NAIST-IS-DD1061201

Doctoral Dissertation

Adaptive Markov chain Monte Carlo for auxiliary variable method and its applications

Takamitsu Araki

September 24, 2013

Department of Information Systems
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Takamitsu Araki

Thesis Committee:

Professor Kazushi Ikeda	(Supervisor)
Professor Yuji Matsumoto	(Co-supervisor)
Professor Shoji Kasahara	(Co-supervisor)

Adaptive Markov chain Monte Carlo for auxiliary variable method and its applications[*]

Takamitsu Araki

Abstract

Markov chain Monte Carlo (MCMC) methods generate samples from a probability distribution, target distribution, by simulating Markov chains, and are efficient for sampling from a high dimensional and complex distribution, which is necessary for various fields such as statistical physics, statistics and machine learning. However, standard MCMC methods cannot generate samples from a multimodal distribution and a posterior distribution in Bayesian variable selection. To generate samples from such distributions, Parallel Tempering (PT) algorithm and Gibbs variable selection (GVS) were proposed. These algorithms use auxiliary distributions and the MCMC methods that use the auxiliary distributions are referred to as auxiliary variable methods (AVMs).

The PT algorithm uses the auxiliary distributions that are constructed by inducing inverse temperatures to the target distribution and are the target distribution whose multimodality is tempered. The PT algorithm generates samples from the auxiliary distributions and the target distribution in parallel, and exchanges the values of the two samples with an acceptance probability. The exchange process releases the samples for the target distribution from the local regions, and hence the samples are correctly distributed according to the target distribution. The performance of the PT algorithm strongly depends on the inverse temperatures and the parameters of the proposal distributions. Conventionally the parameters are tuned by trial-and-error in many pilot runs.

The GVS generates the samples from the discontinuous and multimodal posterior distribution in Bayesian variable selection. The GVS uses pseudo-priors

[*] Doctoral Dissertation, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1061201, September 24, 2013.

to approximate the posterior distribution to a unimodal one. This increases its sampling efficiency. The efficiency of the GVS strongly depends on parameters of the proposal distribution and the pseudo-priors. The conventional GVS sets the parameters by using the samples obtained by a pilot run for a full model, but the parameters are improper.

Generally the performance of the AVMs also depends on the parameters of the proposal distribution and the auxiliary distributions. Therefore a choice of the proper parameters in the AVMs is a crucial problem.

In this dissertation, we propose an adaptive PT algorithm and an adaptive GVS that adapt their parameters while they run. We confirm that these proposed algorithms can obtain the proper parameters through numerical experiments.

Furthermore, we generalize the proposed algorithms to an adaptive MCMC for AVMs that adapts the parameters of the AVMs on the fly. We prove convergence theorems of the algorithm, and show that the adaptive MCMC for AVMs converges under mild sufficient conditions. We also prove the convergence of the adaptive PT algorithm and the adaptive GVS by applying the convergence theorem of the adaptive MCMC for AVMs.

Keywords:

Markov chain Monte Carlo methods, Auxiliary variable methods, Parallel Tempering, Gibbs variable selection, Adaptive Markov chain Monte Carlo methods, Convergence theorem

補助変数法に対する適応的マルコフ連鎖モンテカルロ法とその応用*

荒木 貴光

内容梗概

マルコフ連鎖モンテカルロ法（MCMC 法）はマルコフ連鎖を用いた確率分布からのサンプリング法であり，統計物理学，統計学，機械学習など様々な分野において必要とされる複雑で高次元な確率分布からのサンプリングに有効である．しかし，多峰性のある確率分布やベイジアン変数選択における特殊な事後分布に対しては標準的な MCMC 法では適切にサンプリングできず，それらの分布から適切にサンプリングするためのパラレルテンパリング，ギブス変数選択法など補助分布を用いた MCMC 法が提案された．このような補助分布を用いた MCMC 法のクラスを補助変数法という．

パラレルテンパリングは，多峰性のある確率分布に逆温度パラメータを導入して多峰性を徐々に緩和した分布列を構成し，最後の分布は標準的な MCMC 法でも十分にサンプリングできるほど滑らかな分布とする．それぞれの分布からサンプリングとサンプルの交換を行うことでサンプルが局所領域に留まるのを防ぎ，分布全体からサンプリングする．パラレルテンパリングの性能は逆温度パラメータ，目的分布と補助分布列に対する Metropolis 法の提案分布の分散に強く依存し，従来は多数の予備実験と試行錯誤によってそれらパラメータの調整が行われてきた．

ギブス変数選択法は，ベイジアン変数選択における連続変数と離散変数が混在する不連続で多峰性をもつ事後分布から効率的にサンプリングする方法であり，係数パラメータの周辺事後分布を近似した擬似事前分布を導入し事後分布を単峰に近づけることでサンプリングを効率化する．ギブス変数選択法の性能も同様に擬似事前分布のパラメータ，提案分布の分散共分散行列に強く依存する．従来は

*奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 博士論文, NAIST-IS-DD1061201, 2013 年 9 月 24 日.

一度予備実験を行いそこで得られたサンプルを用いてパラメータを設定するという方法がとられてきたが，それでは適切なパラメータを得ることができない．

一般的に補助変数法の性能も提案分布と補助分布に含まれるパラメータに強く依存するため，適切なパラメータの設定は重要な問題である．

本論文では，これらパラメータ設定の問題に対して，パラレルテンパリングとギブス変数選択法を拡張して，サンプリングしながらパラメータの値を適応的に更新するアルゴリズムである適応的パラレルテンパリングと適応的ギブス変数選択法をそれぞれ提案する．数値実験を行い，これら提案アルゴリズムが適切なパラメータを得ることを確認する．

さらに，上述した提案アルゴリズムを一般化して補助変数法でサンプリングしながらそのパラメータを適応的に更新する適応的な補助変数法を提案する．そのアルゴリズムの収束定理を証明し，緩い条件下で適応的な補助変数法が収束することを示す．この定理を用いて適応的パラレルテンパリングと適応的ギブス変数選択法の収束を示す．

キーワード

マルコフ連鎖モンテカルロ法, 補助変数法, パラレルテンパリング, ギブス変数選択法, 適応的マルコフ連鎖モンテカルロ法, 収束定理

Contents

Chapter 1

Introduction

Markov chain Monte Carlo (MCMC) methods are important algorithms in various fields, e.g. statistics, physics and machine learning (Liu, 2001; Robert and Casella, 2004). The MCMC methods generate samples from a target distribution by using a simple proposal distribution or its conditional distributions. For example, a Metropolis algorithm, the simplest MCMC method, generates a sample candidate from the proposal distribution, and accepts it with an acceptance probability.

However, there are complex target distributions from which standard MCMC methods can not correctly generate the samples. To correctly generate the samples from the complex target distributions, auxiliary distributions are induced to the target distributions and the samples are generated from the joint distributions of the target distribution and the auxiliary distributions. The MCMC methods that use the auxiliary distributions are referred to as auxiliary variable methods, which include Parallel Tempering algorithm and Gibbs variable selection.

1. Motivation

The standard MCMC methods can not correctly produce samples from a multimodal distribution, because the produced samples can be trapped in a local mode for an extremely long period. To overcome this localization problem, Parallel Tempering (PT) algorithm was proposed (Geyer, 1991). The PT algorithm induces auxiliary distributions that are constructed by adding inverse temperatures to the target distribution. Lowering the inverse temperatures flattens a landscape of the auxiliary distribution and thus eases an exploration of the samples in its sample space. The PT algorithm generates samples from the auxiliary distributions and the target distribution, by a Metropolis algorithm in this study, and exchanges a position of the two samples with an acceptance probability. The sample generated from the auxiliary distribution with low inverse temperature can be transmitted to the target distribution through the exchange operations. This releases the sample for the target distribution from a local mode into another one, and enables the sample to converge the target distribution.

The performance of the PT highly depends on the inverse temperatures and the variances of the proposal distributions. Conventionally the parameters have been determined by trial-and-error in many pilot runs.

In Bayesian variable selection, a posterior of a statistical model is a discontinuous and multimodal distribution with continuous and discrete variables, from which the standard MCMC methods can not efficiently generate samples. To efficiently generate the samples from the distribution, Gibbs variable selection (Dellaportas et al. 2002; GVS) was proposed. The GVS induces pseudo-priors to the statistical model in order to facilitate the sampling from the posterior. Due to adding the pseudo-priors, the multimodal posterior approaches to a unimodal one. The GVS generates samples by the Gibbs sampler and the Metropolis-Hastings (MH) algorithm by turns, where we use the Metropolis algorithm as the MH algorithm in this dissertation.

The efficiency of the GVS strongly depends on the parameters of the proposal distribution and the pseudo-priors, and the conventional GVS determines the parameters based on a pilot run for a full model, which contains all covariates. The determined parameters are often improper because the posterior of the full model is different from that of the model in Bayesian variable selection.

The MCMC methods that use auxiliary distributions like the above MCMC methods are called auxiliary variable methods (AVMs) in this dissertation. The AVMs include other various effective MCMC methods such as cluster Monte Carlo methods. The AVMs also depend on the parameters of the proposal distribution and the auxiliary distributions. Therefore a choice of the proper parameters in the AVMs is a crucial problem.

For the standard MCMC methods, Gilks et al. (1998) and Haario et al. (2001)

proposed adaptive MCMC algorithms that tuned the parameters of the proposal distribution by using generated samples during runs. Haario et al. (2001) also proved the convergence theorem of the algorithms.

2. Contribution

We propose an adaptive PT algorithm that adapts the inverse temperatures and the parameters of the proposal distributions, and an adaptive GVS that adapts the parameters of the pseudo-priors and the proposal distribution. By numerical experiments, we confirm that the proposed algorithms can obtain the appropriate parameters.

We generalize the algorithms to an adaptive MCMC for AVMs that adapts the parameter of the proposal distributions and auxiliary distributions while AVMs run, and prove its convergence theorems. We reveal that the adaptive MCMC for AVMs converges under mild sufficient conditions. We also prove the convergence of the adaptive PT algorithm and the adaptive GVS by applying the convergence theorem of the adaptive MCMC for AVMs.

The adaptive PT algorithm enables us to implement the efficient PT algorithm without trial-and-error in many pilot runs. Thus, we can simulate a system that has a complex free energy structure, e.g. a spin glass model and protein, and implement Bayesian estimation of a nonlinear model without any preliminary experiments. The adaptive GVS allows us to more efficiently perform Bayesian variable selection.

3. Organization of Dissertation

The rest of this dissertation is organized as follows.

In Chapter 2, we denote the review of the MCMC methods and the adaptive MCMC algorithms.

In Chapter 3, we propose the adaptive PT algorithm, and validate the performance of the algorithm via the numerical experiments.

In Chapter 4, we propose the adaptive GVS, and evaluate the efficiency of the algorithm by the numerical experiments.

Chapter 5 generalizes the algorithms proposed in Chapter 3 and Chapter 4 to the adaptive MCMC for AVMs, and proves its convergence theorems.

Finally we give discussion and future works in Chapter 6.

4. Notation

In this dissertation, we denote a sample or a parameter, A, at nth iteration by $A^{(n)}$. A probability distribution or a probability density function is denoted by f, and a target distribution or a target density is denoted by π. " $\sim$ " denotes sampling from a distribution.

Chapter 2

Markov chain Monte Carlo Method

When producing independent identically distributed samples from a target distribution is infeasible, Markov chain Monte Carlo (MCMC) methods are used to generate samples from the target distribution.

The MCMC methods simulate Markov chains whose distribution converges to a target distribution. After enough iteration, the generated Markov chains are considered to be distributed according to the target distribution and the samples from the target distribution.

Sampling algorithms that use independent random variables need a perspective of the target distribution, and can not generate samples from high dimensional and complex target distribution whose perspective is unknown.

The simulation of the Markov chains that converge to the target distribution uses only a region around the current position of the Markov chain, and does not use the perspective of the target distribution. Therefore the MCMC methods can efficiently generate samples from the target distribution whose perspective is unknown.

In what follows, we shortly review the two typical MCMC methods, Gibbs sampler and Metropolis-Hastings algorithm, and adaptive MCMC methods, which were proposed to overcome the parameter setting problems of the standard MCMC methods.

1. Gibbs sampler

Let $\pi(x)$ be the target distribution, where $x = (x_1, \ldots, x_p)$, and x_j denote a set of variables. Gibbs sampler (Geman and Geman, 1984) iteratively generates samples from the conditional distribution $\pi(x_j|x_{-j})$ for $j = 1, ..., p$, where $x_{-j} = (x_1, ..., x_{j-1}, x_{j+1}, ..., x_p)$.

A pseudo-code of the Gibbs sampler is as follows.

```
**Initialize** $x^{(0)} = (x_1^{(0)}, ...x_p^{(0)})$.
**for** $n = 0, 1, ...$ **do**
  **for** $j = 1, ..., p$ **do**
    $x_j^{(n+1)} \sim \pi(x_j|x_{-j}^{(n+1)})$, where $x_{-j}^{(n+1)} = (x_1^{(n+1)}, ..., x_{j-1}^{(n+1)}, x_{j+1}^{(n)}, ..., x_p^{(n)})$.
  **end for**
**end for**
```

For enough large number N, $\{x^{(n)}; n \geq N\}$ are considered as the samples from the target distribution $\pi(x)$.

The Gibbs sampler is efficient, only if it is easy to sample from the conditional distribution of the target distribution. Therefore, if sampling from the conditional distribution is difficult, the Metropolis-Hastings algorithm, described in the next section, is used.

2. Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) algorithm (Hastings, 1970) generates a sample candidate from a proposal distribution and then accepts the candidate as the sample with the acceptance probability and rejects the candidate with the complement of the probability. The sample candidate from the simple proposal distribution is considered to be adjusted to the target distribution by the acceptance probability.

The pseudo-code of the MH algorithm is as follows.

```
**Initialize** $x^{(0)}$.
**for** $n = 0, 1, ...$ **do**
  (1) $x' \sim q_\theta(x|x^{(n)})$, where $q_\theta(|)$ denotes both of the proposal distribution and
  its density, and $\theta$ denote proposal parameters.
```

```
    (2) Calculate the acceptance probability $\alpha(x^{(n)}, x') = \min\left\{\frac{\pi(x')q_\theta(x^{(n)}|x')}{\pi(x_n)q_\theta(x'|x^{(n)})}, 1\right\}$
    (3) $u \sim U[0,1]$ (where $U[0,1]$ is a uniform distribution of the interval (0,1)).
    $x^{(n+1)} = \begin{cases} x', & \text{if } u \le \alpha(x^{(n)}, x') \\ x^{(n)} & \text{if } u > \alpha(x^{(n)}, x') \end{cases}$
end for
```

For the enough large number N, $\{x^{(n)}; n \ge N\}$ are considered as the samples from the target distribution $\pi(x)$.

