ORTHOGONAL MCMC ALGORITHMS

Luca Martino , Víctor Elvira , David Luengo , Antonio Artés-Rodríguez , Jukka Corander

Dep. of Mathematics and Statistics, University of Helsinki, 00014 Helsinki (Finland).
Dep. of Signal Theory and Communic., Universidad Carlos III de Madrid, 28911 Leganés (Spain).
Dep. of Circuits and Systems Engineering, Universidad Politécnica de Madrid, 28031 Madrid (Spain).

ABSTRACT

Monte Carlo (MC) methods are widely used in signal processing, machine learning and stochastic optimization. A well-known class of MC methods are Markov Chain Monte Carlo (MCMC) algorithms. In this work, we introduce a novel parallel interacting MCMC scheme, where the parallel chains share information using another MCMC technique working on the entire population of current states. These parallel "vertical" chains are led by random-walk proposals, whereas the "horizontal" MCMC uses a independent proposal, which can be easily adapted by making use of all the generated samples. Numerical results show the advantages of the proposed sampling scheme in terms of mean absolute error, as well as robustness w.r.t. to initial values and parameter choice.

1. INTRODUCTION

Monte Carlo (MC) methods are widely used in signal processing and communications [1, 2, 3]. Markov Chain Monte Carlo (MCMC) methods [4] are well-known Monte Carlo methodologies to draw random samples and compute efficiently integrals involving a complicated multidimensional target probability density function (pdf), $\pi(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^n$. MCMC techniques only need to be able to evaluate the target pdf, but the difficulty of diagnosing and speeding up the convergence has motivated an intense research activity. For instance, several adaptive MCMC methods have been developed in order to adequately fix the parameters of the proposal density, used to suggest candidate samples [3, 5, 4, 6]. Nevertheless, guaranteeing the theoretical convergence is still an issue in most of the cases. In order to explore the state space faster (and specially to deal with high-dimensional applications [7]), several schemes with parallel chains have been recently proposed [2, 6], as well as multiple try and interacting schemes [8], but the problem is still far from being solved.

In this work, we present a novel family of parallel MCMC schemes, the so called orthogonal MCMC (O-MCMC) algorithms, where N different chains are independently run and, at some iterations, they exchange information using another MCMC technique applied on the entire cloud of current states. Assuming that all the MCMC techniques used yield chains converging to the target pdf, the ergodicity is guaranteed: the whole kernel is still valid, since it is a multiplication of ergodic kernels with the same invariant pdf. Our scheme is able to combine both the random-walk and the independent proposal approaches, as both strategies have advantages and drawbacks. On the one hand, random-walk proposal pdfs are often used when there is no information about the target, since this approach turns to be more explorative than using a fixed proposal. On the other hand, a well-chosen independent proposal density usually provides less correlation among the samples in the generated chain. Our method can mix both approaches efficiently: the parallel "vertical" chains (based on random-walk proposals) move around as "lively kids" exploring the state space, whereas the "horizontal" MCMC technique (applied over the population of current states and based on an independent proposal) works as a "loving parent" that redirects "undisciplined kids" towards the "right path" according to the target pdf ("family rules").

Moreover, we also suggest an adaptive black-box strategy: using different fixed variances in each vertical MCMC, and adapting the parameters of the proposal in the horizontal MCMC technique using all the generated samples. The resulting algorithm exhibits both flexibility and robustness w.r.t. initial values and parameter choice. Numerical results show the advantages of the proposed scheme.

2. PROBLEM STATEMENT

In many applications, we are interested in inferring a variable of interest given a set of observations or measurements. Let us consider the variable of interest, $\mathbf{x} \in \mathbb{R}^n$, and let $\mathbf{y} \in \mathbb{R}^d$ be the observed data. The posterior pdf is then

$$p(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{Z(\mathbf{y})} \propto \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}), \qquad (1)$$

This work has been supported by the Spanish government's projects COMONSENS (CSD2008-00010), ALCIT (TEC2012-38800-C03-01), DISSECT (TEC2012-38058-C03-01), COMPREHENSION (TEC2012-38883-C02-01), and by the ERC grant 239784 and AoF grant 251170.

