

# 論理に基づいた確率モデルに於ける一般的MCMC法による ベイズ推論について

## Bayesian Inference in Logic-based Probabilistic Modeling by a General MCMC Method

佐藤泰介<sup>1\*</sup>

<sup>1</sup> 東京工業大学大学院情報理工学研究科

<sup>1</sup> Tokyo Institute of Technology

**Abstract:** We propose a general MCMC method for Bayesian inference in logic-based probabilistic modeling. It covers a broad class of generative models including Bayesian networks and PCFGs. The idea is to generalize an MCMC method for PCFGs to the one for a Turing-complete probabilistic modeling language PRISM in the context of statistical abduction where parse trees are replaced with explanations. We show examples of the marginal probability computation from MCMC samples and Bayesian Viterbi inference using a Naive Bayes model augmented with a hidden class variable.

## 1 はじめに

2000年代の初頭から米国を中心としてBN(Bayesian network, ベイジアンネットワーク)に関係概念を取り入れる事により、より複雑な確率モデリングを可能にする研究の流れがありSRL(statistical relational learning, 統計的関係学習)と呼ばれて来た[1]。一方ヨーロッパでも同じ頃論理に基づいた構造学習であるILP(inductive logic programming, 帰納論理プログラミング)に確率を導入することにより不確実性も扱えるようにしようという研究の流れがあり、PLL(probabilistic logic learning, 確率論理学習)と呼ばれて来た[2]。両者は関係概念と確率を媒介として融合しSRL/PLLあるいは単にSRLと呼ばれている。現在SRL/PLLには多様なアプローチがあるが、確率化されたプログラミング言語により大規模で複雑な構造を持った対象/事象の確率モデリングを容易にしようというアプローチが大半を占める。また土台となるプログラミング言語も手続き型、論理型、関数型、オブジェクト指向型など多種多様である[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]。確率プログラミング言語では、プログラムとして確率モデルを記述し、言語処理系が提供する確率推論機能やパラメータ学習機能により確率を推論しあるいは確率をデータから学習する。本論文では論理型プログラミング言語の代表であるPrologを確率化したPRISMと

いう確率モデリング言語<sup>1</sup>[3, 7]のための汎用のMCMC(Markov chain Monte Carlo, マルコフ連鎖モンテカルロ)法を今年の夏にIJCAI11で発表した論文[15]の内容を交えて紹介する。

## 2 PRISMの概略

PRISM(PROgramming In Statistical Modeling)はPrologに確率的選択を表す組み込み述語`msw/2`と付随するパラメータの学習機構を加えたものである。PRISMでは通常のPrologのプログラム節のボディに普通のゴールに加えて、確率的選択を行う`msw(i, v)`という形のゴールを書くことが出来る。`msw(i, v)`は $i$ という名前のサイコロを振ったら $v(v \in V_i)$ という目が出たという有限個の選択肢 $V_i$ からの確率的選択事象を表す。`msw(i, v)`が真になる確率 $\theta_{i,v}$ をパラメータと呼び $\sum_{v \in V_i} \theta_{i,v} = 1$ が成立する。またPRISMでは $\{\theta_{i,v}\}_{v \in V_i}$ の事前分布として超パラメータ $\{\alpha_{i,v}\}_{v \in V_i}$ を持つディリクレ分布(Dirichlet distribution)を設定できる。パラメータ、超パラメータは`set_sw/2, 3`などの組み込み述語で値を設定するか、後述のように`learn/1, 2`という組み込み述語を使いデータから統計的に学習する。

PRISMによる確率モデリングは生成的である。`msw/2`を含んだ節を複数組み合わせることによりPCFGに於ける文生成のような複雑な確率的データ生成過程をプログラムする。生成されたデータはモデリングしたい

\*連絡先: 東京工業大学  
東京都目黒区大岡山2-12-1  
E-mail: sato@mi.cs.titech.ac.jp

<sup>1</sup><http://mi.cs.titech.ac.jp/prism/>

観測事象を表すアトム（観測アトムと言う）の形で出力される。学習の場合は逆に観測アトム  $G$  をトップゴールとして受け取り、Prolog のトップダウン探索機能とテープリングと呼ばれる解の記憶機能を組み合わせた探索により、 $G$  の説明 と呼ばれる  $msw$  の論理積  $\epsilon$  ( $DB, \epsilon \vdash G$  である) の集合をグラフ構造としてコンパクトに組織化した説明グラフと呼ばれるブル式を構築し、説明グラフに各種パラメータ推定アルゴリズムを適用して  $msw$  のパラメータを推定する [7]。