The most famous MH algorithm is a Metropolis algorithm, whose proposal distribution is symmetric, i.e., $q_\theta(y|x) = q_\theta(x|y)$. The Metropolis algorithm is used in Chapters 3 and 4 in this dissertation.

The MH algorithm can be applied to any probability distribution in contrast to the Gibbs sampler. However, the performance of the MH algorithm strongly depends on the parameters of the proposal distribution, therefore a proper choice of the parameter is a crucial factor for the performance of the algorithm.

The adaptive MCMC algorithm, described in the next section, tunes the parameters of the proposal distribution to the appropriate parameters while it runs.

3. Adaptive MCMC algorithm

The efficiency of the MH algorithm is determined by the parameters of the proposal distribution. For example, we consider the sampling from $\pi(x),\ x \in \mathbb{R}$ using the Metropolis algorithm with proposal variance σ^2. If the proposal variance σ^2 is too small compared to that of the target distribution, the generated Markov chains always move slowly. On the other hand, if the proposal variance σ^2 is too large, the Markov chain stays a same position for a long period.

In many cases, the target distribution $\pi(x)$ is high dimension and the proposal distribution has a covariance matrix, that is, the MH algorithm has the many parameters which affect its performance. The careful tuning of the parameters needs a high computational and artificial cost.

Gilks et al. (1998) and Haario et al. (2001) proposed adaptive MCMC algorithms that tuned the parameters of the proposal distribution by using generated samples so far during runs. For example, the adaptive MCMC algorithm for the Metropolis algorithm with a normal proposal distribution updates its proposal covariance matrix Σ and mean parameters μ, which are needed to update the matrix, after generating the sample $x^{(n+1)}$ in $(n+1)$-th step as follows.

$$\begin{aligned}
&\mu^{(n+1)} \leftarrow \mu^{(n)} + \lambda_n \left(x^{(n+1)} - \mu^{(n)}\right), \\
&\Sigma_{(n+1)} \leftarrow \Sigma^{(n)} + \lambda_n \left(\left(x^{(n+1)} - \mu^{(n)}\right)\left(x^{(n+1)} - \mu^{(n)}\right)^T - \Sigma^{(n)}\right),
\end{aligned}$$

where a learning coefficient λ_n is a decreasing function that converges to zero.

Due to the parameter updates using the past samples, the adaptive MCMC algorithms are non-Markovian, but Haario et al. (2001) established the convergence theorem of the algorithms. Andrieu and Moulines (2006) and Roberts and Rosenthal (2007) generalized the convergence theorems, and relaxed their sufficient conditions.

The above adaptive MCMC algorithms adapt only the parameters of the proposal distribution. In this dissertation, we extend them to adaptive algorithms that adapt not only the proposal parameters but also the parameters of the auxiliary distributions while the AVMs run.

Chapter 3

Adaptive Parallel Tempering algorithm

1. Introduction

MCMC methods can generate samples that follow a target distribution by using a simple proposal distribution. However, in sampling from a complex distribution such as multi-modal one, the standard MCMC methods produce samples that theoretically converge to the target distribution but practically do not. The produced samples can be trapped in a local mode for an extremely long period.

To cope with this localization problem, the parallel tempering (PT) a.k.a. exchange Monte Carlo method was proposed (Geyer 1991; Hukushima and Nemoto 1996). The PT algorithm introduces auxiliary distributions with a parameter called the inverse temperature, generates multiple MCMC samples from target and auxiliary distributions in parallel, and exchanges the positions of two samples. An auxiliary distribution is tempered when the temperature is high and one with a low temperature is similar to the target distribution. This “tempering” implementation and the exchange process help samples escape from a local mode.

However, the PT algorithm strongly depends on the inverse temperatures and the parameters of the proposal distributions, and the turning of the parameters needs many pilot runs and trial-and-error.

In this chapter, we propose an adaptive PT algorithm that tunes the temperatures and the proposal parameters, which include the number of the temperatures, while it runs, and show the effectiveness of the algorithm via numerical experiments.

2. Parallel Tempering Algorithm

The PT algorithm is a typical algorithm that uses auxiliary distributions, $\pi_{t_l}(dx_l)$, $l = 2, ..., L$, where $1 = t_1 > t_2 > \cdots > t_L > 0$. The density of the lth auxiliary distribution is parameterized by the inverse temperature t_l as $\pi_{t_l}(x) \propto \pi(x)^{t_l}$ or $\pi_{t_l}(x) \propto \pi(x)^{t_l} p(x)^{1-t_l}$, where $\pi(x)$ is the density of the target distribution and $p(x)$ is the density of a simple distribution for which a standard MCMC method mixes fast. In other words, the inverse temperature t_l tempers the multimodality of the target distribution $\pi(dx)$ so that the auxiliary densities, $\pi_{t_l}(x_l)$, gradually connect the target density $\pi(x)$ to a simple density $p(x)$ or the uniform distribution.

The PT algorithm executes either of the parallel step and the exchange step at time n, with probability α_r and $1 - \alpha_r$, respectively. The parallel step generates the L samples, $x_l^{(n+1)}$, $l = 1, \ldots, L$, according to $\pi_{t_l}(dx_l)$ for each by using a standard MCMC method. Note that we employed the Metropolis algorithm with an independent proposal distribution that has the variances γ_l in this chapter. The exchange step randomly chooses a sample $x_l^{(n)}$ from the $L - 1$ samples, $x_l^{(n)}$, $l = 1, \ldots, L - 1$, and exchange $x_l^{(n)}$ for $x_{l+1}^{(n)}$ with probability

$$\min\left(1, \frac{\pi_{t_l}(x_{l+1}^{(n)})\pi_{t_{l+1}}(x_l^{(n)})}{\pi_{t_l}(x_l^{(n)})\pi_{t_{l+1}}(x_{l+1}^{(n)})}\right). \tag{3.1}$$

The performance of the PT algorithm strongly depends on the inverse temperatures, more specifically, their intervals and their number. The interval of two adjacent inverse temperatures determines both the similarity of the two distributions and the acceptance probability of an exchange as seen in Eq. (3.1). The acceptance ratio for the exchanges, which is referred to as the exchange ratio in this dissertation, should not take an extreme value. For example, Liu (2001) said a preferable value is a half at any interval. To avoid extreme values and lead to homogeneous exchange ratios, Hukushima (1999) updated temperatures using a recursive formula through preliminary runs and Goswami and Liu (2007) tuned the intervals by iteratively estimating the expected exchange probability through preliminary runs.

Jasra (2007) treated the intervals as a sequence and experimentally compared three inverse-temperature sequences, equal space, logarithmic decay and power decay. The results showed the last was the best. Nagata and Watanabe (2008) proved that when the sequence of inverse temperatures is a geometric progression, the exchange ratios are homogeneous in the low temperature limit. However, the above methods only discussed the intervals and did not take into account the proposal distributions, on which the mixing of samples and the estimation of exchange ratio also depend. In our setting, the Metropolis algorithm has a parameter to be determined, that is, the proposal variances γ_l. It is necessary to re-set the proposal variance when the inverse temperatures are changed a lot, because the appropriate proposal variances obviously depend on the shape of auxiliary distributions.

The more auxiliary distributions the PT algorithm has, the faster the samples mix because flatter auxiliary distributions are available but the more computational complexity is required. To solve the trade-off and determine an appropriate number of distributions, Goswami and Liu (2007) proposed to select the maximum temperature using statistical tests. The tests should be done in an off-line manner, that is, they need preliminary experiments in advance.

3. Adaptive PT Algorithm

We propose an adaptive PT algorithm that adapts the inverse temperatures, the variances of proposal distribution, and the minimum inverse temperature while the algorithm is running. The three adaptation algorithms are described below.

The exchange ratio should take a moderate value. To converge the exchange ratio for x_{l-1} and x_l to a specific value, $\alpha \in (0,1)$, typically a half, the log inverse temperature, $\zeta_l = \log(t_l)$, is updated as

$$\zeta_l^{(n+1)} \leftarrow \zeta_l^{(n)} - a_n^l (ER_{l-1,l}^{(n)} - \alpha), \tag{3.2}$$

where $ER_{l-1,l}^{(n)}$ is a variable that takes one if the exchange occurs between the samples, $x_{l-1}^{(n)}$ and $x_l^{(n)}$, at time n, and zero otherwise. The learning coefficient, a_n^l, is a decreasing random variable with n that satisfies $\lim_{n\to\infty} a_n^l = 0$ *almost sure.*

The proposal distribution of the Metropolis algorithm for a target and auxiliary distribution should have an appropriate variance, which is an average of the variances of all modes of the corresponding target or auxiliary distribution. To converge the proposal variances $\gamma_l = (\gamma_{l1}, ..., \gamma_{lp}) \in \mathbb{R}^p$ of the Metropolis algorithm for the distribution $\pi_{t_l}(dx_l)$ on $\mathbb{R}^p$ to such average values, γ_l and the auxiliary adaptation parameters $\mu_l = (\mu_{l1}, ..., \mu_{lp}) \in \mathbb{R}^p$, which are used only for the adaptation of γ_l, are updated as

$$\begin{aligned} \mu_{lj}^{(n+1)} &\leftarrow \mu_{lj}^{(n)} + b_n(x_{lj}^{(n+1)} - \mu_{lj}^{(n)}), \\ \gamma_{lj}^{(n+1)} &\leftarrow \gamma_{lj}^{(n)} + b_n\left((x_{lj}^{(n+1)} - \mu_{lj}^{(n+1)})^2 - \gamma_{lj}^{(n)}\right), \end{aligned} \tag{3.3}$$

where $x_{lj}^{(n+1)}$ is the jth element of $x_l^{(n+1)} \in \mathbb{R}^p$. The learning coefficient, b_n, is a decreasing function of n that satisfies $\lim_{n\to\infty} b_n = 0$. When $x_l^{(n)}$ is updated to $x_l^{(n+1)}$ by exchanging to $x_{l-1}^{(n)}$ or $x_{l+1}^{(n)}$,

$$\mu_l^{(n+1)} \leftarrow x_l^{(n+1)}. \tag{3.4}$$

Because μ_l is tuned to the mean of each mode by Eq. (3.3) and (3.4), γ_l can learn the variance of each mode by Eq. (3.3).

The auxiliary distribution with the minimum inverse temperature should be so flat that Metropolis samples can frequently move from one mode to another while the total number of auxiliary distributions should be as small as possible. To determine an appropriate value for the minimum inverse temperature, the auxiliary distributions $\pi_{t_l}(dx_l)$ with $l > l^*$ are removed where l^* is the smallest number that satisfies

$$\prod_{j=1}^{p} \gamma_{lj}^{(n)} \geq \prod_{j=1}^{p} V^{(n)}(x_{lj}), \tag{3.5}$$

where $V^{(n)}(x_{lj})$ is the sample variance of x_{lj} at time n. This check is done at time $n = m, 2m, \ldots$, where m is a large number (e.g. 10^4). To improve the reliability, when inequality (3.5) holds a few times d (e.g. 3) in succession, the auxiliary distribution is determined to be enough flat.

Inequality (3.5) shows the relationship between the sample variance and the proposal variance. Due to Eq. (3.3) and (3.4), the latter converges to the average of variances of local regions and hence it is smaller than the sample variance if

Metropolis samples are localized in each mode. Otherwise, the auxiliary distribution is flat enough.

A pseudo code of the adaptive PT algorithm is given in the following.

The adaptive PT algorithm converges. The proof will be given in chapter 5 as a special case of adaptive MCMC algorithms for general auxiliary variable methods.

4. Experiments

To confirm the effectiveness of our algorithm, the following three numerical experiments were carried out:

1. A mixture of four normal distributions.

2. The posterior of a mixture model of six normal distributions.

3. The predictive distribution for Galaxy Data.

In each of the experiments, the burn-in period was a half of the total number of iterations and sample sets, which were used in an estimation and a scatter plot, were chosen from every 50 samples in post burn-in. The proposal distribution of the Metropolis algorithm was an independent normal distribution. Other parameters were $\alpha = 0.5$, $\alpha_r = 0.5$, $a_n^l = 1/(1 + n/(20 + 10l)) \log(\exp(-\zeta_l^{(n)}) + 1)$, $b_n = 1/(5 + 0.1n)$, $m = 10^4$ and $d = 3$. a_n^l and b_n were set so that they could converge to zero slowly and a_n^l could converge more slowly as l increased. $L = 25$ and the intervals of inverse temperatures were equal, that is $t^{(0)} = (1, 24/25, 23/25, \ldots, 1/25)$, at the initial condition. Note that these values are invariant for the each above distribution, i.e., a tuning of these values was not necessary in these experiments.

4.1 A mixture of four normal distributions

To see and visualize the properties of our adaptive PT algorithm, we chose a mixture of four normal distributions in two dimensional space as the target dis-

Algorithm 1 Adaptive PT algorithm

```
**Initialize** $x_l^{(0)}$, $\zeta_l^{(0)}$, $\gamma_l^{(0)}$, $\mu_l^{(0)}$ and $c_l = 0$ ,$l = 1, ..., L$. ($\zeta_1^{(0)}$=0, constant).
**for** $n = 0$ to $N - 1$ **do**
  $u \sim U[0, 1]$ (where $U[0, 1]$ is the uniform distribution of the interval (0,1)).
  **if** $u \leq \alpha_r$ **then**
    **for** $l = 1$ to $L$ **do**
      (*parallel sampling step*)
      Generate $x_l^{(n+1)}$ via Metropolis algorithm for $\pi_{t_l^{(n)}}(dx_l)$, which has the proposal variances $\gamma_l^{(n)}$.
      (*proposal parameter learning step*)
      Update ($\gamma_l^{(n)}$, $\mu_l^{(n)}$) to ($\gamma_l^{(n+1)}$, $\mu_l^{(n+1)}$) by the Eq. (3.3).
    **end for**
  **else**
    (*exchange step*)
    Randomly choose a neighboring pair, $x_l^{(n)}$ and $x_{l+1}^{(n)}$, and exchange them with the probability Eq. (3.1).
    (*inverse temperature learning step*)
    Update $\zeta_{l+1}^{(n)}$ to $\zeta_{l+1}^{(n+1)}$ by Eq. (3.2).
    **if** the exchange is accepted, **then**
      $\mu_k^{(n+1)} \leftarrow x_k^{(n+1)}$, for $k = l, l + 1$.
    **end if**
  **end if**
  (*minimum inverse temperature decision step*)
  **if** $(n \bmod m) = 0$ **then**
    **for** $l = 1$ to $L$ **do**
      If Eq. (3.5) hold, then $c_l \leftarrow c_l + 1$.
    **end for**
    $\Delta \leftarrow \{l | c_l \geq d, l = 1, \ldots, L\}$.
    if $\Delta \neq \emptyset$, then $L \leftarrow \min \Delta$.
  **end if**
**end for**
```

tribution (Fig. 3.1(a)):

$$g(x) = \sum_{i=1}^{4} \frac{1}{8\pi \det(\Sigma_i)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right),$$

where the parameters of the normal distributions are

$$\begin{aligned} \mu_1 &= (0, 44), & \mu_2 &= (44, 0), \\ \mu_3 &= (0, -44), & \mu_4 &= (-44, 0), \\ \Sigma_1 &= \mathrm{diag}(1, 7^2), & \Sigma_2 &= \mathrm{diag}(7^2, 1), \\ \Sigma_3 &= \mathrm{diag}(1, 7^2), & \Sigma_4 &= \mathrm{diag}(7^2, 1). \end{aligned}$$

Note that these normal distributions have quite different variances, 1 and 7^2, where the proposal variance learning is difficult.

