid \theta)\mathrm{B}[n_u]/\mathrm{B}[n]$. Consequently, we can take a sample from $p(b \mid f, \theta)$ in a dynamic programming manner on a BDD for f as follows:

1. Construct a BDD for f.
2. Compute backward probabilities of nodes in the BDD.
3. Set n to the root node of the BDD.
4. Sample $u \in \{0,1\}$ as the value of b_n with probability $p(b_n = u \mid \theta)\mathrm{B}[n_u]/\mathrm{B}[n]$.
5. Update n to n_u and repeat 4 until n reaches the 1-terminal.

The time and space complexity of the above sampling is proportional to the BDD size. The BDD size strongly depends on the boolean function f and the totally-order over b. Unfortunately, in the worst case, the BDD size is exponential in $|b|$. However, Bryant (1986) showed many useful functions can be represented as BDDs with polynomial size. Furthermore, Ishihata et al. (2010) showed that the size of BDDs for HMMs (hidden Markov models) and the time complexity of an EM algorithm working on the BDDs are the same as the *Baum-Weltch* algorithm which is an EM algorithm specialized for HMMs.

4. Component-wise Metropolis-Hastings Sampling for PBPMs

Since the Gibbs sampling described in Section 3 takes sample of x and θ even though what we would like to get is only samples of x, the method is expected to be slower in convergence than direct sampling methods from $p(x \mid f, \alpha)$. In this section, we propose a sampling method directly from $p(x \mid f, \alpha)$ based on the Metropolis-Hasting (M-H) sampling. This sampling is a kind of application of a component-wise M-H sampling for PCFGs proposed by Johnson and Griffiths (2007) to PBPMs. The M-H sampling is an MCMC method for sampling from a target distribution $\pi(s)$ and constructs a Markov chain using a *proposal distribution* $Q(\tilde{s} \mid s)$ which is easy to sample from. It takes a sample $\tilde{s}$ from $Q(\tilde{s} \mid s)$ as a candidate of the next state, where s is the previous state, and accepts $\tilde{s}$ with probability $A(\tilde{s}, s)$ defined as

$$A(\tilde{s}, s) \equiv \min\{1, R(\tilde{s}, s)\}, \qquad R(\tilde{s}, s) \equiv \frac{\pi(\tilde{s})Q(s \mid \tilde{s})}{\pi(s)Q(\tilde{s} \mid s)}.$$

If $\tilde{s}$ is rejected, a state change does not happen. The M-H sampling for $p(x \mid f, \alpha)$ is easily constructed by employing $p(x \mid f, \hat{\theta})$ as a proposal distribution, where we call $\hat{\theta}$ a *production probability*. In this M-H sampling, we take a candidate $\tilde{x}$ from $p(x \mid f, \hat{\theta})$ in the same way as Section 3 and accept $\tilde{x}$ with probability $A(\tilde{x}, x)$ computed by the following $R(\tilde{x}, x)$:

$$R(\tilde{x}, x) = \frac{p(\tilde{x} \mid f, \alpha)p(x \mid f, \hat{\theta})}{p(x \mid f, \alpha)p(\tilde{x} \mid f, \hat{\theta})} = \frac{p(\tilde{x} \mid \alpha)p(x \mid \hat{\theta})}{p(x \mid \alpha)p(\tilde{x} \mid \hat{\theta})}.$$