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf and $Z(\mathbf{y})$ is the model evidence or partition function (useful in model selection). In general, $Z(\mathbf{y})$ is unknown, so we consider the corresponding (usually unnormalized) target pdf,

$$\pi(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}). \qquad (2)$$

Our goal is computing efficiently some moment of $\mathbf{x}$, i.e., an integral measure w.r.t. the target pdf,

$$I = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \qquad (3)$$

where $Z = \int_{\mathcal{X}} \pi(\mathbf{x})d\mathbf{x}$.

3. O-MCMC ALGORITHMS: GENERAL OUTLINE

Consider N parallel chains, $\{\mathbf{x}_{i,t}\}_{t=0}^{\infty}$ with $i = 1, ..., N$, generated by different MCMC techniques with *random-walk* proposal pdfs $q_i(\mathbf{x}|\mathbf{x}_{i,t-1})$, i.e., $\mathbf{x} = \mathbf{x}_{i,t-1} + \epsilon$ where ϵ is a random perturbation. Thus, at the t-th iteration we have a population of current states

$$\mathcal{P}_t = \{\mathbf{x}_{1,t}, \mathbf{x}_{2,t}, \ldots, \mathbf{x}_{N,t}\}.$$

At certain selected iterations, t^* such that $t^* = mT_a$ (where T_a is a constant and $m \in \mathbb{N}$), we apply another MCMC technique *over the entire population* $\mathcal{P}_{t^*}$, yielding a new cloud of samples $\mathcal{P}'_{t^*}$. In this way, the different chains share information. This horizontal MCMC method uses an independent proposal pdf $\varphi(\mathbf{x})$. The general O-MCMC approach is summarized below and depicted in Figure 1 for the particular implementation described in the following section.

1. **Initialization:** Set $t = 1$. Choose the N initial conditions, $\mathcal{P}_0 = \{\mathbf{x}_{1,0}, \mathbf{x}_{2,0}, \ldots, \mathbf{x}_{N,0}\}$; the total number of iterations, T; and an integer value $T_a = MT \in \mathbb{N}$ (where $M \in \mathbb{N}$). Let $\mathcal{T}$ be the number of iterations of the horizontal MCMC algorithm.
2. **Vertical step:** For $t = (m-1)T_a + 1, \ldots, mT_a - 1$ (initially, $m = 1$), run an independent MCMC technique for each $\mathbf{x}_{i,t-1} \in \mathcal{P}_{t-1}$, thus obtaining $\mathbf{x}_{i,t}$ and a new population of states $\mathcal{P}_t = \{\mathbf{x}_{1,t}, \mathbf{x}_{2,t}, \ldots, \mathbf{x}_{N,t}\}$.
3. **Horizontal step:** If $t = mT_a$ ($m = 1, 2, \ldots, M$):
   (a) Apply an MCMC technique, taking in account the entire population $\mathcal{P}_t$, using an independent proposal $\varphi(\mathbf{x})$. Starting from $\mathcal{W}^{(0)} = \mathcal{P}_t$, each iteration of this MCMC technique produce a new population $\mathcal{W}^{(\tau)}$ for $\tau = 1, ..., \mathcal{T}$.
   (b) Set $\mathcal{P}_t = \mathcal{W}^{(\mathcal{T})}$.
4. If $t < T$, set $t = t + 1$ and repeat from Step 2. Otherwise, end.

Ergodicity: If each vertical MCMC algorithm produces an ergodic chain with invariant density $\pi(\mathbf{x})$ [4], then the ergodicity is guaranteed: it can be shown that the resulting product of suitable kernels is itself a suitable kernel.

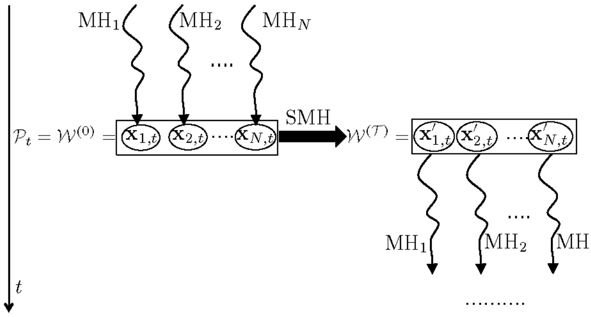

Fig. 1. Sketch of the O-MCMC technique.