PRISM は Prolog の一般化として汎用のプログラミング言語であると同時に汎用の生成的モデルのモデリング言語であり、BN、HMM(hidden Markov model, 隠れマルコフモデル)、PCFG (probabilistic context free grammar, 確率文脈自由文法) と呼ばれた各種生成的確率モデル (generative model) が定義する分布を観測アトムの分布として記述する。

PRISM プログラムを  $DB$  としよう。  $DB$  に現れる  $msw$  のパラメータ全体を  $\theta$ 、超パラメータ全体を  $\alpha$  で表す。現在 PRISM 処理系は  $DB$  に対し以下のモデリング機能を提供している。

**Sampling:**  $DB$  が定義する確率分布  $P_{DB}(\cdot | \theta)$  から観測アトムをサンプリングする。BN に於ける forward sampling に対応する。

**確率推論:** 観測アトム  $G$  の確率  $P_{DB}(G | \theta)$  を gIO (generalized Inside-Outside, 一般化された内側外側確率計算) アルゴリズムにより計算する。観測アトム  $G$  を適切に設定することにより同時確率、周辺確率、条件付き確率が計算される。

**Viterbi 推論:** 観測アトム  $G$  の説明の内最大確率を持つ説明 (Viterbi 説明)  $\epsilon^*$  を推論する。  $G$  の説明の全体を  $\text{expl}(G)$  と置く。パラメータ  $\theta$  を使い、  $\epsilon^* = \text{argmax}_{\epsilon \in \text{expl}(G)} P_{DB}(\epsilon | G, \theta)$  により Viterbi 説明  $\epsilon^*$  を求める方法と、事後分布  $P_{DB}(\theta | G, \alpha)$  から定まるベイズ Viterbi 説明  $\epsilon^* = \text{argmax}_{\epsilon \in \text{expl}(G)} P_{DB}(\epsilon | G, \alpha)$  を変分ベイズにより近似的に求める方法がある。

**パラメータ学習:** 観測アトムの列  $\mathcal{G}$  から  $msw$  のパラメータ  $\theta$  または超パラメータ  $\alpha$  を統計的推論により学習する。学習法として EM(expectation-maximization) アルゴリズムが使われており目的に応じて

- EM (尤度  $P_{DB}(\mathcal{G} | \theta)$  を最大化する  $\theta$  を求める EM)
- MAP-EM (事後確率  $P_{DB}(\mathcal{G} | \theta)P(\theta | \alpha)$  を最大化する  $\theta$  を求める EM)
- DAEM (EM が局所解に陥るのを避けるために焼き鈍しを取り入れた EM)

- VB-EM (近似的に事後分布  $P_{DB}(\theta | \mathcal{G}, \alpha)$  を求める変分ベイズ用 EM)

が利用可能である。

**モデルスコア:** ベイズの立場から周辺尤度を基準としたモデル選択を行うため、計算困難な周辺尤度の近似として

- BIC (Bayesian information criterion)
- CS (Cheeseman-Stutz) スコア
- VFE (variational free energy, 変分自由エネルギー)

を計算する。

**補助述語:** その他平均や分散の計算、一様分布や正規分布からサンプリング、log-gamma 関数の計算などを行う便利な組み込み述語が多数用意されている。

PRISM の gIO アルゴリズムは説明グラフを使うことで、HMM の前向き後ろ向きアルゴリズムや PCFG の内側外側確率計算などに見られるダイナミックプログラミングによる確率計算を (同一の時間計算量で) 一般化した形で実現している。BN の場合接合木を PRISM プログラムで記述し、gIO アルゴリズムを適用することにより接合木上の BP (信念伝播) が実現される [16]。BN で標準的な bucket elimination[17] の実現も容易である。