The point here is that computing the marginal likelihood $p(f \mid \alpha)$, which is intractable but required to compute the target distribution $p(x \mid f, \alpha)$, is not required in the above computation. By substituting Equation (1) and (2), we have

$$\begin{aligned} R(\tilde{x}, x) &= \prod_{j=1} \frac{\mathrm{Z}\left(\alpha_{k_j v} + \sigma_{jv}(\tilde{x})\right)}{\mathrm{Z}\left(\alpha_{k_j v} + \sigma_{jv}(x)\right)} \prod_{v=1}^{M_j} \frac{\hat{\theta}_{jv}^{\sigma_{jv}(x)}}{\hat{\theta}_{jv}^{\sigma_{jv}(\tilde{x})}} \\ &= \prod_{j,v \,:\, \sigma_{jv}(x-\tilde{x}) \neq 0} \frac{\Gamma\left(\alpha_{k_j v} + \sigma_{jv}(\tilde{x})\right)}{\Gamma\left(\alpha_{k_j v} + \sigma_{jv}(x)\right)} \hat{\theta}_{jv}^{\sigma_{jv}(x-\tilde{x})}. \end{aligned}$$

The problem remained here is how to decide a production probability $\hat{\theta}$ of the proposal distribution $p(x \mid f, \hat{\theta})$. If we choose $\hat{\theta}$ randomly from the posterior distribution $p(\theta \mid x, \alpha)$ given the current sample x, this M-H sampling is most of same as the Gibbs sampling described in Section 3 but expected to be slower since rejections only happen in the M-H sampling.

To realize sampling with lower rejection, we here extend the above naive M-H sampling to a component-wise M-H sampling. We divide x into T components $x^{(1)}, \ldots, x^{(T)}$ ($x^{(t)} \subseteq x$, $1 \leq t \leq T$) and assume that a function $f(x)$ can be factorized as a product of T sub-functions $f^{(t)}(x^{(t)})$. Then, a base model $p(f, x \mid \theta)$ is also factorized as a product of T base models $p(f^{(t)}, x^{(t)} \mid \theta)$ with the common parameter θ such as i.i.d. observations. If a conditional distribution $p(x^{(t)} \mid x^{(-t)}, f^{(t)}, \alpha)$ were easily computable such as $p(x_i \mid x_{-i}, f, \alpha)$, we could construct a component-wise Gibbs sampling, where $x^{(-t)} = x \backslash x^{(t)}$. Unfortunately, $p(x^{(t)} \mid x^{(-t)}, f^{(t)}, \alpha)$ is intractable since the number of possible $x^{(t)}$ is generally exponential in $|x^{(t)}|$. However, as with the above M-H sampling for $p(x \mid f, \alpha)$, we can take a sample directly from $p(x^{(t)} \mid x^{(-t)}, f^{(t)}, \alpha)$ by an M-H sampling with a proposal distribution $p(x^{(t)} \mid f^{(t)}, \hat{\theta})$. To close the proposal distribution $p(x^{(t)} \mid f^{(t)}, \hat{\theta})$ to the target distribution $p(x^{(t)} \mid x^{(-t)}, f^{(t)}, \alpha)$, we set $\hat{\theta}$ to $\mathrm{E}[\theta]_{p(\theta \mid x^{(-t)}, \alpha)}$ which is the mean of the posterior distribution $p(\theta \mid x^{(-t)}, \alpha)$ given $x^{(-t)}$ computed as follows:

$$\hat{\theta}_{jv} = \frac{\alpha_{k_i v} + \sigma_{j_i v}(x^{(-t)})}{\sum_{v'=1}^{N_i} \alpha_{k_i v} + \sigma_{j_i v}(x^{(-t)})}.$$

So, the component-wise M-H sampling is constructed as follows:

1. Sample t from $\{1, \ldots, T\}$ uniformly.
2. Set $\hat{\theta}$ to $\mathrm{E}[\theta]_{p(\theta \mid x^{(t)}, \alpha)}$.
3. Sample a candidate $\hat{x}^{(t)}$ from the proposal distribution $p(x^{(t)} \mid f^{(t)}, \hat{\theta})$.
4. Accept $\hat{x}^{(t)}$ as new sample of $x^{(t)}$ with probability $A(\hat{x}^{(t)}, x^{(t)})$ defined by the following $R(\tilde{x}^{(t)}, x^{(t)})$:

$$R(\tilde{x}^{(t)}, x^{(t)}) \equiv \prod_{j,v \,:\, \sigma_{jv}(x^{(t)} - \tilde{x}^{(t)}) \neq 0} \frac{\Gamma\big(\alpha_{k_j v} + \sigma_{jv}(x^{(-t)}) + \sigma_{jv}(\tilde{x})\big)}{\Gamma\big(\alpha_{k_j v} + \sigma_{jv}(x^{(-t)}) + \sigma_{jv}(x)\big)} \theta_{jv}^{\sigma_{jv}(x^{(t)} - \tilde{x}^{(t)})}.$$