4. SPECIFIC O-MCMC IMPLEMENTATION

In this section, we provide a specific O-MCMC implementation: using a standard *Metropolis-Hastings* (MH) algorithm [4] with random walk proposals $q_i(\mathbf{x}|\mathbf{x}_{i,t-1})$ for the vertical chains, and a *Sample Metropolis-Hastings* (SMH) algorithm [9, Chapter 4] with proposal $\varphi(\mathbf{x})$ independent from $\{\mathbf{x}_{i,t-1}\}_{i=1}^{N}$ for the horizontal chain.

4.1. Vertical Chains: Metropolis-Hastings algorithm

For each $i = 1, \ldots, N$ and for a given time step t, one MH update of the i-th chain is obtained as

1. Draw $\mathbf{z} \sim q_i(\mathbf{x}|\mathbf{x}_{i,t-1})$.
2. Set $\mathbf{x}_{i,t} = \mathbf{z}$ with probability

$$\alpha(\mathbf{x}_{i,t-1}, \mathbf{z}) = \min\left[1, \frac{\pi(\mathbf{z})q_i(\mathbf{x}_{i,t-1}|\mathbf{z})}{\pi(\mathbf{x}_{i,t-1})q_i(\mathbf{z}|\mathbf{x}_{i,t-1})}\right].$$

Otherwise, set $\mathbf{x}_{i,t} = \mathbf{x}_{i,t-1}$.

4.2. Horizontal Chain: Sample Metropolis-Hastings

For the sake of simplicity, in this section we do not show the subindex t in the samples $\mathbf{x}_i$. Let us consider a generalized target density,

$$\pi_g(\mathbf{x}_1, \ldots, \mathbf{x}_N) \propto \prod_{i=1}^{N} \pi(\mathbf{x}_i),$$

where each marginal, $\pi(\mathbf{x}_i)$ with $i = 1, ..., m$ and $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^n$, coincides with the true target pdf. The SMH algorithm starts with an initial population $\mathcal{W}^{(0)} = \mathcal{P}_t$, and returns the population of samples

$$\mathcal{W}^{(\tau)} = \{\mathbf{x}_1^{(\tau)}, ..., \mathbf{x}_N^{(\tau)}\}$$

at the τ-th iteration. The underlying idea of SMH is replacing one “bad” sample in the population with a “better” one per iteration, according to a certain suitable probability. The algorithm is designed so that, after a "burn-in" period τ_b, the elements in $\mathcal{W}^{(\tau')}$ ($\tau' > \tau_b$) are distributed according to π_g, i.e., $\mathbf{x}_i^{(\tau')}$ are i.i.d. samples from $\pi(\mathbf{x})$ (since π_g is built using N target pdfs as independent marginals). For $\tau = 1, ..., \mathcal{T}$, the SMH algorithm consists of the following steps:

1. Start with $\mathcal{W}^{(0)} = \mathcal{P}_t$ and $\tau = 0$.
2. Draw $\mathbf{x}_0^{(\tau)} \sim \varphi(\mathbf{x})$, where φ is the proposal density.
3. Choose a "bad" sample $\mathbf{x}_k^{(\tau)}$ in the population, i.e., $k \in \{1, ..., N\}$, according the to the *inverse* of the importance sampling weights: $\frac{\varphi(\mathbf{x}_k^{(\tau)})}{\pi(\mathbf{x}_k^{(\tau)})}$.
4. Accept the new population, $\mathcal{W}^{(\tau+1)} = \{\mathbf{x}_1^{(\tau+1)} = \mathbf{x}_1^{(\tau)}, \ldots, \mathbf{x}_k^{(\tau+1)} = \mathbf{x}_0^{(\tau)}, \ldots, \mathbf{x}_N^{(\tau+1)} = \mathbf{x}_N^{(\tau)}\}$, with probability
$$\alpha(\mathbf{x}_{1:N}^{(\tau)}, \mathbf{x}_0^{(\tau)}) = \frac{\sum_{i=1}^{N} \frac{\varphi(\mathbf{x}_i^{(\tau)})}{\pi(\mathbf{x}_i^{(\tau)})}}{\sum_{i=0}^{N} \frac{\varphi(\mathbf{x}_i^{(\tau)})}{\pi(\mathbf{x}_i^{(\tau)})} - \min\limits_{0 \leq i \leq N} \frac{\varphi(\mathbf{x}_i^{(\tau)})}{\pi(\mathbf{x}_i^{(\tau)})}}.$$
Otherwise, set $\mathcal{W}^{(\tau+1)} = \mathcal{W}^{(\tau)}$.
5. If $\tau < \mathcal{T}$, set $\tau = \tau + 1$ and repeat from Step 2. Otherwise, end.

Let us remark that the difference between $\mathcal{W}^{(\tau)}$ and $\mathcal{W}^{(\tau+1)}$ is at most one sample, and the acceptance probability, $0 \leq \alpha(\mathbf{x}_{1:N}^{(\tau)}, \mathbf{x}_0^{(\tau)}) \leq 1$, depends on the entire population, $\mathbf{x}_i^{(\tau)}$ for $i = 0, \ldots, N$. The ergodicity can be easily proved by using the detailed balance condition and considering the extended target pdf. Note also that the SMH algorithm becomes the standard MH method for $N = 1$. Hence, for $N = 1$ the specific O-MCMC implementation described here consists of applying alternatively two MH kernels with different types of proposals: a random walk proposal, $q_i(\mathbf{x}|\mathbf{x}_{i,t})$, and an independent one, $\varphi(\mathbf{x})$. This a well-known scheme (cf. [4, 9]), which can be seen as a particular case of the O-MCMC family of algorithms. Finally, it is important to remark that the population of proposals is never impoverished by the SMH algorithm, even if a poor choice of $\varphi(\mathbf{x})$ is made. In the worst case, the newly proposed samples are always discarded and computational time is wasted. In the best case, a proposal located in a low probability region can jump close to a mode of the target. Hence, there is a lot to gain and little to lose by placing the horizontal MCMC on top of the vertical chains.

5. BLACK-BOX IMPLEMENTATION

As in any other Monte Carlo technique, the performance of the O-MCMC algorithm depends on the initialization, as well as on the choice of the proposals and their parameters. Fortunately, the sensitivity of O-MCMC schemes w.r.t. these two issues is strongly reduced in comparison to a standard MH algorithm, as illustrated in the simulations. In any case, if some prior information about the target is available, it should be used to choose the initial parameters. However, if no prior information is available, a possible *black-box* implementation of O-MCMC is as follows:

- Choose the initial states, $\mathbf{x}_{i,0}$ with $i = 1, \ldots, N$, spread through the state space, in order to cover as much as possible of the target's domain, $\mathcal{X} \subseteq \mathbb{R}^n$.
- For each proposal, $q_i(\mathbf{x}|\mathbf{x}_{i,t-1})$, choose different scale parameters (e.g., different covariance matrices), incorporating both small and large values to take advantage simultaneously of local (i.e., small scale) and global (i.e., large scale) exploratory behaviours. For instance, a grid of variances could be used in practice.

In order to design an algorithm as robust as possible, we suggest keeping the scale parameters fixed for the vertical MCMC algorithms (i.e., without any adaptation) to avoid a loss of diversity within the set of chosen variances. On the other hand, we propose adapting the variance of the horizontal proposal, $\varphi(\mathbf{x})$, since it is not critical, as discussed at the end of the previous section.