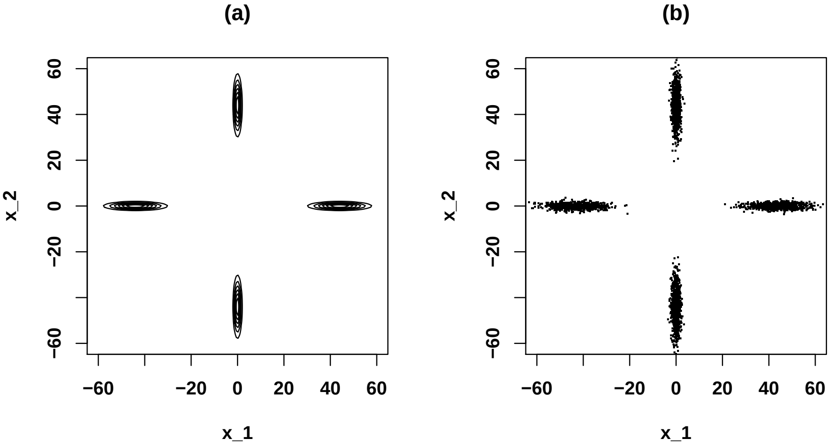

Figure 3.1. A mixture of four normal distributions. a) The target distribution. b) Samples by the adaptive PT algorithm.

The adaptive PT algorithm ran for 3×10^5 iterations, where the auxiliary distributions are $\pi_{t_l}(x) \propto g(x)^{t_l}$ and the initial proposal variances are $\gamma_{lj}^{(0)} = 3 \times 10^2$.

As a result, our algorithm mixed well and obtained samples from all possible modes (Fig. 3.1(b)). In fact, the number of inverse temperatures was reduced to five after 3×10^4 iterations but the auxiliary distribution $\pi_{\hat{t}_5}(dx)$ is flat enough (Fig. 3.2), where $\hat{t}_5$ is the inverse temperature t_5 tuned by the adaptive PT algorithm.

The larger the variances of the auxiliary distribution become, the larger the proposal variances should become. In fact, the sums of the adapted proposal variances were $(\hat{\gamma}_{1,1} + \hat{\gamma}_{1,2}, \dots, \hat{\gamma}_{5,1} + \hat{\gamma}_{5,2}) = (32.26, 41.86, 245.8, 1124, 8704)$.

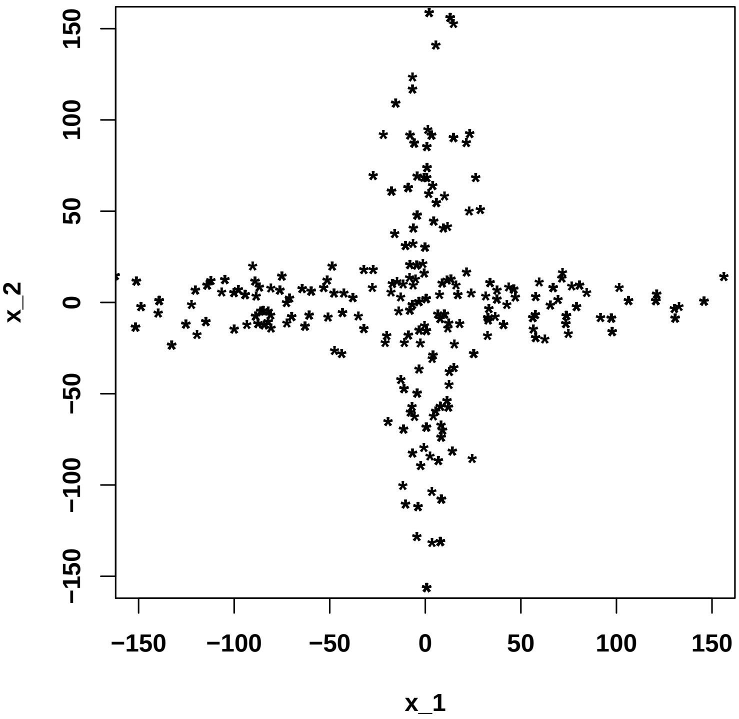

Figure 3.2. Samples by Metropolis algorithm with the proposal variance $\hat{\gamma}_5$ from the auxiliary distribution $\pi_{\hat{t}_5}(dx)$. They cover all the modes in Fig. 3.1.

The estimated exchange ratios converged to $(0.501, 0.507, 0.499, 0.498)$, all of which are almost $\alpha = 0.5$. Then, the adapted inverse temperatures were $(\hat{t}_2, \dots, \hat{t}_5) = (0.328, 0.108, 0.0307, 0.00937)$.

$t_2^{(n)}$ and $\gamma_2^{(n)}$ converge quickly from even the extreme starting points (Fig. 3.3), and the others also converge as fast as them.

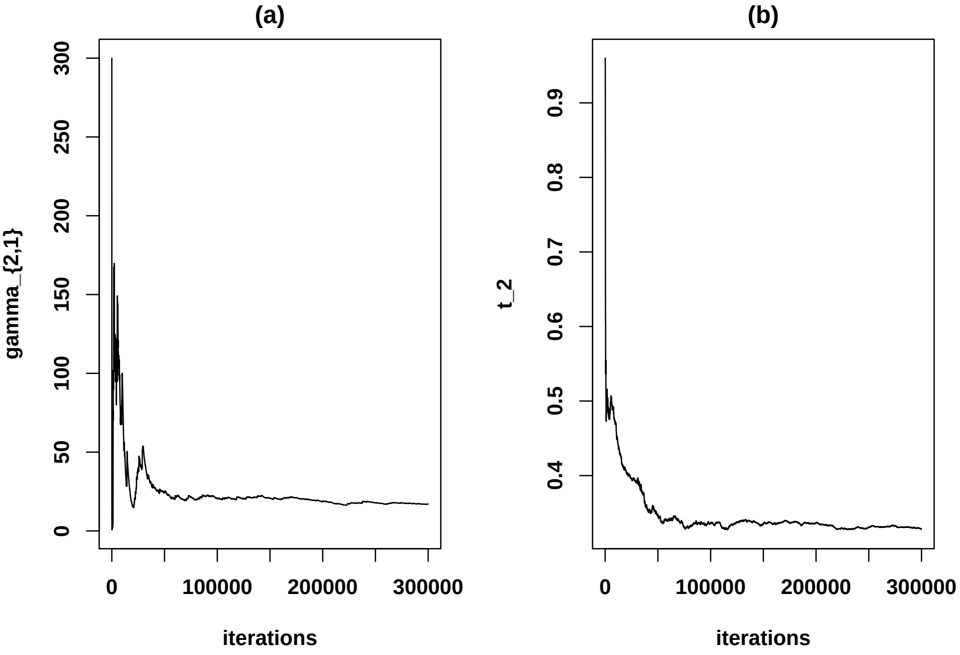

Figure 3.3. trace plot (a):proposal variance $\gamma_{2,1}^{(n)}$, (b):inverse temperature $t_2^{(n)}$.

4.2 The posterior of a mixture model of six normal distributions

In the mixture model, we estimated the average of component specific means, μ_m, by the posterior mean of μ_m as is seen in Jasra et al. (2007). The mixture model had normal distributions, that is,

$$f(y|\mu, w, \sigma^2) = \sum_{m=1}^{M} \frac{w_m}{\sqrt{2\pi}\sigma_m} \exp\left(-\frac{1}{2\sigma_m^2}(y-\mu_m)^2\right), \tag{3.6}$$

where $w_M = 1 - \sum_{m=1}^{M-1} w_m$. The priors are a normal-inverse Gamma-Dirichlet prior as follows.

$$\begin{aligned}\mu_m &\sim N(\xi, \kappa^2), \quad m = 1, \ldots, M,\\ \sigma_m^2 &\sim IG(\alpha_g, \beta_g), \quad m = 1, \ldots, M,\\ w_m &\sim \mathcal{D}(\varrho), \quad m = 1, \ldots, M-1,\end{aligned}$$

where $\mathcal{D}(\varrho)$ is the symmetric Dirichlet distribution with parameter ϱ. In the following, the hyper-parameters were $\alpha_g = 12$, $\beta_g = 10$ and $\varrho = 1$. The parameters ξ and κ^2 were determined by the median and four times the variance of the given data, respectively.

The data of size 150, $y_{1:150}$, were independently and identically distributed according to a mixture model of the form (3.6) with parameters, $M = 6$, $w_1 = \cdots = w_6 = 1/6$, $(\mu_1, \ldots, \mu_6) = (-8, -3, 1, 4, 8, 13)$, $\sigma_1^2 = \sigma_6^2 = 1.5^2$ and $\sigma_2^2 = \cdots = \sigma_5^2 = 0.5^2$. In this case, the posterior $\pi(\mu, w, \sigma^2|y_{1:150})$ was a 17 dimensional distribution and had $6! = 720$ symmetric modes due to the invariance against permutation of the labels of the parameters.

The auxiliary distributions were set to

$$\pi_{t_l}(\mu, w, \sigma^2|y_{1:150}) \propto \left(\prod_{i=1}^{150} f(y_i|\mu, w, \sigma^2)\right)^{t_l} p(\mu, w, \sigma^2),$$

where $p(\mu, w, \sigma^2)$ was the prior.

Our algorithm was compared to the conventional PT algorithm with the fixed parameters. For each of the parameters, ζ, γ and L, the parameter values of the conventional algorithm were shifted from the values tuned by the adaptive PT algorithm, $\hat{\zeta}_l$, $\hat{\gamma}_l$ and $\hat{L}$, as follows.

(a) $\zeta_l \leftarrow \hat{\zeta}_l \cdot \varphi_\zeta$, for $l = 1, \ldots, L$, $\varphi_\zeta = 0.5, 0.6, \ldots, 2, 3$,
$L \leftarrow \hat{L}$, $\gamma_l \leftarrow \hat{\gamma}_l$.

(b) $\gamma_l \leftarrow \hat{\gamma}_l \cdot \varphi_\gamma^2$, for $l = 1, \ldots, L$, $\varphi_\gamma = 0.1, 0.3, \ldots, 2, 3$,

(c) $L \leftarrow \hat{L} + \varphi_L$, $\varphi_L = -5, -4, \ldots, 4, 5$,
$\gamma_l \leftarrow \hat{\gamma}_l$, $\zeta_l \leftarrow \hat{\zeta}_l$.
(If $\varphi_L > 0$, ζ_l and γ_l were adapted for $l = \hat{L} + 1, ..., \hat{L} + \varphi_L$, for fairness of the comparison.)

We ran the adaptive PT algorithm and the conventional PT algorithms for 10^6 iterations. The initial sample values were $w_{l,m}^{(0)} = 1/6$, $\sigma_{l,m}^{2(0)} \sim IG(\alpha_g, \beta_g)$, $\mu_{l,m}^{(0)} \sim U[\min(y_{1:150}), \max(y_{1:150})]$ for each run. The initial parameter values of $\gamma_{1j}^{(0)}, \ldots, \gamma_{Lj}^{(0)}$ were the sorted L random numbers from $U[0.0001, 800]$. The variables of posterior were divided into four blocks, the numbers of which were (5,4,4,4). Each Metropolis algorithm updated for the every block.

We evaluated the sample mean of the posterior of μ_m, $m = 1, \ldots, 6$, as the estimator of $(\sum_{m=1}^{6} \mu_m)/6$ in 50 runs independently. The accuracy of the estimation was evaluated by the root mean square error (RMSE), which takes the root average of the errors of the six estimators to evaluate the total error of them, that is,

$$\mathrm{RMSE}(i) = \left(\frac{1}{6} \sum_{m=1}^{6} (\bar{\mu}_m(i) - 2.5)^2 \right)^{1/2},$$

where $\bar{\mu}_m(i)$ is the sample mean of posterior of μ_m in the ith trial, and 2.5 is the true value.

As a result, our algorithm can establish appropriate parameters and achieve very low RMSEs (Fig. 3.4). Fig. 3.4(a) and (b) show that the RMSEs of our algorithm were less than those of the conventional PT algorithms with shifted parameters. On the other hand, when the number of temperatures increases, the RMSEs don't increase (Fig. 3.4(c)) but the computational costs of the algorithms increase. In fact, the shifted inverse temperatures could not control the exchange ratios well (Fig. 3.5).