### 3 ベイズ推論と PRISM

統計的推論の中で重要な推論としてベイズ推論がある。隠れ変数  $x$ 、観測変数  $y$ 、パラメータ  $\theta$  を持つ確率モデル  $P(x, y | \theta)$  を考えよう。ベイズ推論ではパラメータの事前分布 (prior distribution)  $P(\theta)$  を導入し、データ  $y$  の周辺尤度 (marginal likelihood)  $P(y) \propto \int \sum_x P(x, y | \theta)P(\theta)d\theta$ 、 $y$  が観測された後の事後分布 (a posteriori distribution)  $P(\theta | y) \propto \int \sum_x P(x, y | \theta)P(\theta)d\theta$ 、新しいデータ  $y'$  に対するベイズ予測分布 (predictive distribution)  $P(x | y') = \int P(x | \theta, y')P(\theta | y)d\theta$  を使ったベイズ Viterbi 解  $x^* = \text{argmax}_x P(x | y')$  などを計算する。事前分布は人工知能に於ける個人の信念の度合いを表現手段として、あるいは機械学習に於ける過学習を防ぐ手段として、周辺尤度はモデル選択の基準として、事後分布はパラメータの点 / 区間推定に、予測分布は判別問題に使われる。

ベイズ推論に於ける計算の困難は周辺尤度ならびに事後分布の計算にあり、多くの場合事後分布の積分が閉じた式にならず、各種 MCMC や変分ベイズ (variational

Bayes) などの近似計算法を使う。PRISM ではベイズ推論のため  $\text{msw}/2$  が表すカテゴリカル分布 (categorical distribution, Bernoulli 分布の一般化) の事前分布としてディリクレ分布を導入し、変分ベイズ法により近似的な周辺尤度 (VFE) や事後分布の計算を行っている [18]。

確率変数  $X$  のとる値を  $V_X = \{v_1, \dots, v_m\}$  としよう。対応する  $\text{msw}$  アトム  $\{\text{msw}('X', v_1), \dots, \text{msw}('X', v_m)\}$  のパラメータを  $\theta_{v_1}, \dots, \theta_{v_m}$  ( $\sum_{k=1}^m \theta_{v_k} = 1$ ) と置く。 $\theta_{v_1}, \dots, \theta_{v_m}$  の事前ディリクレ分布  $P_{\text{Dir}}(\cdot)$  は  $\Gamma(\cdot)$  をガンマ関数として

$$\begin{aligned} P_{\text{Dir}}(\theta_{v_1}, \dots, \theta_{v_m} \mid \alpha_{v_1}, \dots, \alpha_{v_m}) \\ &= \frac{1}{Z} \prod_{v \in V_X} \theta_v^{\alpha_v - 1} \\ Z &= \frac{\prod_{v \in V_X} \Gamma(\alpha_v)}{\Gamma(\sum_{v \in V_X} \alpha_v)} \end{aligned}$$

により与えられる。パラメータの平均値は  $P_{\text{Dir}}(\cdot)$  の超パラメータ  $\{\alpha_{v_1}, \dots, \alpha_{v_m}\}$  により  $(\alpha_{v_1}/\alpha_0, \dots, \alpha_{v_m}/\alpha_0)$  ( $\alpha_0 = \sum_{k=1}^m \alpha_{v_k}$ ) となる。 $P_{\text{Dir}}(\cdot)$  の超パラメータが大きければ大きいほどパラメータの分散は小さくなる。

現在の SRL/PLL で変分ベイズ法を実装しているシステムは PRISM 以外に見当たらない。また CS スコア や VFE のモデルスコアの計算機能を提供しているものも PRISM 以外に見当たらない。PRISM の提供する変分ベイズ推論を使うことより BN や PCFG など知られたモデル以外のモデルに対しても周辺尤度、事後分布、ベイズ Viterbi 説明が近似的ではあるが能率よく求めることができる [18]。また PRISM の提供する変分ベイズ推論は隠れ変数がある場合のベイズスコアに基づく構造学習を容易にしている。

## 4 MCMC

変分ベイズによるベイズ推論は決定性の計算であり、サンプリングに基づいた方法よりはるかに高速であるが、近似計算には違いないので、もし本当に (例えば) 周辺尤度の値を知りたいければサンプリングに基づいた方法を使わざるを得ない。BN のような (離散的) グラフィカルモデルの場合 Gibbs サンプリングなどの MCMC 法が良く知られているが、ルールベースのより複雑な組み合わせの構造を持つ PCFG に対する MCMC 法は最近まであまり知られていなかった。しかし最近になり統計的自然言語処理分野で Mark Johnson 等が PCFG に対する MCMC を提案した [19]。彼らは PCFG の生成ルール選択にディリクレ事前分布を導入し、コーパス中の文の構文木の集まりを状態とみなし、コーパスを