This component-wise M-H sampling is expected to converge faster than the Gibbs sampling in Section 3 since it updates the production probability $\hat{\theta}$ after each component sampled whereas the Gibbs sampling samples θ after all components sampled.

5. Experiments

5.1. LDA as statistical abduction

To show statistical abduction and our MCMC methods are general, we here formulate LDA (latent Dirichlet allocation) which is a well-known generative model for bag-of-words (Blei et al. (2003)) as statistical abduction on a PBPM, and apply them for finding topics on the LDA model.

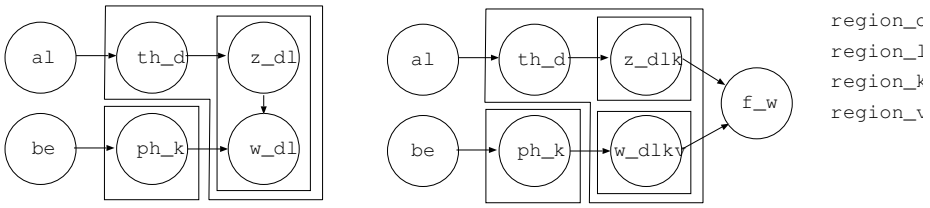

Figure 2: Graphical representations of LDA. The left one is of the original LDA and the right one is of a PBPM representing LDA.

Suppose we have D documents and their vocabulary consists of V words. LDA assumes that each document has a *topic distribution* and each topic has a *word distribution*. We use w_{dl} and z_{dl} to denote the l-th word of the d-th document and its *hidden* topic, respectively. Then, LDA defines a joint distribution of w_{dl} and z_{dl} as follows:

$$p(w_{dl}, z_{dl} \mid \theta, \phi) \equiv p(w_{dl} \mid z_{dl}, \phi) p(z_{dl} | \theta), \quad p(z_{dl} = k \mid \theta) \equiv \theta_{dk}, \quad p(w_{dl} = v \mid z_{dl} = k, \phi) \equiv \phi_{kv},$$

where $\theta_d \equiv \{\theta_{dk}\}_{k=1}^K$ and $\phi_k \equiv \{\phi_{kv}\}_{v=1}^V$ are parameters of the topic distribution of the d-th document and the word distribution of the k-th topic, respectively. In addition, LDA assumes parameters θ_d and ϕ_k are generated from Dirichlet distributions with parameters α and β, respectively. The graphical representation of LDA is shown at the left in Fig. 2.

Given a bag-of-words $w \equiv \{w_{dl} \mid 1 \leq d \leq D, 1 \leq l \leq L_d\}$ and parameters α and β, our task is to infer the most probable topic z_{dl} corresponding to each word w_{dl}. Now, we introduce new discrete random variables w_{dlk} with probability $p(w_{dlk} = v \mid \phi) \equiv \phi_{kv}$ corresponding to the conditional probability $p(w_{dl} = v \mid z_{dl} = k, \phi)$. Then, a probabilistic event f_w representing that "a bag-of-words w is observed" is described as the following boolean formula:

$$f_w = \bigwedge_{d=1}^{D} \bigwedge_{l=1}^{L_d} \text{“}w_{dl} = v_{dl}\text{”}, \qquad \text{“}w_{dl} = v_{dl}\text{”} \equiv \bigvee_{k=1}^{K} \text{“}z_{dl} = k\text{”} \wedge \text{“}w_{dlk} = v_{dl}\text{”},$$

where v_{dl} is the word ID corresponding to the observed w_{dl} in w. As shown in 2.1, probabilistic events "$z_{dl} = k$" and "$w_{dlk} = v$" can be represented as boolean formulas in independent boolean random variables z_{dlk} and w_{dlkv}, respectively. So, an LDA model can be described as a PBPM of which graphical representation is at the right in Fig. 2, and finding the most probable topics on LDA models is formulated as statistical abduction on PBPMs.