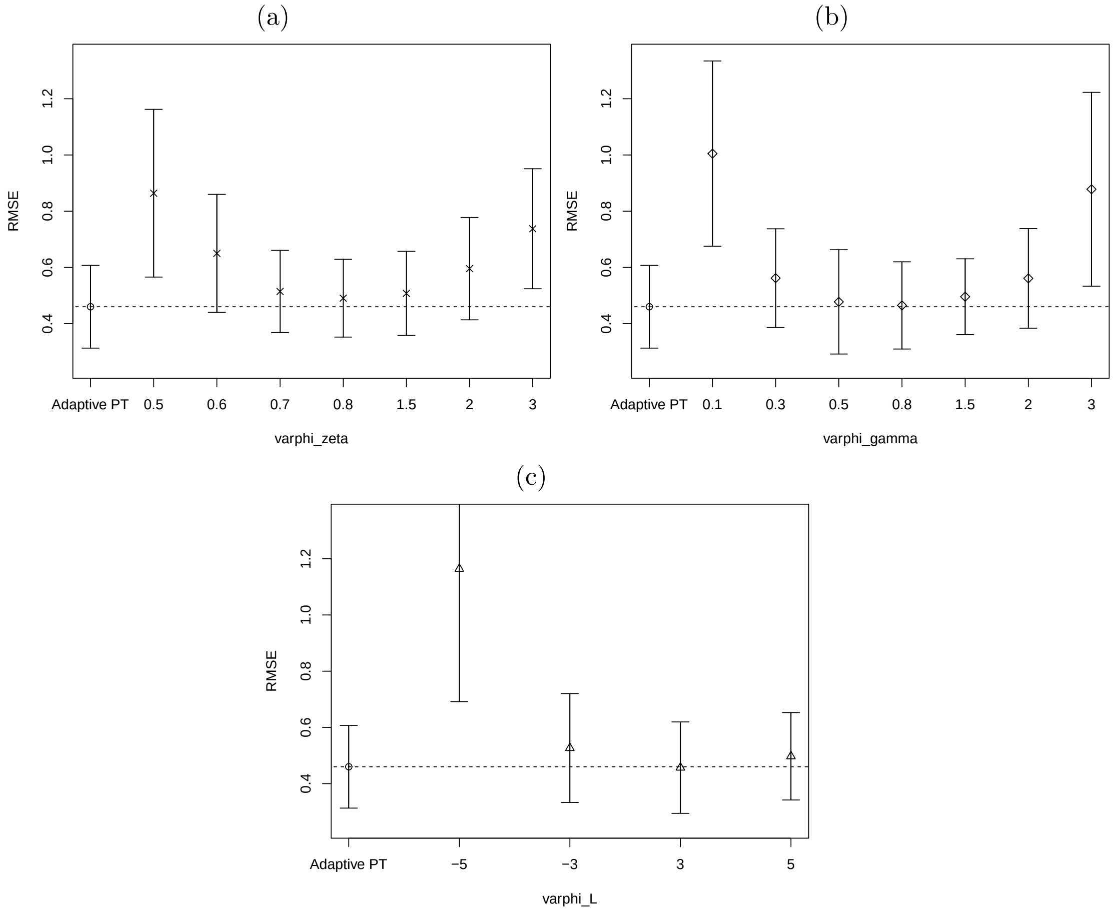

Figure 3.4. RMSEs in 50 runs. Each plot displays the average and the standard deviation of RMSEs for each algorithm by the mark and the error bar, respectively. The adaptive PT algorithm : (∘). Those of the conventional PT algorithms are plotted for each of shifted parameters. The inverse temperatures : × in (a), The proposal variances : ⋄ in (b), The number of inverse temperatures : △ in (c).

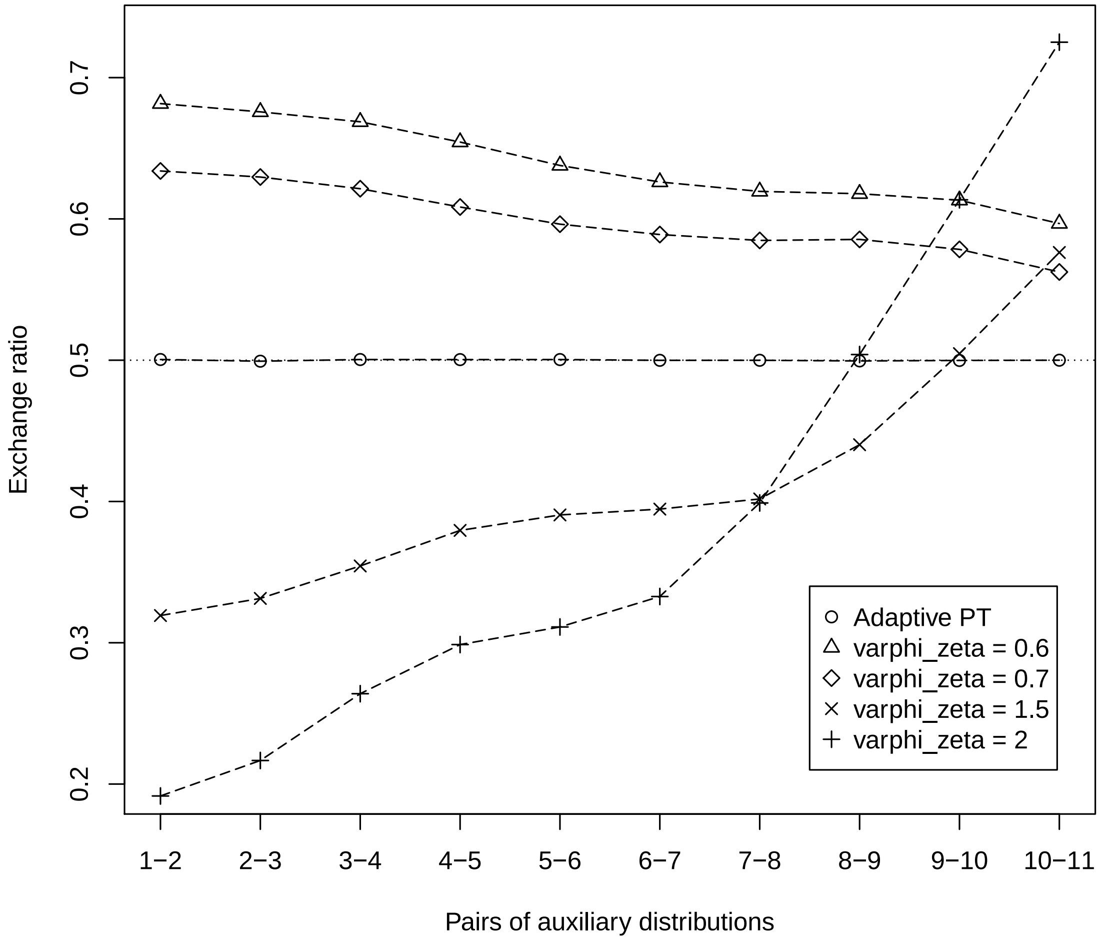

Figure 3.5. The averages of the estimated exchange ratios in 50 runs. Those of the adaptive PT algorithm were almost the predefined value (0.5), a dotted line, while those of the conventional PT algorithm with shifted inverse temperatures could neither be tuned to a constant nor a moderate value (0.5).

4.3 The predictive distribution for Galaxy Data

We calculated the predictive distribution of the normal mixture model for galaxy data. The galaxy data consist of velocities of 82 galaxies, and were first presented by Postman, Huchra and Geller (1986). We assumed the velocities $y_1, ..., y_n$ were independently generated from the mixture model Eq. (3.6) with $M = 4$. We sampled from the posterior that consist of the above model and priors, and the data. By using these samples, we estimated the predictive density

$$\begin{aligned} g(\tilde{y}|y_{1:n}) &= E[f(\tilde{y}|\mu, w, \sigma^2)|y_{1:n}] \\ &= \int\int\int f(\tilde{y}|\mu, w, \sigma^2)\pi(\mu, w, \sigma^2|y_{1:n})d\mu dw d(\sigma^2). \end{aligned} \tag{3.7}$$

The prior parameters were set to $\xi = 20,\ \kappa^2 = 10^2,\ \alpha_g = 11,\ \beta_g = 10,\ \varrho = 1.$

We ran the adaptive PT algorithm and the conventional PT algorithm for 10^5 iterations. The proposal variances of the conventional PT algorithm were $0.01^2\hat{\gamma}_l$, for $l = 1, ldots, L$, where $\hat{\gamma}_l$ were the proposal variances obtained by the adaptive PT algorithm, and the others were set as the parameters obtained by the adaptive PT algorithm. The initial values of the samples and the parameters were set by the way section 4.2.

We calculated the 20 predictive densities for each of the algorithms (Fig. 3.6 and 3.7). The predictive density for the adaptive PT algorithm didn't vary and were stable (Fig. 3.6), but on the other hand the predictive density for the conventional PT algorithm with the improper proposal variances varied and were unstable (Fig. 3.7).

Through these experiments, we found that the appropriate convergence order of the learning coefficients is $1/n$. When the convergence order of the learning coefficients is $1/n^2$, the learned parameters vary much because the learning coefficients converge extremely fast. The learning coefficients whose convergence order is $1/n^{1/2}$ converge too slowly, so that the convergences of the learning parameters are also extremely slow.

4.4 Comparison of the computational costs

Our algorithm and the conventional PT algorithm with the settings in section 4.1 and 4.2 were run for 10^3 iterations ten times independently, and the computational

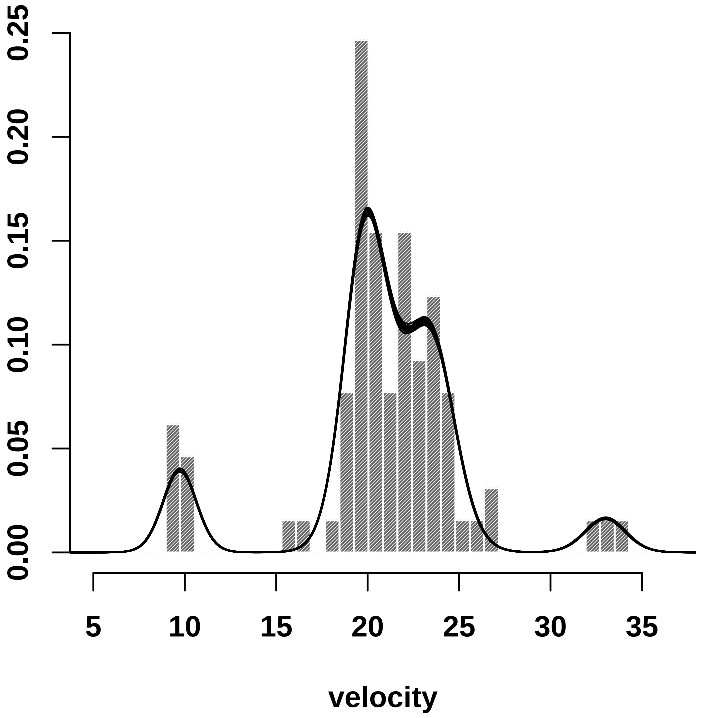

Figure 3.6. The histogram of the galaxy dataset and the 20 predictive densities by the adaptive PT algorithm.

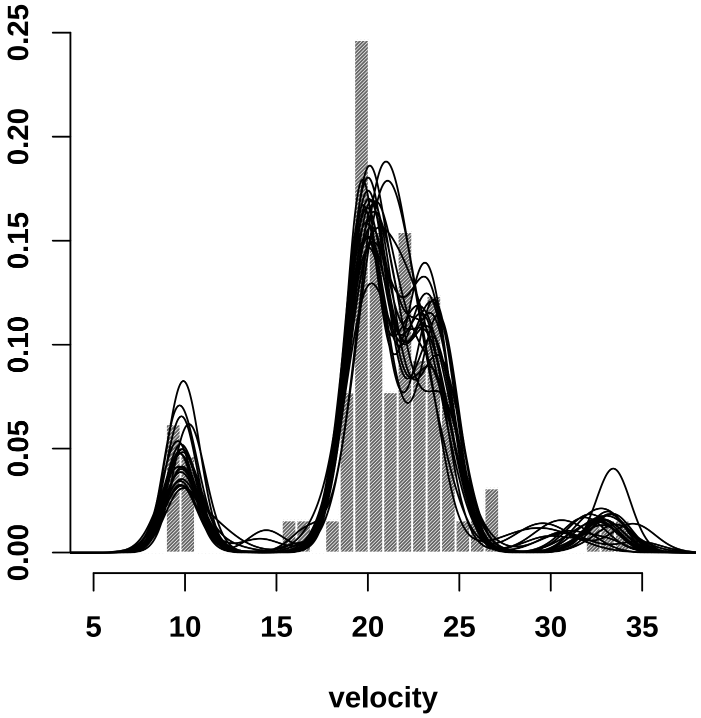

Figure 3.7. The histogram of the galaxy dataset and the 20 predictive densities by the conventional PT algorithm.

times are showed in the table 3.1. This result indicates the difference of their computational costs is negligible.

	Section 4.1	Section 4.2
Our algorithm	1.428 (0.0717)	9.729 (0.2641)
Parallel Tempering	1.383 (0.0596)	9.487 (0.2540)

Table 3.1. Averages and standard deviations of the ten computational times (second) of the algorithms. The standard deviations are enclosed by parentheses.

5. Conclusion

We proposed the adaptive PT algorithm that tunes its parameters while it runs, and showed that the algorithm can adapt its parameters on the fly so that samples mix rapidly by experiments with a mixture model. We also presented that the performance of the PT algorithm depends on its parameters and the adaptive PT algorithm finds good parameters through experiments for Bayesian estimation.

Chapter 4

Adaptive Gibbs variable selection

1. Introduction

Bayesian variable selection plays an important role in multivariate statistical analysis, and various sampling algorithms for Bayesian variable selection based on Markov chain Monte Carlo (MCMC; Robert and Casella (2004)) have been proposed. The most famous algorithms are Stochastic Search Variable Selection (SSVS; George and McCullogh (1993)), Kuo and Mallick's method (Kuo and Mallick (1998)) and Gibbs variable selection (GVS; Dellaportas et al. (2002)). The SSVS and the Kuo and Mallick's method are efficient only for the models whose conditional posterior densities can be obtained directly. On the other hand, the GVS is efficient not only for the above models but for the models whose conditional densities cannot be calculated by closed form, such as generalized linear models and nonlinear models, e.g., Markov mixture models, and so on. Reversible jump MCMC (RJMCMC; Green (1995)), which is a general algorithm for Bayesian model selection, is also available in such the case, but less efficient than the GVS (Dellaportas et al., 2002).

The GVS induces pseudo-priors that approximate the marginal posterior distributions of coefficient parameters and support the sampling of the GVS. The parameters of the pseudo-priors are determined by a pilot run for full model (Dellaportas et al. (2002)). However, the obtained parameters shift the pseudo-priors from the marginal posteriors of coefficients due to a correlation of the posterior distribution of coefficient parameters. The shifted pseudo-priors give the slow mixing of the GVS.

The GVS adopts random walk type Metropolis (RWM) algorithm as a sampling method of the coefficient parameters for the models whose conditional posterior distribution of coefficients cannot be obtained by a closed form. The covariance matrix of the proposal distribution is set by using samples from the pilot run for the full model Paroli and Spezia (2007). However, since the samples from the full model have not enough information to estimate the appropriate scale of the proposal distribution, above proposal covariances are often improper.

In this chapter, we propose an adaptive GVS that adapts the proposal covariances and pseudo-prior by learning the mean and the covariances of the coefficient posterior and the scale parameter on the fly. We show that our algorithm can obtain proper parameters during its run and is more efficient than the conventional GVS through their applications to the Bayesian variable selection of a logistic regression model.