与えた場合の PCFG の事後分布から M-H (Metropolis-Hasitngs) 法により構文木をサンプリングしてみせた。

彼らの PCFG 用の MCMC 法を PRISM に一般化するの容易である。図 1 の対応により、PCFG 用の MCMC 法に於ける文をゴールに、構文木を説明に置き換えれば、ほぼそのまま PRISM 用の MCMC 法になる。

	PCFG	PRISM
観測変数	文	ゴール
隠れ変数	構文木	説明 (証明木と同等)
記述	CFG 規則	プログラム
確率計算	IO アルゴリズム	gIO アルゴリズム

図 1: PCFG と PRISM の対応

プログラム  $DB$  により分布  $P_{DB}(\cdot)$  が定義されているとしよう。PRISM 用の M-H (Metropolis-Hasitngs) サンプリングは観測されたアトムの列  $\mathcal{G} = G_1, \dots, G_T$  に対し、対応する説明列  $\mathcal{E}^{(k)} = \epsilon_1^{(k)}, \dots, \epsilon_T^{(k)}$  を ( $k = 1, 2, \dots$ ) を状態とする Markov chain を作り出す。具体的には各ステップで (ステップ  $k$  とする) アトム列  $\mathcal{G}$  からランダムにアトムを一つ ( $G_t$  とする) を選び、ステップ  $k$  の状態  $\mathcal{E}^{(k)}$  における  $G_t$  の説明  $\epsilon_t^{(k)}$  を  $P_{DB}(\epsilon'_t \mid G_t, \bar{\theta}_{-t}^{(k)})$  からサンプルした新しい説明で置き換えることを繰り返す (図 2 参照)。技術的なポイントは説明  $\epsilon_t^{(k)}$  のサンプリングを [19] に習い、PRISM の一般化された内側確率計算によりダイナミックプログラミングを用いて効率的に行うところにある。 $\mathcal{E}_{-t}^{(k)}$  は説明列  $\mathcal{E}^{(k)}$  から  $\epsilon_t^{(k)}$  を削除した残り、 $\bar{\theta}_{-t}^{(k)} = \mathbf{E}[\theta \mid \mathcal{E}_{-t}^{(k)}, \alpha]$  は  $\mathcal{E}_{-t}^{(k)}$  により条件付けられたパラメータの条件付き期待値である。

[Initialize] Set  $\mathcal{E}^{(0)} = \epsilon_1^{(0)}, \dots, \epsilon_T^{(0)}$  for some  $\epsilon_t^{(0)}$ s.

[Repeat] for  $k = 0, 1, \dots$

Let  $\mathcal{E}^{(k)} = \epsilon_1^{(k)}, \dots, \epsilon_T^{(k)}$  be the current state.

Choose randomly  $t$  in  $[1, T]$ .

Sample a new explanation  $\epsilon'_t \sim P_{DB}(\epsilon'_t \mid G_t, \bar{\theta}_{-t}^{(k)})$ .

Let  $\mathcal{E}'$  be  $\mathcal{E}^{(k)}$  with  $\epsilon_t^{(k)}$  replaced by  $\epsilon'_t$ .

Compute the acceptance probability

$$A = \min \left( 1, \frac{P_{DB}(\mathcal{E} \mid \mathcal{E}', \alpha) P_{DB}(\mathcal{E}' \mid \mathcal{G}, \bar{\theta}_{-t}^{(k)}, \alpha)}{P_{DB}(\mathcal{E}' \mid \mathcal{E}, \alpha) P_{DB}(\mathcal{E} \mid \mathcal{G}, \bar{\theta}_{-t}^{(k)}, \alpha)} \right).$$

Move to a new state  $\mathcal{E}^{(k+1)} = \mathcal{E}'$  with probability  $A$  else  $\mathcal{E}^{(k+1)} = \mathcal{E}^{(k)}$ .