5.1. Adaptation of the horizontal proposal $\varphi(\mathbf{x})$

Following a similar approach to the strategies proposed in [3] and [6], we suggest using (after a training period $T_{train} < T$) *all* the generated samples (i.e., for each $t > T_{train}$ and from all the chains) in order to adapt the location and scale parameters of $\varphi(\mathbf{x})$. For instance, if $\varphi_t(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ we can use the following approach:

- If $t \leq T_{train}$: set $\boldsymbol{\mu}_t = \boldsymbol{\mu}_0$, $\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_0$ (where $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ are the initial choices).
- If $t > T_{train}$: set $\boldsymbol{\mu}_t = \frac{1}{Nt}\sum_{j=1}^{t}\sum_{i=1}^{N} \mathbf{x}_{i,j}$, and $\boldsymbol{\Sigma}_t = \frac{1}{Nt}\sum_{j=1}^{t}\sum_{i=1}^{N}(\mathbf{x}_{i,j} - \boldsymbol{\mu}_t)(\mathbf{x}_{i,j} - \boldsymbol{\mu}_t)^\top$. Namely, use the empirical mean and covariance matrix estimators, which can be computed recursively [3].

6. SIMULATIONS

For the simulations, we consider a bivariate multimodal target pdf, which is itself a mixture of 5 Gaussians, i.e.,

$$\pi(\mathbf{x}) = \frac{1}{5}\sum_{i=1}^{5} \mathcal{N}(\mathbf{x}; \boldsymbol{\nu}_i, \boldsymbol{\Sigma}_i), \quad \mathbf{x} \in \mathbb{R}^2, \tag{4}$$

with means $\boldsymbol{\nu}_1 = [-10, -10]^\top$, $\boldsymbol{\nu}_2 = [0, 16]^\top$, $\boldsymbol{\nu}_3 = [13, 8]^\top$, $\boldsymbol{\nu}_4 = [-9, 7]^\top$, and $\boldsymbol{\nu}_5 = [14, -14]^\top$, with covariance matrices $\boldsymbol{\Sigma}_1 = [2,\ 0.6; 0.6,\ 1]$, $\boldsymbol{\Sigma}_2 = [2,\ -0.4; -0.4,\ 2]$, $\boldsymbol{\Sigma}_3 = [2,\ 0.8; 0.8,\ 2]$, $\boldsymbol{\Sigma}_4 = [3,\ 0; 0,\ 0.5]$, and $\boldsymbol{\Sigma}_5 = [2,\ -0.1; -0.1,\ 2]$.

	O-MCMC (T=2000)						**Parallel chains (T=2000)**			**Parallel chains (T=4000)**		
N	5		100		1000		5	100	1000	5	100	1000
T_a	1	100	1	100	1	100	–	–	–	–	–	–
$\sigma = 2$	0.9734	1.2322	1.1529	1.5363	2.3618	2.4587	4.3753	2.6925	2.6924	4.3477	2.7198	2.6304
$\sigma = 5$	0.9661	1.1778	0.6655	0.7839	1.1433	1.1948	2.9385	1.3408	1.3352	2.6392	1.2450	1.2409
$\sigma = 10$	0.8733	0.9426	0.2597	0.2695	0.0949	0.0943	1.2682	0.2788	0.0952	0.8967	0.2028	0.0641
$\sigma = 70$	1.0730	1.1491	0.4829	0.4813	0.5077	0.5022	1.8784	0.6046	0.5433	1.5275	0.4140	0.3019

Table 1. Mean absolute error in the estimation of the mean of the target (first component), averaged over 1000 runs, for different values of σ and T_a. For O-MCMC, we set $T = 2000$, and $\varphi(\mathbf{x}) = \mathcal{N}(\mathbf{x}; [0, 0]^\top, \lambda^2 \mathbf{I}_2)$ with $\lambda = 10$.

We apply O-MCMC to estimate the mean (true value $[1.6, 1.4]^\top$) of the target using different values for the number of parallel chains $N \in \{5, 100, 1000\}$. Furthermore, we choose deliberately a "bad" initialization to test the robustness of the algorithm and its ability to improve the corresponding trivial parallel MH implementation. Specifically, we set $\boldsymbol{x}_{i,0} \sim \mathcal{U}([-4, 4] \times [-4, 4])$ for $i = 1, \ldots, N$.