2. Gibbs Variable Selection

Typically p-variate statistical models may have the coefficient parameter vector, $\theta = (\theta_1, ..., \theta_p)$, associated with covariates x_j, $j = 1, ..., p$, and an indicator variable vector, $\gamma = (\gamma_1, ..., \gamma_p)$ that presents which of the covariates are included in the model, that is, $gamma_j$ takes one if the covariate x_j is included, and zero otherwise. For example, a regression model, one of the most simple multivariate models, is written as $y = \sum_{j=1}^{p} x_j\theta_j + \epsilon$, where y is a response and ϵ is a noise. The Bayesian variable selection of the statistical models needs to estimate posteriors of the coefficient parameters, θ_j, and the indicator variables, γ_j, and the GVS generates samples from the posterior distributions.

The GVS sets $\theta_j = \gamma_j\beta_j$, where β_j is referred to as the effect size, and generates samples from the posterior of γ_j and β_j to obtain the samples from the posterior of γ_j and θ_j. If γ_j takes one, β_j is equal to the coefficient θ_j, and otherwise β_j is distributed according to a pseudo-prior $f_{\lambda_j}(\beta_j)$, where $\lambda_j \in \Lambda$ is a parameter vector. The pseudo-priors are not included in the posterior of θ and γ, but facilitate to produce the sample sequence from the posterior of γ and β. The prior of β_j given γ_j is

$$f_{\lambda_j}(\beta_j|\gamma_j) = \gamma_j f(\beta_j) + (1-\gamma_j) f_{\lambda_j}(\beta_j), \tag{4.1}$$

where $f(\beta_j)$ is a coefficient prior.

The GVS conducts Gibbs sampling steps for γ and β_γ, and a Metropolis-Hastings step for $\beta_{\setminus\gamma}$ by turns, where β_γ denotes the components of β included in the model, whose corresponding indicators, γ_j, take one, and $\beta_{\setminus\gamma}$ consists of the others. The Gibbs sampling step produces samples from the conditional posterior distributions

$$\begin{aligned} f_\lambda(\gamma_j|\gamma_{-j}, \beta, D) &\propto f(D|\beta, \gamma) \prod_{k=1}^{p} f_{\lambda_k}(\beta_k|\gamma_k) f(\gamma_k), \ j = 1, ..., p, \\ f_\lambda(\beta_{\setminus\gamma}|\gamma, \beta_\gamma, D) &= \prod_{\beta_j \in \beta_{\setminus\gamma}} f_{\lambda_j}(\beta_j), \end{aligned} \tag{4.2}$$

where γ_{-j} denotes the components of γ except γ_j and D denotes the observation data. The Metropolis-Hastings step executes the Metropolis-Hastings update for

$$f(\beta_\gamma|\gamma, \beta_{\setminus\gamma}, D) \propto f(D|\beta, \gamma) \prod_{\beta_j \in \beta_\gamma} f(\beta_j). \tag{4.3}$$

Practically if this conditional distribution (4.3) can be obtained analytically, this step directly samples from the distribution, that is, conducts a Gibbs sampling, otherwise a random-walk Metropolis (RWM) sampling is applied. In this dissertation, we employ the RWM sampling because it can be applied to more various models.

The pseudo-priors should well approximate the marginal coefficient posteriors, $f(\beta_j|\gamma_j = 1, D)$, and the proposal distribution should provide an appropriate average Metropolis-acceptance probability, typically 0.234 in multidimensional settings (Roberts et al., 1997), for the sake of rapid mixing of the GVS. A pilot run in the GVS samples from the posterior of the coefficients of the full model, $f(\beta|\gamma_1 = \cdots = \gamma_p = 1, D)$, and estimates the means and the covariances of the coefficient posterior, μ and Σ, by the sample means, $\hat{\mu}$, and the sample covariances, $\hat{\Sigma}$, respectively. The GVS employs the estimated parameters as those of the pseudo-priors and the proposal distribution (Dellaportas et al. (2002), Paroli and Spezia (2007)). However, the features of the pseudo-priors with the means

$\hat{\mu}_j$ and variances $\hat{\Sigma}_{jj}$ are different from those of the marginal coefficient posteriors, $f(\beta_j|\gamma_j = 1, D)$, because they are different from the features of the marginal coefficient posterior of the full model, $f(\beta_j|\gamma_1 = \cdots = \gamma_p = 1, D)$, from which the pilot run generates the samples, due to the correlation of the posterior of coefficients. Typically the proposal distribution has the covariances $c\hat{\Sigma}_{ij}$, where c is a scale parameter set as $(23.4)^2/p$ if the model is high dimensional (Roberts et al. (1997)), and one otherwise. Because the proper scale of the proposal distribution, which leads to the appropriate average Metropolis-acceptance probability, practically depends on the features of the coefficient posterior such as a dimension and a shape, the proposal distribution often provides the improper average Metropolis-acceptance probability.

3. Adaptive Gibbs Variable Selection

We propose the adaptive GVS algorithm that adapts the pseudo-prior parameters and the proposal covariances by learning covariances and means of the coefficient posterior and the scale parameter while the GVS is running.

The correlation coefficients of the proposal distribution, the variances and the means of the pseudo-prior should correspond with those of the coefficient posterior, and the scale of the proposal distribution should lead to the appropriate Metropolis acceptance rate such as 0.234. The adaptive GVS algorithm learns the covariances and the means of the coefficient posterior, μ and Σ, and the scale parameter of the proposal distribution, c, which provides the appropriate acceptance rate, by using generated samples $\beta^{(n)}$ and $\gamma^{(n)}$, and introduces the learned parameters $\mu^{(n)}$, $\Sigma^{(n)}$, and $c^{(n)}$ to the pseudo-priors and the proposal distribution. That is, the pseudo-priors have mean $\mu_j^{(n)}$ and variances $\Sigma_{jj}^{(n)}$ and the covariances of the proposal distribution are $c^{(n)}\Sigma_{ij}^{(n)}$ at the $(n+1)th$ iteration. The learning algorithms of μ, Σ and c are described as follows.

The covariance parameters, Σ_{ij}, and the mean parameters, μ_j, is updated as

$$
\begin{aligned}
\mu_j^{(n+1)} &\leftarrow \mu_j^{(n)} + \gamma_j^{(n+1)} h(a_j^{(n)})(\beta_j^{(n+1)} - \mu_j^{(n)}), \\
\Sigma_{ij}^{(n+1)} &\leftarrow \Sigma_{ij}^{(n)} + \gamma_j^{(n+1)} \gamma_i^{(n+1)} u(a_i^{(n)}, a_j^{(n)}) \\
&\qquad \times \left((\beta_j^{(n+1)} - \mu_j^{(n)})(\beta_i^{(n+1)} - \mu_i^{(n)}) - \Sigma_{ij}^{(n)} \right), \\
a_j^{(n+1)} &\leftarrow a_j^{(n)} + \gamma_j^{(n+1)}, \\
&\qquad\qquad j = 1, ..., p,\ i = 1, ..., p,
\end{aligned} \tag{4.4}
$$

where $a_j^{(0)} = 1$, and $a_j^{(n)}$ increases with updating jth parameters, μ_j and Σ_{ij} for i such that $\gamma_i = 1$. The learning coefficients $h(n)$ and $u(n, m)$ are decreasing functions that satisfy $\lim_{n\to\infty} h(n) = 0$ and $\lim_{n\to\infty, m\to\infty} u(n, m) = 0$, respectively. Note that the parameters are updated by only sample β_j for the coefficient posterior, $f(\beta_j|D, \gamma_j = 1)$.

To converge the mean acceptance rate to a specific value, $\alpha \in (0, 1)$, mainly 0.234, the scale parameter, c, is updated as

$$
c^{(n+1)} \leftarrow c^{(n)} + s_n(ER^{(n+1)} - \alpha), \tag{4.5}
$$

where $ER^{(n)}$ is a variable that takes one if a proposal value in the Metropolis sampling of $\beta_{\gamma^{(n+1)}}$ is accepted at time n, and zero otherwise. The learning coefficient, s_n, is a decreasing function of n that satisfies $\lim_{n\to\infty} s_n = 0$.

A pseudo code of the adaptive GVS algorithm is given in the Algorithm 1.

The total computational costs of the conventional GVS and the adaptive GVS are almost same, because the pilot run of the GVS often takes longer times than the learning steps of our algorithm.

The adaptive GVS converges. The proof is proved in chapter 5 by applying the convergence theorem of the adaptive MCMC algorithms for general auxiliary variable methods.

Algorithm 2 Adaptive GVS algorithm

```
Initialize $\beta^{(0)}$, $\gamma^{(0)}$, $\Sigma^{(0)}$, $\mu^{(0)}$ and $c^{(0)}$.
for $n = 0$ to $N - 1$ do
    (Gibbs sampling step)
    for $j = 1$ to $p$ do
        $\gamma_j^{(n+1)} \sim f_{\lambda_j^{(n)}}(\gamma_j | \gamma_{-j}^{(n)}, \beta^{(n)}, D)$, where $\lambda_j^{(n)} = (\mu_j^{(n)}, \Sigma_{j,j}^{(n)})$ and $\gamma_{-j}^{(n)} = (\gamma_1^{(n+1)}, \ldots, \gamma_{j-1}^{(n+1)}, \gamma_{j+1}^{(n)}, \ldots, \gamma_p^{(n)})$.
    end for
    $\beta_{\setminus\gamma^{(n+1)}}^{(n+1)} \sim \prod_{\beta_j \in \beta_{\setminus\gamma^{(n+1)}}} f_{\lambda_j^{(n)}}(\beta_j)$.
    (Metropolis sampling step)
    Generate $\beta_{\gamma^{(n+1)}}^{(n+1)}$ via the RWM algorithm for $f(\beta_{\gamma^{(n+1)}} | \gamma^{(n+1)}, \beta_{\setminus\gamma^{(n+1)}}^{(n+1)}, D)$, which has the proposal covariance matrix $c^{(n)}\Sigma_{\gamma^{(n+1)}}^{(n)}$, where $\Sigma_\gamma$ denotes the covariance matrix that consists of the covariances $\Sigma_{ij}$ where $\gamma_i = 1$ and $\gamma_j = 1$.
    (Parameter learning step)
    Update ($\mu^{(n)}$, $\Sigma^{(n)}$) to ($\mu^{(n+1)}$, $\Sigma^{(n+1)}$) by the Eq. (4.4).
    Update $c^{(n)}$ to $c^{(n+1)}$ by Eq. (4.5).
end for
```

4. Experiments

We estimated the marginal probability of inclusion for each of p covariates, x_j, and the predictive distribution, for the logistic regression model,

$$f(y|x, \beta, \gamma) = \frac{1}{1 + \exp\left(-y(\sum_{j=1}^{p} x_j \beta_j \gamma_j)\right)}, \tag{4.6}$$

where $y \in \{-1, 1\}$ is a response variable. The priors are

$$\begin{aligned} f(\beta_j) &= N(\mu_{\beta_j}, \sigma^2_{\beta_j}), \\ f(\gamma_j) &= \tau_j^{\gamma_j}(1-\tau_j)^{1-\gamma_j}, \\ f_\lambda(\beta_j) &= N(\mu_j, \sigma_j^2), \quad \lambda = (\mu_j, \sigma_j^2), \end{aligned}$$

where $N(\mu, \sigma^2)$ is a normal density with a mean μ and a variance σ^2, and $0 < \tau_j < 1$. The hyper-parameters were $\mu_{\beta_j} = 0$, $\sigma^2_{\beta_j} = 9$ and $\tau_j = 0.5$.

The learning coefficients of our algorithm were $h(n) = 1/(n+50)$, $u(n, m) = 1/\sqrt{(n+50)(m+50)}$ and $s_n = 1/(n+500)$. The initial parameter values $\mu^{(0)}$ and $\Sigma^{(0)}$ were set to a mode of the full model posterior density of β, and the minus inverse Hessian of the log full model posterior density at the mode, respectively. Other parameters were $\alpha = 0.234$ and $c^{(0)} = (2.38)^2/p$. The initial sample values $\beta^{(0)}$ and $\gamma^{(0)}$ were set to the mode of the posterior of the full model and the vector whose all elements are one, respectively.

In the GVS, the proposal scale was $(2.38)^2/p$, and the pilot run was executed for 10^4 iterations. The initial sample values $\beta^{(0)}$ and $\gamma^{(0)}$ were set to the sample mean of β in the pilot run and the vector whose all elements are one, respectively.

We used synthetic data and real data, cardiac Arrhythmia data.

4.1 Synthetic Data

We generated data from the true logistic model $f(y|x, \theta^*)$. The true model has 100 covariates and the coefficients, $\theta^*_j = \{-0.5;$ for $j = 1, \ldots, 5, 71, \ldots, 75$, -0.1; for $j = 31, \ldots, 35$, 1; for $j = 51, \ldots, 55$, 0.1; for $j = 96, \ldots, 100$, 0; *otherwise*$\}$. That is, the true model includes the 25 covariates. The covariates were generated from the normal distribution whose means are all 0 and variances are all 1 and covariances between x_j and x_i for $i, j = 1, \ldots, 30$, are 0.8 and covariances between

x_j and x_i for $i, j = 71, \dots, 100$, are 0.7. We independently generated 300 data from the model.

We ran the adaptive GVS and the conventional GVS for 2^5 iterations and generated samples from the posterior based on the synthetic data and the model. We calculated the estimated marginal probabilities of inclusions $\hat{P}_j^{(n)} = \frac{1}{n}\sum_{i=1}^{n} \gamma_j^{(i)}$ every 10-th iteration. We also calculated the error of the estimated probabilities $\sum_{j=1}^{100} |\hat{P}_j^{(n)} - P_j^*|$, where P_j^* takes one if x_j is included in the model, and 0 otherwise. We calculated the errors in 10 runs and the mean and the standard deviation of them every 10 iterations (Fig. 4.1). The mean of the errors in the adaptive GVS converged to the lowest value faster than that in the conventional GVS, and the standard deviation in the adaptive GVS was much smaller than that in the conventional GVS.

4.2 Cardiac Arrhythmia Data

We considered the cardiac arrhythmia data $\{y_{1:n_d}, x_{1:n_d}\}$ (Guvenir et al., 1997), which contains 257 covariates and 452 instances, i.e., $n_d = 452$. The 245 instances are normal, $y = 0$, and the others have disease, $y = 1$. We excluded covariates which contain missing values.