図 2: M-H sampling for PRISM

一般に MCMC に於いて Markov chain の初期状態

の選択は重要である。PRISMでは初期状態として変分ベイズにより計算した近似事後分布  $P_{DB}(\theta | \mathcal{G})$  から得られるパラメータの平均値  $\bar{\theta}_{VB}$  を使った Viterbi 解、すなわち  $\epsilon_t^{(0)} = \operatorname{argmax}_{\epsilon_t \in \operatorname{expl}(G_t)} P_{DB}(\epsilon_t | G_t, \bar{\theta}_{VB})$  ( $1 \leq t \leq T$ ) を使っている。このようにする事で Markov chain の遷移をある程度周辺尤度の高い状態から始めることができる。

観測アトム列  $\mathcal{G} = G_1, \dots, G_T$  から図 2 の M-H サンプルングを繰り返すことにより説明列のサンプル  $\{\mathcal{E}^{(k)}\}_{k=1}^m$  が得られているとしよう。近似的に  $\mathcal{E}^{(k)} \sim P_{DB}(\mathcal{E} | \mathcal{G}, \alpha)$  が成立している。周辺尤度  $P_{DB}(\mathcal{G} | \alpha)$  は  $\{\mathcal{E}^{(k)}\}_{k=1}^m$  を使い以下のように近似計算される [18]。

$$\begin{aligned} P_{DB}(\mathcal{G} | \alpha) &= \frac{P_{DB}(\theta, \mathcal{G} | \alpha)}{P_{DB}(\theta | \mathcal{G}, \alpha)} \text{ for arbitrary } \theta \\ &= \frac{P_{\text{Dir}}(\theta | \alpha) P_{DB}(\mathcal{G} | \theta)}{\sum_{\mathcal{E}} P_{DB}(\theta | \mathcal{E}, \alpha) P_{DB}(\mathcal{E} | \mathcal{G}, \alpha)} \quad (1) \end{aligned}$$

$$\begin{aligned} &\sum_{\mathcal{E}} P_{DB}(\theta | \mathcal{E}, \alpha) P_{DB}(\mathcal{E} | \mathcal{G}, \alpha) \\ &\approx \frac{\sum_{m=1}^K P_{DB}(\theta | \mathcal{E}_m, \alpha)}{K} \quad (2) \end{aligned}$$

(1) の分子の  $P_{\text{Dir}}(\theta | \alpha)$  (ディリクレ分布) と  $P_{DB}(\mathcal{G} | \theta)$  は容易に計算される。一方分母の期待値は (2) にあるように  $\{\mathcal{E}^{(k)}\}_{k=1}^m$  により近似計算される。両者を合わせて周辺尤度  $P_{DB}(\mathcal{G} | \alpha)$  の近似計算が可能になる。なお (2) の  $\theta$  は任意であるが、PRISMでは前述の  $\bar{\theta}_{VB}$  を使っている。

同様に新しい観測アトム  $G_{\text{new}}$  に対するベイズ Viterbi 説明  $\epsilon_{\text{new}}^*$  は以下のように近似計算される [18]。

$$\begin{aligned} \epsilon_{\text{new}}^* &= \operatorname{argmax}_{\epsilon_{\text{new}} \in \operatorname{expl}(G_{\text{new}})} P_{DB}(\epsilon_{\text{new}} | G_{\text{new}}, \mathcal{G}, \alpha) \\ &= \operatorname{argmax}_{\epsilon_{\text{new}} \in \operatorname{expl}(G_{\text{new}})} P_{DB}(\epsilon_{\text{new}} | \mathcal{G}, \alpha) \quad (3) \end{aligned}$$

$$\begin{aligned} &P_{DB}(\epsilon_{\text{new}} | \mathcal{G}, \alpha) \\ &= \sum_{\mathcal{E}} P_{DB}(\epsilon_{\text{new}} | \mathcal{E}, \alpha) P_{DB}(\mathcal{E} | \mathcal{G}, \alpha) \\ &\approx \frac{\sum_{m=1}^K P_{DB}(\epsilon_{\text{new}} | \mathcal{E}_m, \alpha)}{K} \quad (4) \end{aligned}$$

実際には (3) の右辺に於ける  $G_{\text{new}}$  の説明集合  $\operatorname{expl}(G_{\text{new}})$  が巨大なので、 $\operatorname{expl}(G_{\text{new}})$  の中から尤度が高そうな説明を少数 (5~10 個) 選んで候補とし、各候補に対して (4) の近似計算により周辺尤度推定し、最も高い周辺尤度を与える候補を  $\epsilon_{\text{new}}^*$  としている。