We generated a small dataset by the same way as Griffiths et al. (2004). The dataset consisted of 1,000 documents and the vocabulary was 25 words. Each document can be represented as a 5×5 grid image, where the intensity of the i-th pixel corresponds to the count of the i-th word in the document. The documents were generated by sampling words from 10 topics corresponding to horizontal and vertical lines as shown in Fig. 3. A word distribution of a topic was a uniform distribution over its line, and topic distributions of documents were sampled from a Dirichlet distribution with $\alpha = 1$. Every document contained 100 words and their subset is shown in Fig. 3.

We applied the Gibbs sampling (GS) proposed in Section 3 and the component-wise M-H sampling (CMHS) proposed in Section 4 to this dataset, together with the collapsed Gibbs

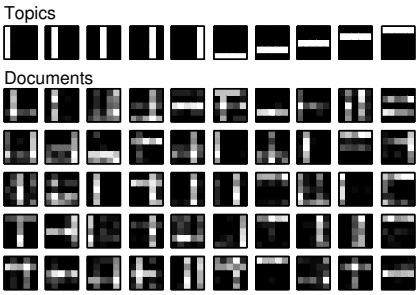

Figure 3: Dataset

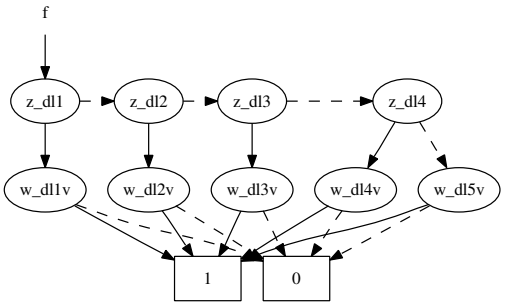

Figure 4: BDD for "$w_{dl} = v$"

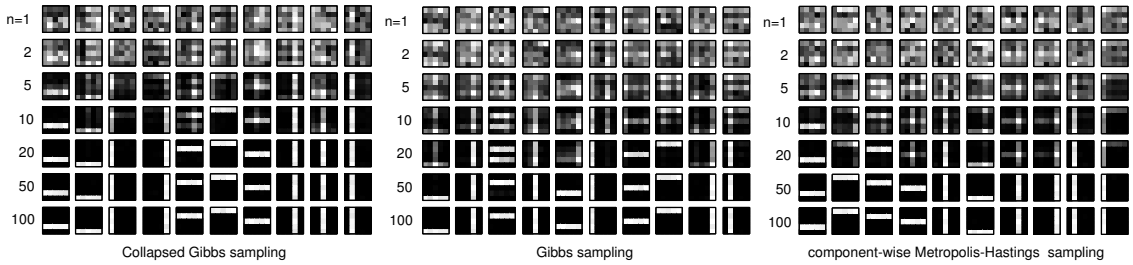

Figure 5: Obtained topics

sampling (CGS) for LDA (Griffiths et al. (2004)). In CMHS, we divided a boolean function f_w into 100,000 components "$w_{dl}=v_{dl}$" corresponding to one word in the dataset. Fig. 4 shows a BDD for "$w_{dl}=v$" with $K = 5$ and the BDD size is proportional to K. So, we can sample a topic z_{dl} corresponding to word w_{dl} in $O(K)$ time. We defined one step of CMHS as sampling 100,000 topics since GS and CGS sample 100,000 topics in a step. We ran each sampling method 100 steps and repeated 100 times. The results of these computations are shown in Fig. 5 and 6, and they show estimated topic distribution and convergence of log likelihood of each method, respectively. In the results, θ_d and ϕ_k are estimated by the mean of their posterior distributions given the set of samples. The results show all three methods are able to recover the underlying topics, and CMHS quickly stabilizes than GS. However, CMHS is slower than CGS since CMHS sometimes rejects candidates.