We calculated the estimated marginal probabilities of inclusions $\hat{P}_j^{(n)}$ and the estimated predictive distribution

$$\hat{f}(y_{1:n_d}|x_{1:n_d}) = \frac{1}{m}\sum_{n=1}^{m} f(y_{1:n_d}|x_{1:n_d}, \beta^{(n)}, \gamma^{(n)}),$$

where m denote the number of the iterations in post burn-in, and $f(y_{1:n_d}|x_{1:n_d}, \beta, \gamma) = \sum_{i=1}^{n_d} f(y_i|x_i, \beta, \gamma)$. For a comparison, we used the estimated inefficiency factor (IF)

$$1 + 2\sum_{i=1}^{M}\left(1 - \frac{i}{m}\right)\hat{\rho}(i), \tag{4.7}$$

where $\hat{\rho}(i)$ is an estimated autocorrelation of the sample sequence after burn-in, and M is a truncation point after which the estimated autocorrelation $\hat{\rho}$ is negligible, and chosen based on when $\hat{\rho}$ decays to zero. The estimated IF is proportional to a variance estimator of the sample mean, which are $\hat{P}_j^{(m)}$ and $\hat{f}(y_{1:n_d}|x_{1:n_d})$ in this experiment.

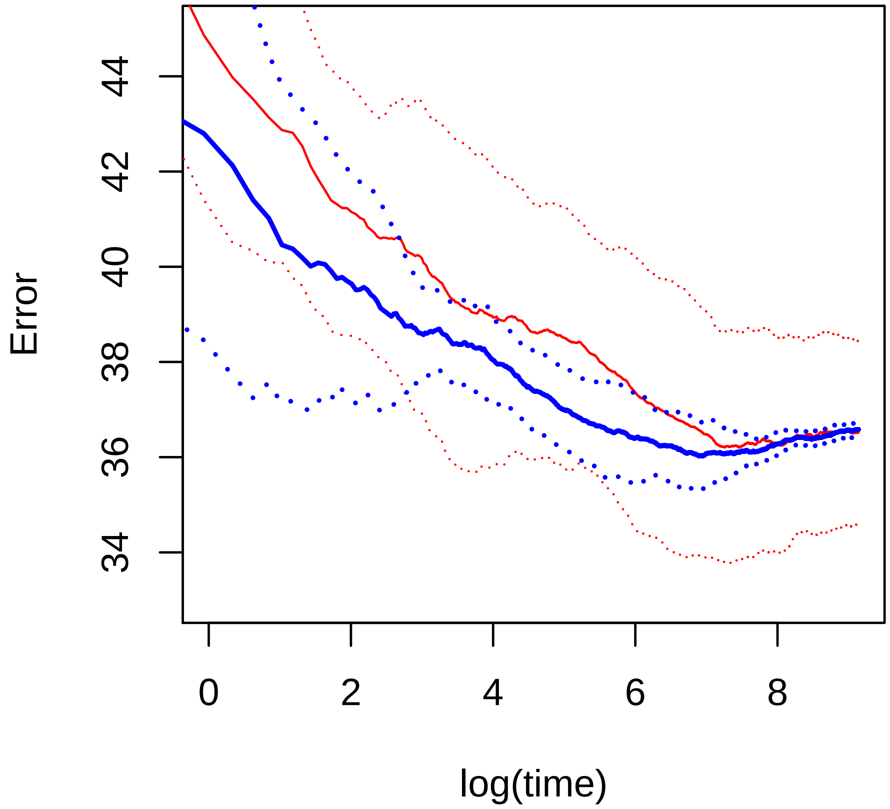

Figure 4.1. Trace plots of the means (solid line) and the standard deviations (dotted line) of the errors in 10 runs. Bold and blue lines: the adaptive GVS. Thin and red lines: the conventional GVS.

The sampling algorithms were run for 3×10^5 iterations, and the burn-in periods were 10^5. The sample set was chosen from every 10th sample.

The parameters $\mu_1^{(n)}$ and $\Sigma_{11}^{(n)}$ converge quickly (Fig. 4.2). The others also converge as fast as them. The estimated IFs for $\hat{P}_j^{(n)}$ of our algorithm were lower than those of the conventional GVS (Fig. 4.3). The predictive distribution estimated by our algorithm converged faster than that estimated by the conventional GVS (Fig. 4.4), and the estimated IFs for the estimated predictive distribution, $\hat{f}(y_{1:n_d}|x_{1:n_d})$, of our algorithm and that of the conventional GVS are 154.7 and 321.5, respectively.

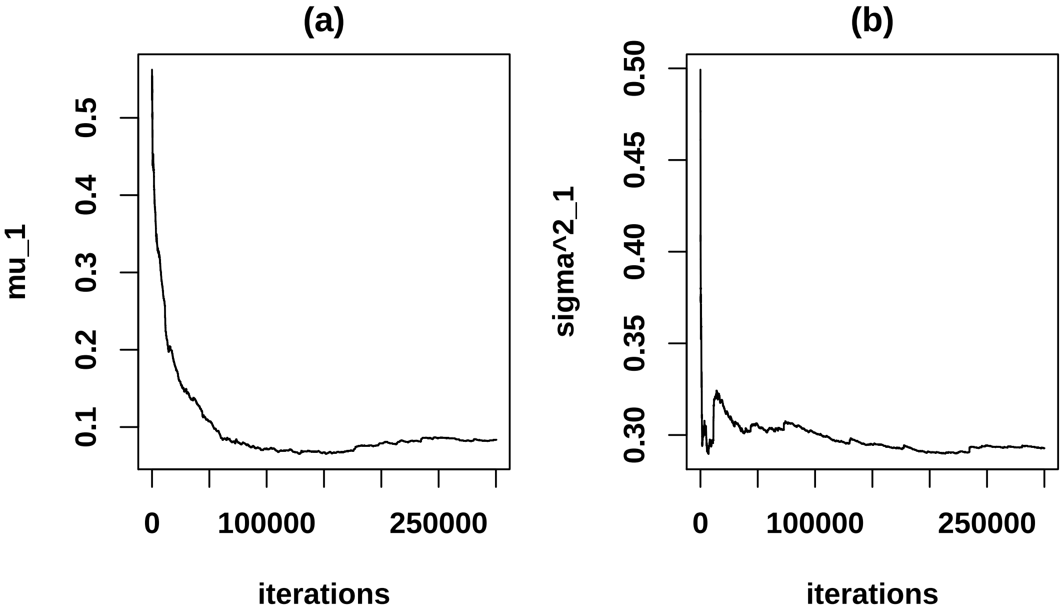

Figure 4.2. trace plot (a):posterior mean $\mu_1^{(n)}$, (b):posterior variance $\Sigma_{11}^{(n)}$

The estimated mean Metropolis acceptance rate of our algorithm was 0.2328, which is close to the target value, 0.234. This leads to well mixing of our algorithm. The learned covariance and mean parameters were more close to those of the marginal posterior distributions of the coefficients than those obtained by a pilot run (Fig. 4.5). Thus the pseudo-priors of our algorithm were closed to the marginal posterior distributions of the coefficients, which improved the mixing of our algorithm.

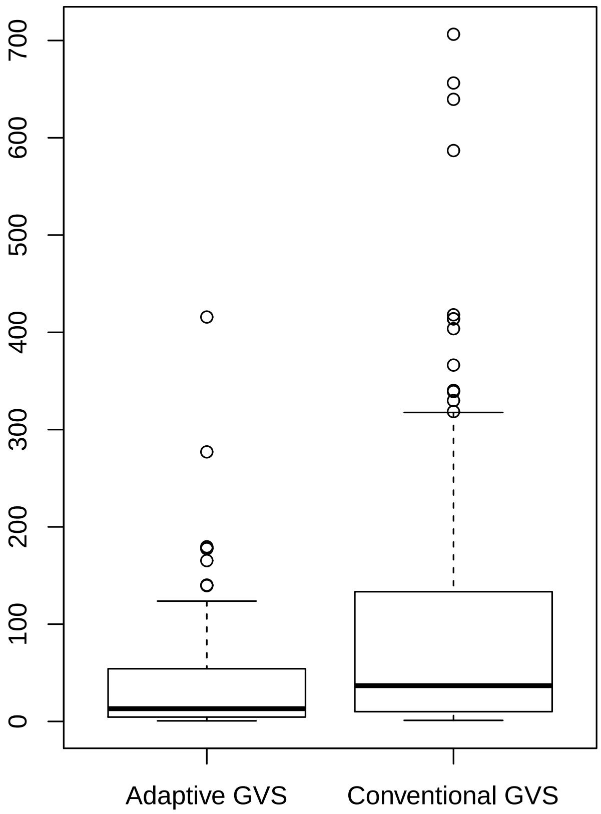

Figure 4.3. Box plots of the estimated IFs for $\hat{P}_j^{(n)}$ except those of the covariates whose estimated probability of inclusion was 1.

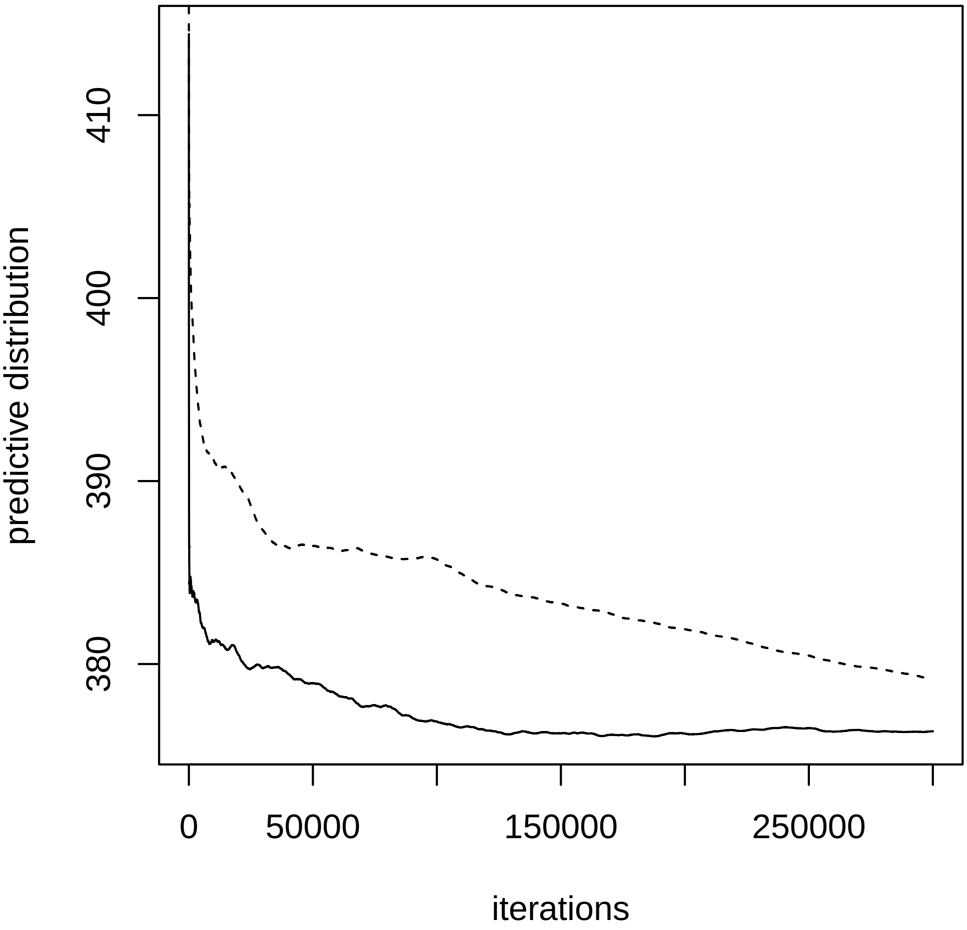

Figure 4.4. Trace plots of the estimated predictive distributions by the adaptive GVS (solid line) and the conventional GVS (dashed line).

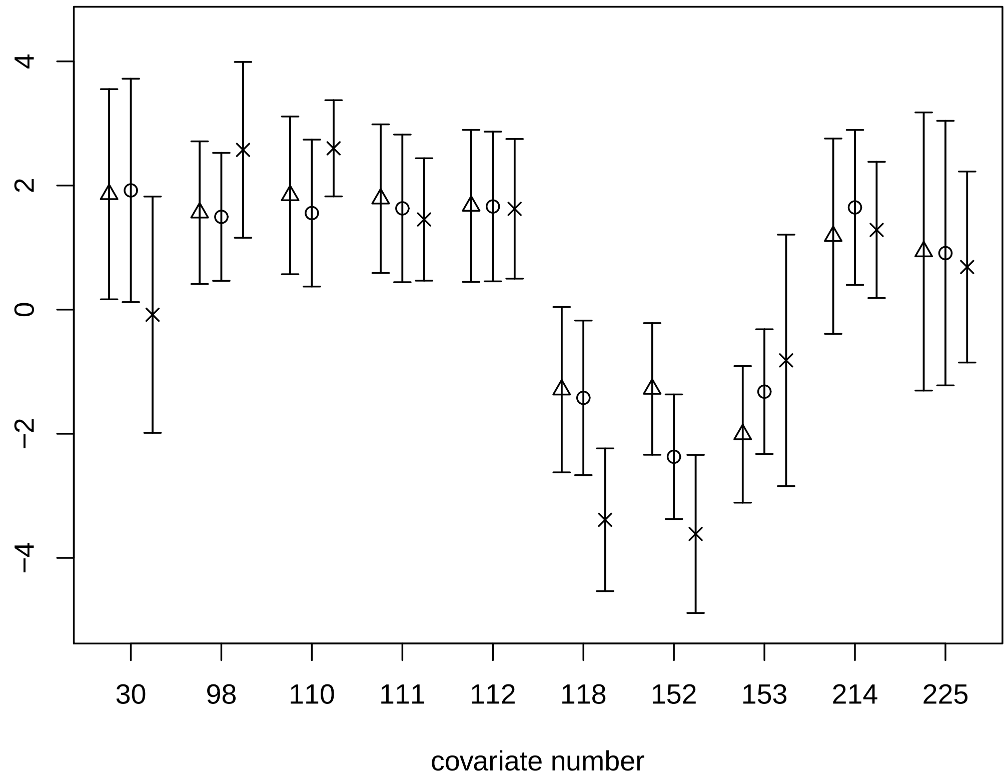

Figure 4.5. Means and standard deviations of the coefficient posterior. Each plot displays a mean and a standard deviation by the mark and the radius of the error bar, respectively. The means and the standard deviations learned by the adaptive GVS : (○). The sample means and standard deviations by the conventional GVS : (△) and the pilot run for the full model : (×). These 10 covariates have the estimated probability of inclusion which are the closest to 0.5.