## 5 PRISMのMCMCによるベイズ推論例

これまで述べた MCMC の応用として簡単な構造探索問題を取り扱う。図 3 は NB (naive Bayes) を拡張した NBH (NB with a hidden class) の例である。図 3 の NB は UCI repository 中の voting data<sup>2</sup> をモデル化している。 $C$  はクラスで democrat または republican を取り、 $A_1, \dots, A_{16}$  は 16 回分の投票結果で、各  $A_i$  は yes, no, 不明の 3 値を取る。以後不明は missing data として扱う。

NB はあまり分類問題に対し良い性能を發揮しないが、我々は NB に図 3 にあるように隠れクラス変数  $HC$  を導入し、NB の改良を試みる。即ち  $P_{NBH}(\cdot)$  を NBH が定義する分布として、投票履歴  $A_1, \dots, A_{16}$  から投票者の所属する政党 (democrat, republican)  $C^*$  を

$$C^* = \operatorname{argmax}_C P_{NBH}(C | A_1, \dots, A_{16})$$

により分類する分類課題の正解率を上げることを試みる。その際問題となるのは隠れクラスの数である。少なすぎれば効果がないし、多すぎれば data sparseness が問題となる。そこで今回提案された PRISM の一般的 MCMC 機能を使い、隠れクラス数を 2~50 に変えつつ、NBH が与える voting data の周辺尤度を計測し、最大周辺尤度を与える隠れクラス数を採用することとする。

2~50 の各隠れクラス数に於いて MCMC を 2000 ステップ走らせ、後半 1000 ステップのサンプリング結果を使って周辺対数尤度を推定することを 20 回繰り返した。図 4 に各クラス数に於ける周辺対数尤度の平均値 (estimated log ML) と分散を示す。対比のため同時に PRISM の変分ベイズ機能を使って周辺対数尤度の近似である変分自由エネルギーも示してある。図 4 によれば、周辺対数尤度は隠れクラス数 12 の時最大であり、一方変分自由エネルギーは隠れクラス数 3 で最大になる。

2~20 の各隠れクラス数に於いて 10 fold cross-validation を 10 回繰り返して計測した NBH の正解率の平均を図 5 にプロットしてある。MCMC はベイズ Viterbi 解を、VB は変分ベイズによる近似事後分布を使った近似ベイズ Viterbi 解を、EM は EM 学習により得られたパラメータを使った通常の Viterbi 解を使っている。図 5 によれば、まず隠れクラス数が少ない場合いずれもベースラインとなる NB の正解率 (90.8%) を 5% 程上廻っており、voting data に対し隠れクラスの導入が NB の正解率を上げる効果があることをハッキリ示している。一方隠れクラス数が多くなると EM はほぼ単調に正解

<sup>2</sup>1984 年の米国下院に於ける 16 回の投票結果 ( $C, A_1, \dots, A_{16}$ ) 435 名分となる。

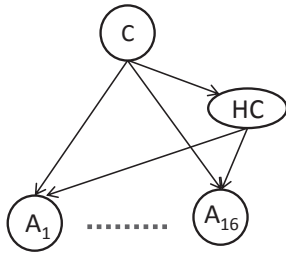


図 3: NBH model

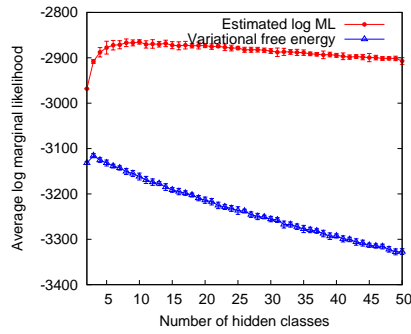


図 4: Average log marginal likelihood

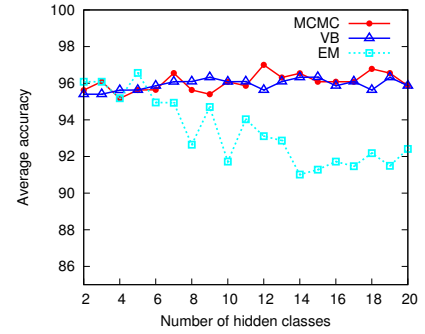


図 5: Average accuracy

率が悪化するのに対し、MCMC, VB はそのようなことはなく、data sparseness に対するベイズアプローチの効果が現れている。また隠れクラス数 12 (MCMC により推定した対数周辺尤度を最大にする隠れクラス数) の時 MCMC を使ったベイズ Viterbi 推論が最高の正解率 97% を達成している。