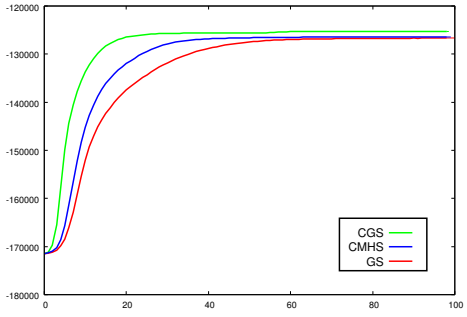

Figure 6: Convergence of log likelihoods

5.2. Diagnosis for failure in logic circuits as statistical abduction

To compare the performance of Bayesian inference and MLE (most likelihood estimation), we here applied our method to a stochastic error finding problem, together with a method which performs MLE.

Poole (1993) formulated a stochastic error finding problem in logic circuits as statistical abduction. In the formulation, an error gate is *stochastically* stuck at 0 or 1, and the task is to predict where error gates are in a target logic circuit given its structure (knowledge base) and pairs of input and output values (observations). We here use PRISM (Sato and Kameya (2001)), which is a Prolog-based probabilistic modeling language for statistical abduction, to describe a structure of the target logic circuit and derive boolean formulas of the observations.

We first introduce two predicates $type(G,T)$ and $conn(P,Q)$ to describe a structure of the target circuit, where $type(G,T)$ defines the type of a gate G as T, and $conn(P,Q)$ represents that two ports P and Q are connected. For instance, a logic circuit c representing a boolean function $(b_1 \wedge b_2) \vee b_3$ is described as the following PRISM programs:

$$
\begin{array}{lll}
type(g1, and). & type(g2, or). & \\
conn(in(c,1), in(g1,1)). & conn(in(c,2), in(g1,2)). & conn(in(c,3), in(g2,1)). \\
conn(out(g1), in(g2,2)). & conn(out(g2), out(c)). &
\end{array}
$$

where $in(G,N)$ and $out(G)$ denote the N-th input port and the output port of G, respectively. We next introduce a predicate $val(P,V)$ to represent the value of a port P being V. Then, the function of a gate G is described as the following PRISM programs:

$$
\begin{aligned}
func(G,V) :- & type(G,T), (T = or, or(G,V)\ ;\ T = and, and(G,V)).\\
or(G,V) :- & (val(in(G,1),1), V=1\\
& ; val(in(G,2),1), V=1\\
& ; val(in(G,1),0), val(in(G,2),0), V=0).\\
and(G,V) :- & (val(in(G,1),0), V=0\\
& ; val(in(G,2),0), V=0\\
& ; val(in(G,1),1), val(in(G,2),1), V=1).
\end{aligned}
$$

In this problem setting, some gates might be error and error gates are stochastically stuck at 0 or 1. To handle such uncertainty, PRISM has a particular predicate $msw(S,V)$ which represents a probabilistic switch S taking a value V. We now introduce a probabilistic switch $st(G)$ corresponding to the state of a gate G and takes one of three values $\{ok, stk0, stk1\}$. Using $st(G)$, a predicate $val(P,V)$ is defined as follows:

$$
\begin{aligned}
val(Q,V) :- & \ conn(P,Q), val(P,V).\\
val(out(G),V) :- & \ msw(st(G),S),\\
& (\ S = ok, func(G,V)\\
& ;\ S = stk0, V=0\\
& ;\ S = stk1, V=1).
\end{aligned}
$$

The above PRISM programs can derive boolean formulas of observations in $msw(st(G), V)$, where an observation is a pair of input and output values of the target circuit. For instance, suppose we set (0, 0, 0) to the inputs of the target circuit c and observed its output being 1. Then, PRISM derives a boolean formula $msw(st(g1), stk1) \vee msw(st(g2), stk1)$ as the explanation of the observation.