From the results of these numerical experiments, the appropriate convergence order of the learning coefficients seems to be $1/n$ as well as that of the adaptive PT algorithm.

5. Conclusion

In this chapter, we proposed an adaptive algorithm that adapts parameters of a proposal distribution and pseudo-priors during generating samples. We also showed the proposed algorithm mixes faster than the conventional GVS through the two experiments of the Bayesian variable selection of the logistic regression model.

Chapter 5

Generalization to auxiliary variable methods

1. Introduction

The PT algorithm and the GVS use auxiliary distributions, the tempered distributions and the pseudo-priors. The auxiliary distributions are also used in other several algorithms, e.g., the cluster Monte Carlo methods that efficiently produce samples by block-wise updates based on auxiliary distributions (Swendsen and Wang 1987; Higdon 1998). These algorithms are referred to as auxiliary variable methods (AVMs) in this dissertation.

Although the performance of the standard MCMC methods such as Metropolis-Hastings algorithm (Hastings 1970) depend on only the proposal distribution, the performance of an AVM depends on both the proposal distribution and the auxiliary distributions. Hence the parameters of the proposal and auxiliary distributions have to be chosen so that the Markov chain of the AVM mixes as fast as possible. They have been tuned by rough methods or trial-and-error in pilot runs so far because their relationship to the mixing speed has not been clear.

For the standard MCMC methods, Gilks et al. (1998) and Haario et al. (2001) proposed adaptive MCMC algorithms that tuned the parameters of a proposal distribution by using past samples during runs. Haario et al. (2001) also proved the convergence theorem of their algorithms, which was developed later (Andrieu and Moulines 2006; Roberts and Rosenthal 2007).

The AVMs have the parameters not only in the proposal distribution, but also in the auxiliary distributions. Thus the above adaptive MCMC algorithms can not be applied to the AVMs.

In this chapter, we propose an adaptive MCMC for AVMs by extending the above adaptive algorithms to general AVMs, where the algorithm adapts the parameters of the proposal and the auxiliary distributions of AVMs on the fly. We prove the convergence theorems of our algorithm in a similar way to Roberts and Rosenthal (2007). We also prove the convergence of the adaptive PT algorithm and the adaptive GVS by using the theorem of the general adaptive MCMC for AVMs.

2. Adaptive MCMC for AVMs

The idea of the adaptive PT algorithm and the adaptive GVS is applicable to general AVMs. AVMs are mathematically formulated as below.

Let $\pi(dx)$ be a distribution on a state space $\mathcal{X}$ with σ-algebra $F_{\mathcal{X}}$ and $\pi_\lambda(dy|x)$ be a conditional distribution on a state space $\mathcal{Y}$ with σ-algebra $F_{\mathcal{Y}}$ given $F_{\mathcal{X}}$, where $\lambda \in \Lambda$ is a parameter vector. Then, the marginal distribution on $\mathcal{X}$ of the joint distribution $\pi_\lambda(dx, dy) = \pi_\lambda(dy|x)\pi(dx)$ is $\pi(dx)$ irrespective of $\pi_\lambda(dy|x)$.

In case of MCMC methods with auxiliary variables, $\pi(dx)$ corresponds to the target distribution and $\pi_\lambda(dy|x)$ to the auxiliary distributions. We term an MCMC method that draw samples (x', y') from $\pi_\lambda(dx, dy)$ to obtain x' an auxiliary variable method. In the PT algorithm, for example, the auxiliary distributions are $\pi_\lambda(dy|x) = \prod_{l=2}^{L} \pi_{t_l}(dx_l)$, $\lambda = (t_2, \ldots, t_L)$ and the auxiliary variables are $y = (x_2, ..., x_L)$.

In order to introduce adaptation, we need to consider time-varying parameters. Let $\{P_\theta((x, y), (dx, dy))\}_{\theta\in\Theta}$ be a family of Markov transition kernels on $\mathcal{X} \times \mathcal{Y}$ with stationary distribution $\pi_\lambda(dx, dy)$, that is,

$$\begin{aligned} (\pi_\lambda P_\theta)(dx, dy) &= \iint_{x',y'} \pi_\lambda(dx', dy') P_\theta((x', y'), (dx, dy)) \\ &= \pi_\lambda(dx, dy), \end{aligned}$$

where θ contains λ. Then, the adaptive MCMC for AVMs updates the parameters θ during generating chains $(x^{(n)}, y^{(n)})$ by P_θ as the following pseudo code.

Algorithm 3 Adaptive MCMC for AVMs

```
  **Initialize** $(x^{(0)}, y^{(0)}), \theta^{(0)}$.
  **for** $n = 0$ to $N-1$ **do**
    [1] $(x^{(n+1)}, y^{(n+1)}) \sim P_{\theta^{(n)}}((x^{(n)}, y^{(n)}), (dx, dy))$
    [2] Update $\theta^{(n)}$ to $\theta^{(n+1)}$ by using the result of step 1 such as $(x^{(n+1)}, y^{(n+1)})$.
  **end for**
```

In the adaptive PT algorithms, for example, the time-varying parameter vector is $\theta = (\gamma_1, \ldots, \gamma_L; t_2, \ldots, t_L)$.

3. Convergence Theorem

Atchade (2011) and Fort et al. (2011) proved convergence theorems of adaptive MCMC algorithms that adapt the parameters of the target distribution. The conditions for convergence in their theorems are, however, technical and strict. For example, the stationary distribution must converge. These conditions will considerably restrict the available parameter learning algorithms.

In this section, we show some convergence theorems that our algorithm in the previous section converges under weaker conditions. Here, convergence means that an algorithm is ergodic, that is,

$$\lim_{n\to\infty} \|A^{(n)}((x, y, \theta), dx) - \pi(dx)\| = 0, \qquad \forall (x, y) \in \mathcal{X} \times \mathcal{Y}, \theta \in \Theta,$$

where $\|\mu(dx) - \nu(dx)\| = \sup_{A\in\mathcal{F}_\mathcal{X}} |\mu(A) - \nu(A)|$ and

$$A^{(n)}((x, y, \theta), B_\mathcal{X}) = P\left[x^{(n)} \in B_\mathcal{X} | x^{(0)} = x, y^{(0)} = y, \theta^{(0)} = \theta\right], \quad B_\mathcal{X} \in F_\mathcal{X}.$$

Theorem 1 *The adaptive MCMC for AVM is ergodic if the following conditions hold:*

(a) Simultaneous uniform ergodicity

$$\begin{aligned} &\forall \varepsilon > 0,\ \exists N \in \mathbb{N}\ s.t. \\ &\qquad ||P_\theta^N((x, y), dx) - \pi(dx)|| \leq \varepsilon, \quad \forall (x, y) \in \mathcal{X} \times \mathcal{Y}, \theta \in \Theta. \end{aligned} \tag{5.1}$$

(b) Diminishing adaptation

$$\lim_{n\to\infty} \sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \| P_{\theta^{(n+1)}}\left((x,y),(dx,dy)\right) - P_{\theta^{(n)}}\left((x,y),(dx,dy)\right) \|$$
$$= 0 \textit{ in probability.} \tag{5.2}$$

Proof 1 *See A.*

The above conditions do not require that the auxiliary parameter $\lambda^{(n)}$ and the stationary distribution $\pi_{\lambda^{(n)}}$ converge. The condition (a) can be replaced with more concrete condition that checks only properties of the Markov transition kernel as follows.

(a') (Simultaneously strongly aperiodically geometrical ergodicity) There exists $C \in \mathcal{F}_{\mathcal{X}\times\mathcal{Y}}$, $V : \mathcal{X}\times\mathcal{Y} \to [1,\infty)$, $\delta > 0$, $\tau < 1$, and $b < \infty$, such that $\sup_C V < \infty$ and the following conditions hold for all $\theta \in \Theta$.

(i) (Strongly aperiodic minorisation condition) There exist a probability measure $\nu_\theta\left(dx,dy\right)$ on C such that

$$P_\theta((x,y),(dx',dy')) \geq \delta\nu_\theta(dx',dy'), \quad \textit{for all } x,y\in C.$$

(ii) (Geometric drift condition)

$$\left(P_\theta V\right)(x,y) \leq \tau V(x,y) + b\mathbf{1}_{\{C\}}(x,y), \quad \textit{for all } x,y \in \mathcal{X}\times\mathcal{Y},$$

where $(P_\theta V)(x,y) \equiv \iint P_\theta\left((x,y),(dx',dy')\right)V(x',y')dx'dy'$, and $\mathbf{1}_{\{\cdot\}}(x)$ is an indicator function.

Theorem 2 *The adaptive MCMC for AVM is ergodic if the condition (b) in Theorem 1, the condition (a') and $E[V(x^{(0)},y^{(0)})] < \infty$ hold.*

Proof 2 *Straightforward from Proposition 3 and the proofs of Theorem 3 in Roberts and Rosenthal (2007), and Theorem 1.*

Theorem 3 (Weak law of large numbers) *Suppose the adaptive MCMC for AVM satisfies the conditions (a) and (b) and let $g : \mathcal{X} \to \mathbb{R}$ be a bounded measurable function. Then,*

$$\frac{1}{n}\sum_{i=1}^{n} g(x^{(i)}) \to \int g(x)\pi(dx) \quad \textit{in probability}$$

as $n \to \infty$, for any initial values $(x,y) \in \mathcal{X}\times\mathcal{Y}$ and $\theta \in \Theta$.

Proof 3 *Straightforward from the coupling argument (Roberts and Rosenthal 2007).*

The convergence of the adaptive PT algorithm is proved by applying Theorem 2 as below.

Theorem 4 *The adaptive PT algorithm is ergodic if the following conditions hold:*

(s1) *The support S of the target distribution $\pi(dx)$ is compact and the density $\pi(x)$ is continuous and positive on S.*

(s2) *The family of proposal densities $\{q_\gamma\}_{\gamma\in\Gamma^p}$ is continuous and positive on $S^2 \times \Gamma^p$, where $\Gamma = [c, C]$.*

Proof 4 *See B.*

It will be possible to remove the assumptions that S is compact by extending Theorem 6 of Bai et al. (2011).

To prove the convergence of the adaptive GVS, we formulate the GVS as AVMs. We consider that the target distribution in the GVS is the joint posterior distribution of indicator variables γ_j and coefficients $\theta_j = (\gamma_j\beta_j)$,

$$f(\theta, \gamma|D) \propto f(D|\theta) \prod_{j=1}^{p} \big(\gamma_j f(\theta) + (1-\gamma_j)\delta_{\{0\}}(\theta)\big) f(\gamma_j), \tag{5.3}$$

where $\delta_{\{x\}}(y)$ is an indicator function, and $f(\theta_j)$ is a coefficient prior in Eq. (4.1). Note that this target distribution has no parameters. We also consider that β_j are auxiliary variables.

Thus we prove that the samples of γ_j and θ_j generated by the adaptive GVS converge to the posterior of them. The convergence theorem is proved as follows.

Theorem 5 *The adaptive GVS is ergodic if the following conditions hold:*

(a) *Either the support of the $f(D|\beta, \gamma)$ or the supports of priors, $f(\beta_j)$ and $f_{\lambda_j}(\beta_j)$, are compact set S and the $f(D|\beta, \gamma)$, $f(\beta_j)$ and $f_{\lambda_j}(\beta_j)$ are continuous and positive on S.*

(b) *The family of proposal densities* $\{q_{\Sigma,\mu,c}\}_{\Sigma,\mu\in\Omega}$ *is continuous and positive on* $S^2 \times \Omega$. $\Omega = \mathcal{Z} \times \mathcal{K} \times \mathcal{C}$*, where* $\mathcal{Z}$ *is a compact set of* $\mathbb{R}^{p^2}$ *and* $\mathcal{K}$ *is a bounded set on* $\mathbb{R}^p$ *and* $\mathcal{C}$ *is a bounded set on* $\mathbb{R}_+$*.*

Proof 5 *Similar arguments to the proof of Theorem 4.*

It will also be possible to remove the compact support assumptions by extending Theorem 6 of Bai et al. (2011).

4. Conclusion

This chapter proposed the adaptive MCMC for AVMs that learns parameters of proposal distributions and auxiliary distributions simultaneously while AVMs run, and proved convergence theorems that give weak sufficient conditions for convergence.

We also proved the convergence of the adaptive PT algorithm and the adaptive GVS by applying the convergence theorem of the adaptive MCMC for AVMs.

Chapter 6

Conclusion

This dissertation proposed the adaptive MCMC algorithms for the PT algorithm and the GVS, and generalized them to the adaptive MCMC algorithms for AVMs. We also proved the convergence theorems of the proposed algorithms.

Firstly, we extended the PT algorithm to the adaptive algorithm that adapts its parameters while it runs, and showed that the extended algorithm can obtain the proper parameters via numerical experiments for Bayesian estimation.

Secondly, we extended the GVS to the adaptive algorithm that adapts its parameters on the fly, and confirmed that the extended algorithm is more efficient than the GVS with the parameters obtained by the conventional method through the numerical experiments for Bayesian Variable Selection.

Finally, we generalized the proposed algorithms to the adaptive MCMC for general AVMs that adapts the parameters of the AVMs on the fly, and proved its convergence theorems that have mild sufficient conditions for the convergence. We also proved the convergence of the adaptive PT algorithm and the adaptive GVS by applying the convergence theorem of the adaptive MCMC for AVMs.

1. Discussion

The learning coefficients a_n^l, b_n in the adaptive PT algorithm and $h(\cdot)$, $u(\cdot, \cdot)$ in the adaptive GVS control convergence speeds of the corresponding updating parameters. As the convergence speeds of the learning coefficients get faster, the convergence speeds of the updating parameters also get faster but the variances of the converged parameters increase. Thus the learning coefficients should converge with moderate speed. However, since our adaptive algorithms are robust for the convergence speeds of the learning coefficients, we don't need to tune the learning coefficients in detail. Practically, the adaptive PT algorithm with the same learning coefficients performed well in the three numerical experiments. Also the adaptive GVS was more efficient than the GVS in the two numerical experiments in spite of using the same learning coefficients.

Users of the PT algorithm have needed to carefully tune the inverse temperatures, proposal variances and the number of inverse temperatures through many preliminary runs so far. The adaptive PT algorithm enables us to obtain the appropriate parameters automatically while the algorithm runs. Therefore the computational and artificial cost of the parameter choice of the PT algorithm is removed and the users are released from the tuning work.