## 6 おわりに

生成的確率モデリングの為の論理に基づいたモデリング言語である PRISM 用の M-H サンプリングに基づいた MCMC 法を紹介した。この MCMC 法は全く一般的であり、Prolog を拡張した PRISM に依って記述される全ての確率モデルに適用できる。[18] には HMM や PCFG 以外にこれまで推定されたことのない確率左隅文法 (probabilistic left-corner grammar) の周辺尤度推定例が載っている。一方ここでは載せてないが profile-HMM など制約のきつい確率モデルで実際にモデルサイズを変えつつ周辺尤度を推定してみると、周辺尤度が非常に不連続に変化することが分かった。このような事実はこれまで汎用で手軽な周辺尤度推定の道具がなく、それゆえ測定を試みる人もなく見過ごされて来たのだらうと思われる。

現在 PRISM により実装されている MCMC のサンプリングルーチンを C に書き換えており、この作業が終わり次第、PRISM に組み込む予定である。PRISM に組み込まれた一般的 MCMC 法がベイズ推論の普及に役立つことを期待する。

## 参考文献

- [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 4911. Springer (2008) 1–27
- [3] Sato, T., Kameya, Y.: PRISM: a language for symbolic-statistical modeling. In: Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI'97). (1997) 1330–1335
- [4] Pfeffer, A.: IBAL: A probabilistic rational programming language. In: Proceedings of the 17th International Conference on Artificial Intelligence (IJCAI'01). (2001) 733–740
- [5] Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D., Kolobov, A.: BLOG: Probabilistic models with unknown objects. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05). (2005) 1352–1359
- [6] De Raedt, L., Kimmig, A., Toivonen, H.: ProbLog: A probabilistic Prolog and its application in link discovery. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07). (2007) 2468–2473
- [7] 亀谷由隆, 佐藤泰介, 周能法, 泉祐介, 岩崎達也: PRISM: 確率モデリングのための論理プログラミング処理系. コンピュータソフトウェア 24(4) (2007) 2–22
- [8] Park, S., Pfenning, F., Thrun, S.: A probabilistic language based on sampling functions. ACM Transactions on Programming Languages and Systems (TOPLAS) 31(1) (2008) 1–46

- [9] Goodman, N., Mansinghka, V., Roy, D., Bonawitz, K., Tenenbaum, J.: Church: a language for generative models. In: Proceedings of the Twenty fourth Conference on Uncertainty in Artificial Intelligence (UAI'08). (2008)
- [10] Kiselyov, O., Shan, C.: Embedded probabilistic programming. In: Proceedings of the IFIP working conference on domain-specific languages, LNCS 5658. (2009) 360–384
- [11] Pfeffer, A.: Figaro: An object-oriented probabilistic programming language. Technical report, Charles River Analytics (2009)
- [12] McCallum, A., Schultz, K., Singh, S.: Factorie: Probabilistic programming via imperatively defined factor graphs. In: Advances in Neural Information Processing Systems 22. (2009) 1249–1257
- [13] Wingate, D., Stuhmueller, A., Goodman, N.D.: Lightweight implementations of probabilistic programming languages via transformational compilation. Proceedings of the 14th international conference on Artificial Intelligence and Statistics (2011)
- [14] Riguzzi, F., Terrance Swift, T.: The pita system: Tabling and answer subsumption for reasoning under uncertainty. *Theory and Practice of Logic Programming (TPLP)* **11**(4-5) (2011) 433–449
- [15] Sato, T.: A general MCMC method for Bayesian inference in logic-based probabilistic modeling. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11). (2011) 1472–1477
- [16] Sato, T.: Inside-Outside probability computation for belief propagation. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07). (2007) 2605–2610
- [17] Dechter, R.: Bucket elimination: A unifying framework for probabilistic inference. In: Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI'96). (1996) 211–219
- [18] Sato, T., Kameya, Y., Kurihara, K.: Variational Bayes via Propositionalized Probability Computation in PRISM. *Annals of Mathematics and Artificial Intelligence* **54** (2009) 135–158
- [19] Johnson, M., Griffiths, T., Goldwater, S.: Bayesian inference for PCFGs via Markov chain Monte Carlo. In: Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL'07). (2007) 139–146