Given the boolean formulas of observations, we would like to predict which gate is an error. In this paper, we additionally assume that non-deterministic knowledge such as "*or* gates tend to be stuck at 1" and "*and* gates tend to be stuck at 0" are given as prior knowledge. Such non-deterministic knowledge seems difficult to describe by logic, however, it can be reflected in the model as prior distributions by introducing Bayesian inference for statistical abduction. PRISM has a built-in method for performing Bayesian inference, however, it assumes that disjuncts in explanations are probabilistically exclusive. Unfortunately, the explanations derived by the above PRISM programs does not necessarily satisfy the assumption. So, in this paper, we use our MCMC method instead of PRISM built-in method to perform Bayesian inference for this problem.

In this experiment, we applied our CMHS to predicting errors in a 3-bit adder circuit, together with the *BO-EM* algorithm (Ishihata et al. (2010)) which is an EM algorithm based on BDDs, and compared their predicting accuracy. (Actually, we also applied GS but omit the result since it was almost same as CMHS.) A 3-bit adder consists of 12 gates $g_1, \ldots, g_{12}$ (5 *and*, 5 *xor* and 2 *or* gates). We use θ_{i1}, θ_{i2} and θ_{i3} to denote the probability of $st(g_i)$ taking ok, $stk0$ and $stk1$, respectively. So $\theta_i = \{\theta_{iv}\}_{v=1}^{3}$ defines the distribution of $st(g_i)$. We randomly generated 1,000 3-bit adders with mixing error gates with probability 0.1, where the distribution of each g_i was defined as

- If g_i is not an error gate, $(\theta_{i1}, \theta_{i2}, \theta_{i3}) = (1, 0, 0)$,
- If g_i is an error *xor*/*or* gate, $(\theta_{i1}, \theta_{i2}, \theta_{i3}) = (0, 0.1, 0.9)$,
- If g_i is an error *and* gate, $(\theta_{i1}, \theta_{i2}, \theta_{i3}) = (0, 0.9, 0.1)$,

and sampled N ($N = 20, 40, 60, 80, 100$) input and output pairs from each circuit. So, the average distribution of *xor*/*or* gates and that of *and* gates were $(0.9, 0.01, 0.09)$ and $(0.9, 0.09, 0.01)$, respectively. To reflect these knowledge to statistical abduction, we introduced two Dirichlet distribution with parameters $(\alpha_{11}, \alpha_{12}, \alpha_{13}) = (0.9, 0.01, 0.09)$ and $(\alpha_{21}, \alpha_{22}, \alpha_{23}) = (0.9, 0.09, 0.01)$, and assumed *xor*/*or* gates were generated from the first one and *and* gates were from the second one. Given these prior distributions and boolean formulas of observations derived by the above PRISM programs, we estimated each θ_i using a single sample taken after 100 iterations of CMHS. Using the estimated parameter θ_i, we predicted g_i as error if θ_{i1} was smaller than the threshold decided to maximize the F-measure for the test set with $N = 20$. The left side in Fig. 7 depicts the precision, recall and F-measure of BO-EM as function of the number of observations N, and the left one depicts those of CMHS. The results shows CMHS achieved better F-measures than BO-EM in every N and also shows that introducing non-deterministic knowledge as prior distributions is efficient in prediction of stochastic error in logic circuit.