Conventionally the parameters of the GVS are determined by using the samples from the posterior of the full model, and the proper parameters are mostly not obtained. The adaptive GVS can update its parameters to more appropriate values than those from the conventional method on the fly, and thus allows to generate samples more efficiently than the GVS with the parameters obtained by the conventional method.

The adaptive MCMC for AVMs and its convergence theorems provide guides of the extensions of other AVMs such as cluster Monte Carlo methods to the adaptive algorithms. The convergence theorems give knowledge that how parameter-update algorithms are able to be induced to the AVMs. The convergence theorems also show that various algorithms that update the parameters are available to the adaptive MCMC for AVMs.

2. Future works

Although we discussed the PT algorithm in a real space so far, we consider the idea of adaptation is applicable to that in a discrete space.

The GVS will be efficient for the Bayesian variable selection of more complex model such as a structural equation model and a non-Gaussian graphical model, so that we will apply the adaptive GVS to these models.

We will apply our adaptive framework of AVMs to other AVMs such as a partial decoupling method, which is one of the cluster Monte Carlo methods, and so on.

Acknowledgements

博士課程の３年間ご指導頂きました主指導教官である池田和司先生に心より感謝申し上げます．入学前に数理情報学研究室を訪問し，マルコフ連鎖モンテカルロ法の研究を行いたいと申し上げたところ，先生の専門分野でないにも関わらず研究室の受け入れを快諾して頂きました．それから３年間，研究内容，プレゼンテーション，論文執筆などあらゆる面において丁寧で適確な指導を頂きました．また，重要な文献を教えて頂いたり，重要な研究集会を紹介し派遣して頂くなど徹底して研究をサポートして頂きました．これらによって，私はこの３年間著しく成長を遂げることができました．

笠原正治先生には大変お忙しいなか私の副指導教官を引き受けて頂き誠にありがとうございました．公聴会，最終審査でのご指摘や本博士論文の添削などありがとうございました．また，大規模システム管理研究室の学生室の一角を数理情報学研究室に提供して頂きました．そこで一年間作業させて頂きましたが，非常に快適な環境でストレスなく集中して研究することができました．大変感謝しております．

松本裕二先生には大変お忙しいなか私の副指導教官を引き受けて頂き誠にありがとうございました．公聴会，最終審査でのご指摘ありがとうございました．また，公募申請における照会可能者に指定させて頂くことを快諾して頂きました．大変感謝しております．

渡辺一帆先生には研究に限らず学生生活も含めた様々な面で相談に乗って頂いて助けて頂きました．まず研究においては，機械学習についての議論，セミナー，学会の発表練習，予稿集や論文など様々な原稿の添削など様々な指導ありがとうございました．また，学生生活における様々な問題について相談に乗って頂き，何度も精神的に救われました．本当にありがとうございました．

竹之内高志先生には主にセミナー発表時において鋭い意見を沢山いただいたおかげで，多様な視点から自分の研究を省みることができました．誠にありがとうございました．

柴田智広先生，作村祐一先生，久保孝富先生にはセミナーや論文紹介などにおいてアドバイスを頂きました．誠にありがとうございました．

大規模システム管理研究室事務補佐員の橋本洋子さんには私が他研究室の学生であるにも関わらず，設備や騒音問題など作業環境の改善に取り組んで頂きました．誠にありがとうございました．数理情報学研究室秘書の谷本史さん，足立敏美さんには出張申請などの書類手続きをはじめ様々な面で学生生活をサポート

していただき，誠にありがとうございました．数理情報学研究室の学生の皆様にはしばしば食事に付き合って頂きありがとうございました．大変楽しい時を過ごすことができました．また，研究設備や学生生活などについて相談に乗っていただきありがとうございました．

最後に博士後期課程での研究生活を応援してくれた家族に心から感謝申し上げます．

Bibliography

Andrieu, C., Moulines, E., 2006. On the ergodicity properties of some adaptive MCMC algorithms. Annals of Applied Probability 16 (3), 1462–1505.

Atchade, Y., 2011. A computational framework for empirical Bayes inference. Stat. Comput. 21 (4), 463–473.

Bai, Y., Roberts, G., Rosenthal, J., 2011. On the containment condition for adaptive Markov chain Monte Carlo algorithms. Advances and Applications in Statistics 21 (1), 1–54.

Dellaportas, P., Forster, J. J., Ntzoufras, I., 2002. On Bayesian model and variable selection using MCMC. Stat. Comput. 12 (1), 27–36.

Fort, G., Moulines, E., Priouret, P., 2011. Convergence of adaptive and interacting Markov chain Monte Carlo algorithms. Ann. Stat. 39 (6), 3262–3289.

Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6, 721–741.

George, E., McCullogh, R., 1993. Variable selection via Gibbs sampling. Journal of the American Statistical Association 88 (423), 881–889.

Geyer, C., 1991. Markov chain Monte Carlo maximum likelihood. Proc. 23rd Symp. Interface Comput. Sci. Statist, 156–216.

Gilks, W., Roberts, G., Sahu, S., 1998. Adaptive Markov chain Monte Carlo through regeneration. Journal of the American Statistical Association 93 (443), 1045–1054.

Goswami, G., Liu, J., 2007. On learning strategies for evolutionary Monte Carlo. Stat. Comput. 17 (1), 23–38.

Green, P., 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika 82 (4), 711–732.

Guvenir, H., Acar, B., Demiroz, G., Cekin, A., 9 1997. A supervised machine learning algorithm for arrhythmia analysis. Proceedings of the Computers in Cardiology Conference, 433 – 436.

Haario, H., Saksman, E., Tamminen, J., 2001. An adaptive Metropolis algorithm. Bernoulli 7 (2), 223–242.

Hastings, W., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57 (1), 97–109.

Higdon, D. M., 1998. Auxiliary variable methods for Markov chain Monte Carlo with applications. J. Am. Stat. Assoc. 93 (442), 585–595.

Hukushima, K., 1999. Domain-wall free-energy of spin glass models : Numerical method and boundary conditions. Physical Review E 60, 3606–3613.

Hukushima, K., Nemoto, K., 1996. Exchange Monte Carlo method and application to spin glass simulations. Journal of the Physical Society of Japan 65 (6), 1604–1608.

Jasra, A., Stephens, D., Holmes, C., 2007. On population-based simulation for statics inference. Stat. Comput. 17 (3), 263–279.

Kuo, L., Mallick, B., 1998. Variable selection for regression models. Sankhya Ser. B 60, 65–81.

Liu, J., 2001. Monte Carlo Strategies in Scientific Computing. Springer, New York.

Nagata, K., Watanabe, S., 2008. Asymptotic behavior of exchange ratio in exchange Monte Carlo method. Neural Networks 21 (7), 980–988.

Paroli, R., Spezia, L., 2007. Bayesian variable selection in Markov mixture models. Communications in Statistics - Simulation and Computation 37 (1), 25–47.

Robert, C., Casella, G., 2004. Monte Carlo Statistical Methods. Springer.

Roberts, G., Gelman, A., Gilks, W., 1997. Weak convergence and optimal scaling of random walk Metropolis algorithms. Annals of Applied Probability 7 (1), 110–120.

Roberts, G., Rosenthal, J., 2007. Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. J. Appl. Probab. 44 (2), 458–475.

Swendsen, R., Wang, J., 1987. Nonuniversal critical dynamics in Monte Carlo simulations. Physical Review Letters 58, 86–88.

Appendix

A. Proof of Theorem 1

Let $\epsilon > 0$, and choose $N \in \mathbb{N}$ as in condition (a). From condition (b) and the coupling argument in the proof of Theorem 1 of Roberts and Rosenthal (2007), the following result holds.

There exists $n^* \in \mathbb{N}$ large enough so that for $K > n^* + N$, there exists a second chain $\{x'^{(n)}, y'^{(n)}\}_{n=K-N}^{K}$, such that $(x'^{(K-N)}, y'^{(K-N)}) = (x^{(K-N)}, y^{(K-N)})$, $(x'^{(n+1)}, y'^{(n+1)}) \sim P_{\theta(K-N)}((x'^{(n)}, y'^{(n)}), dx, dy)$ for $n = K-N, ..., K-1$, and $P(x^{(K)} \neq x'^{(K)}) \leq 2\epsilon$.

Then it follows that

$$||P(x^{(K)} \in dx) - P(x'^{(K)} \in dx)|| \leq 2\epsilon, \tag{6.1}$$

where $P(x^{(K)} \in dx)$ denotes the distribution of $x^{(K)}$, because of $||P(y \in dx) - P(z \in dx)|| \leq P(y \neq z)$.

On the other hand, from the condition (a), for all $A_{\mathcal{X}} \in F_{\mathcal{X}}$, we have

$$\begin{aligned} \epsilon &\geq \left| E[P^N_{\theta(K-N)}((x^{(K-N)}, y^{(K-N)}), A_{\mathcal{X}}) - \pi(A_{\mathcal{X}})] \right| \\ &= \left| P(x'^{(K)} \in A_{\mathcal{X}}) - \pi(A_{\mathcal{X}}) \right|. \end{aligned}$$

That is,

$$||P(x'^{(K)} \in dx) - \pi(dx)|| \leq \epsilon. \tag{6.2}$$

From inequality (6.1) and (6.2), we have

$$||P(x^{(K)} \in dx) - \pi(dx)|| \leq 3\epsilon. \tag{6.3}$$

Since $K \geq n^* + N$ is arbitrary, the algorithm is ergodic.

B. Proof of Theorem 4

We prove the sufficient conditions of convergence in Theorem 2 are satisfied. Firstly, we prove the condition (a') holds.

Let Borel σ-algebra on $\mathbb{R}^p$ be $\mathcal{B}(\mathbb{R}^p)$. For $\boldsymbol{x} \in S^L$, $\boldsymbol{\gamma} \in \Gamma^{pL}$, $\boldsymbol{t} \in \mathcal{T}^L$ and $\boldsymbol{B} = B_1 \times B_2 \times \cdots \times B_L,\ B_l \in \mathcal{B}(S)$, the transition kernel of the PT algorithm is

$$K_{\boldsymbol{\gamma},\boldsymbol{t}}(\boldsymbol{x}, \boldsymbol{B}) = \alpha_r \prod_{l=1}^{L} P_{\gamma_l,t_l}(x_l, B_l) + (1-\alpha_r) \sum_{l=2}^{L} \varsigma_l k_{l,l-1}(\boldsymbol{x}, \boldsymbol{B}), \tag{6.4}$$

where $0 \le \varsigma_l \le 1$, $\sum_{l=2}^{L} \varsigma_l = 1$, $P_{\gamma_l,t_l}(x_l, dx_l)$ and $k_{l,l-1}(\boldsymbol{x}, d\boldsymbol{x}')$ are the Metropolis transition kernel for $\pi_{t_l}(dx_l)$ and the transition kernel of the exchange process of x_l and x_{l-1}, respectively.

By condition (s1), we have $d \equiv \sup_{x\in S, t\in\mathcal{T}} \pi_t(x) < \infty$. By the compactness of S and condition (s2), we have also $\delta \equiv \inf_{x,x'\in S, \gamma\in\Gamma^p} q_\gamma(x, x') > 0$.

For $x \in S$ and $t \in \mathcal{T}$, denote $R_{x,t} = \left\{ y \in S | \frac{\pi_t(y)}{\pi_t(x)} \le 1 \right\}$. For $x_l \in S$, $B_l \in \mathcal{B}(S)$, $t_l \in \mathcal{T}$ and $\gamma_l \in \Gamma$, we have

$$\begin{aligned}
&P_{\gamma_l,t_l}(x_l, B_l) \\
&= \int_{B_l} q_{\gamma_l}(x_l, x_l') \min\left(1, \frac{\pi_{t_l}(x_l')}{\pi_{t_l}(x_l)}\right) dx_l' \\
&+ \mathbf{1}_{\{B_l\}}(x_l) \int_S q_{\gamma_l}(x_l, \tilde{x}_l) \left\{ 1 - \min\left(1, \frac{\pi_{t_l}(\tilde{x}_l)}{\pi_{t_l}(x_l)}\right) \right\} d\tilde{x}_l \\
&\ge \int_{B_l\cap R_{x_l,t_l}} q_{\gamma_l}(x_l, x_l') \frac{\pi_{t_l}(x_l')}{\pi_{t_l}(x_l)} dx_l' + \int_{B_l\cap R^c_{x_l,t_l}} q_{\gamma_l}(x_l, x_l') dx_l' \\
&\ge \frac{\delta}{d} \int_{B_l\cap R_{x_l,t_l}} \pi_{t_l}(x_l') dx_l' + \frac{\delta}{d} \int_{B_l\cap R^c_{x_l,t_l}} \pi_{t_l}(x_l') dx_l' \\
&= \frac{\delta}{d} \pi_{t_l}(B_l).
\end{aligned}$$

From Eq. (6.4), this inequality leads to

$$\begin{aligned}
K_{\boldsymbol{\gamma},\boldsymbol{t}}(\boldsymbol{x}, \boldsymbol{B}) &\ge \alpha_r \prod_{l=1}^{L} P_{\gamma_l,t_l}(x_l, B_l) \\
&\ge \alpha_r \prod_{l=1}^{L} \frac{\delta}{d} \pi_{t_l}(B_l) \\
&= \alpha_r \frac{\delta^L}{d^L} \pi_{\boldsymbol{t}}(\boldsymbol{B}),
\end{aligned} \tag{6.5}$$

where $\pi_{\boldsymbol{t}}(\boldsymbol{B}) = \prod_{l=1}^{L} \pi_{t_l}(B_l)$ is a probability measure on S^L. Since the inequality (6.5) holds for all $\boldsymbol{B} \in \mathcal{B}(S^L)$, the condition (a')(i) follows.

Let $0 < \tau < 1$, $V(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \in S^L$, and $V(\boldsymbol{x}) = 1/\tau$ otherwise, and $b = 1 - \tau$. Then we have

$$(K_{\boldsymbol{\gamma},\boldsymbol{t}}V)(\boldsymbol{x}) \leq \tau V(\boldsymbol{x}) + b\mathbf{1}_{\{S^L\}}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathbb{R}^{pL}. \tag{6.6}$$

This inequality implies that the condition (a')(ii) is satisfied. Also we have $E[V(x^{(0)}, y^{(0)})] \leq 1/\tau < \infty$.

From Eq.s (3.2) and (3.3), it follows that $t_l^{(n+1)} - t_l^{(n)} \to 0$ *almost sure* and $\gamma_l^{(n+1)} - \gamma_l^{(n)} \to 0$ as $n \to \infty$. The minimum inverse temperature decision process changes the value of ς_l only finite times. Thus, the condition (b) in Theorem 1 holds.

The proof is complete.