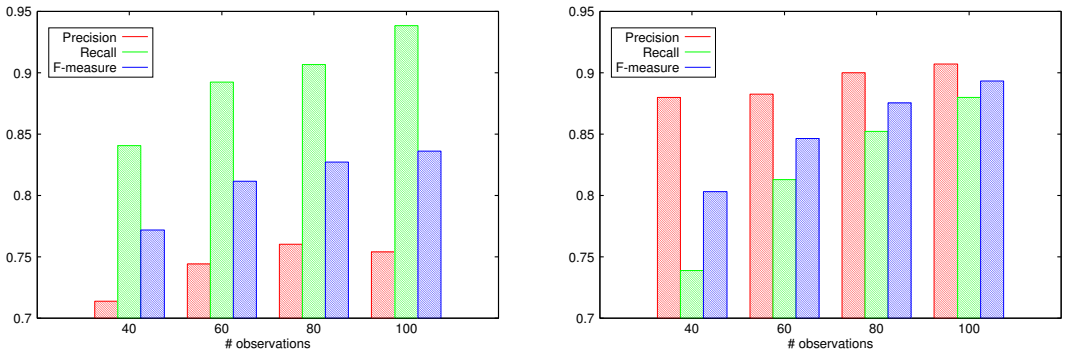

Figure 7: Predicting accuracy of BO-EM (left) and CMHS (right)

6. Related work and Future work

Johnson and Griffiths (2007) proposed a component-wise Metropolis-Hastings algorithm for PCFGs (probabilistic context free grammars), and recently Sato (2011) generalized the method for a logic-based probabilistic modeling language PRISM (Sato and Kameya (2001)). Our method is an application of their methods to PBPMs, however, it differs a great deal in that it has no restriction on formulas whereas PRISM assumes disjuncts in formulas are probabilistically exclusive.

ProbLog (De Raedt et al. (2007)) is a recent probabilistic extension of Prolog and employs a BDD-based probability learning method called CoPrEM algorithm (Gutmann et al. (2010)). However, to the best of our knowledge, Bayesian inference for ProbLog has not been proposed yet. It would be interesting to apply our methods for Bayesian inference on ProbLog.

More recently, a couple of new statistical abduction frameworks have been proposed. Raghavan and Mooney (2011) proposed BALPs (Bayesian abductive logic programs) which integrates BLPs (Bayesian logic programs) (Kersting and De Raedt (2001)) and abduction, and Singla and Mooney (2011) combined abductive inference and MLNs (Markov logic networks) (Richardson and Domingos (2006)). BLPs and MLNs are similar in that they can be considered as templates for constructing graphical models, the former defines Bayesian networks and the latter Markov random fields. The difference of those methods and our methods is that BALPs employ EM learning and a general learning method for abductive Markov logic has not been proposed yet.

Variational Bayes (VB) inference is another approximation method for performing Bayesian inference, and the VB-EM algorithm (Beal and Ghahramani (2003)) is known as an EM like iterative computation for VB inference. Ishihata et al. (2011) generalized the BO-EM algorithm (Ishihata et al. (2010)), which is an EM algorithm working on BDDs, to the VB-EM algorithm and applied it to statistical abduction. Comparing the performance of their method with that of our MCMC methods is a future work.

Inoue et al. (2009) and Synnaeve et al. (2011) applied the BO-EM algorithm to evaluating abductive hypotheses about metabolic pathway. Replacing BO-EM with our method enables us to perform Bayesian inference for their problem and allows us to introduce non-deterministic knowledge such as preference and/or frequency of chemical reactions.

7. Conclusion

We proposed two MCMC methods for performing Bayesian inference for statistical abduction. As a component of those algorithms, we derived an efficient sampling algorithm based on dynamic programming on BDDs. To demonstrate the framework of statistical abduction and our methods are general, we described LDA which is a well-known generative model for bag-of-words as statistical abduction and applied our methods to it. Then, we used our methods to predict stochastic errors in logic circuit and showed their performances are better than the method based on maximum likelihood estimation.

Acknowledgement

This work was partly supported by Grant-in-Aid for Science Research from the Japan Society for the Promotion of Science Fellows 22-7115.

References

Sheldon B. Akers. Binary decision diagrams. *IEEE Transaction on Computers*, 27(6): 509–516, 1978.

M.J. Beal and Z. Ghahramani. The Variational Bayesian EM Algorithm for Incomplete Data: with Application to Scoring Graphical Model Structures. *Bayesian Statistics*, 7, 2003.

David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

Randal E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transaction on Computers*, 35(8):677–691, 1986.

Luc De Raedt, Angelika Kimming, and Hannu Toivonen. ProbLog: A probabilistic Prolog and its application in link discovery. In *Proc. of IJCAI'07*, 2007.

J.E. Gentle. *Random number generation and Monte Carlo methods*. Springer, second edition, 2003.

Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007. ISBN 0262072882.

Thomas L. Griffiths, Mark Steyvers, Thomas L. Griffiths, and Mark Steyvers. Finding scientific topics, 2004.

Bernd Gutmann, Ingo Thon, and Luc De Raedt. Learning the parameters of probabilistic logic programs from interpretations. Department of Computer Science, K.U.Leuven, 2010.

Katsumi Inoue, Taisuke Sato, Masakazu Ishihata, Yoshitaka Kameya, and Hidetomo Nabeshima. Evaluating abductive hypotheses using an EM algorithm on BDDs. In *Proc. of IJCAI'09*, 2009.

Masakazu Ishihata, Yoshitaka Kameya, Taisuke Sato, and Shin-ichi Minato. An EM algorithm on BDDs with order encoding for logic-based probabilistic models. In *Proc. of ACML'10*, 2010.

Masakazu Ishihata, Yoshitaka Kameya, and Taisuke Sato. Variational Baye inference for logic-based probabilistic models on BDDs. *Presented at: ILP'11*, 2011.

Mark Johnson and Thomas L. Griffiths. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proc. of NAACL07*, 2007.

Rohit J. Kate and Raymond J. Mooney. Probabilistic Abduction using Markov Logic Networks. In *Proc. of IJCAI'09 Workshop on PAIR'09*, 2009.

Kristian Kersting and Luc De Raedt. Towards Combining Inductive Logic Programming with Bayesian Networks. In *Proc. of ILP'01*, 2001.

Stephen Muggleton. Stochastic Logic Programs. In *New Generation Computing*. Academic Press, 1996.

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

David Poole. The Independent Choice Logic for modelling multiple agents under uncertainty. *Artificial Intelligence*, 94:7–56, 1997.

Sindhu Raghavan and Raymond Mooney. Bayesian Abductive Logic Programs. In *Proc. of AAAI'10 Workshop on Star-AI'10*, 2010.

Sindhu Raghavan and Raymond Mooney. Abductive Plan Recognition by Extending Bayesian Logic Programs. In *Proc. of ECML/PKDD'11*, 2011.

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

Taisuke Sato. A General MCMC Method for Bayesian Inference in Logic-based Probabilistic Modeling. In *Proc. of IJCAI'11*, 2011.

Taisuke Sato and Yoshitaka Kameya. Parameter Learning of Logic Programs for Symbolic-statistical Modeling. *Journal of Artificial Intelligence Research*, 15:391–454, 2001.

Taisuke Sato, Yoshitaka Kameya, and Ken-ichi Kurihara. Variational Bayes via propositionalized probability computation in PRISM. *Annals of Mathematics and Artificial Inteligence*, 54(1-3):135–158, 2009.

Parag Singla and Raymond J. Mooney. Abductive Markov Logic for Plan Recognition. In *Proc. of AAAI'11*, 2011.

Gabriel Synnaeve, Katsumi Inoue, Andrei Doncescu, Hidetomo Nabeshima, Yoshitaka Kameya, Masakazu Ishihata, and Taisuke Sato. Kinetic Models and Qualitative Abstraction for Relational Learning in Systems Biology. In *BIOSTEC Bioinformatics 2011*, 2011.