We consider $q_i(\mathbf{x}|\mathbf{x}_{i,t-1}) = \mathcal{N}(\mathbf{x}; \mathbf{x}_{i,t-1}, \mathbf{C}_i)$ using the same isotropic covariance matrix, $\mathbf{C}_i = \sigma^2 \mathbf{I}_2$, for every proposal. We test different values of $\sigma \in \{2, 5, 10, 70\}$ to gauge the performance of O-MCMC. As horizontal proposal, we use a Gaussian pdf, $\varphi(\mathbf{x}) = \mathcal{N}(\mathbf{x}; [0, 0]^\top, \lambda^2 \mathbf{I}_2)$ with $\lambda = 10$. We set $T = 2000$ (we use all the generated samples without removing any "burn-in" period), and $T_a \in \{1, 100\}$, i.e., $M = \frac{T}{T_a} \in \{20, 2000\}$. To keep the same computational cost in each experiment, we set $\mathcal{T} = T_a$, i.e., the total number of iterations of SMH is always $T = \mathcal{T} M$. We also consider the case of standard parallel MH chains with $T = 2000$ and $T' = 2T = 4000$ for a fair comparison w.r.t. O-MCMC, in which we use T vertical and T horizontal MCMC iterations.

Table 1 shows the mean absolute error (MAE) in the estimation of the first component of the mean averaged over 1000 independent runs. O-MCMC always outperforms the independent parallel chains (IPCs) for $T = 2000$, showing a much more stable behaviour w.r.t. the parameter choice (σ). Considering $T' = 4000$ for the IPCs, O-MCMC provides better results for small values of σ (i.e., $\sigma = 2$ and $\sigma = 5$) and a reduced number of chains ($N = 5$). For large scale parameters ($\sigma \in \{10, 70\}$) and a large number of chains ($N \in \{100, 1000\}$), the IPCs provide lower values of MAE. The main reason for this is probably the long "burn-in" period of SMH, which increases with N, since it is working in a huge space (the dimension of π_g: $\mathcal{X}^N \subseteq \mathbb{R}^{nN}$). However, O-MCMC still shows a more robust behaviour w.r.t. σ even in this case, implying that a poor choice of σ could easily lead to worse results for the IPCs even by using $T' = 2T$.

7. CONCLUSIONS

We have introduced a novel family of algorithms, so called O-MCMC schemes, that incorporate a horizontal MCMC to share information among a cloud of parallel MCMC chains. Compared to the fully independent parallel chains approach, the novel technique shows a more robust behaviour w.r.t. the parameterization and better performance for a small number of chains and scale parameters. In future works, we plan to consider alternative approaches for the horizontal chain, and test the adaptive black-box strategy suggested in Section 5.

8. REFERENCES

[1] A. Doucet and X. Wang, "Monte Carlo methods for signal processing," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 152–170, Nov. 2005.

[2] J. Ye, A. Wallace, and J. Thompson, "Parallel Markov chain Monte Carlo computation for varying-dimension signal analysis," in *Proc. EUSIPCO 2009*, Glasgow (Scotland), 24–28 Aug. 2009, pp. 2673–2677.

[3] D. Luengo and L. Martino, "Fully adaptive Gaussian mixture Metropolis-Hastings algorithm," in *Proc. ICASSP 2013*, Vancouver (Canada), pp. 6148–6152.

[4] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, 2004.

[5] L. Martino, J. Read, and D. Luengo, "Independent doubly adaptive rejection Metropolis sampling," in *Proc. ICASSP 2014*, Florence (Italy), 4–9 May 2014.

[6] R. Craiu, J. Rosenthal, and C. Yang, "Learn from the neighbor: Parallel-chain and regional adaptive MCMC," *Journal of the American Statistical Association*, vol. 104, no. 448, pp. 1454–1466, 2009.

[7] W. J. Fitzgerald, "Markov chain Monte Carlo methods with applications to signal processing," *Signal Processing*, vol. 81, no. 1, pp. 3–18, January 2001.

[8] R. Casarin, R. V. Craiu, and F. Leisen, "Interacting multiple try algorithms with different proposal distributions," *Statistics and Computing*, vol. 23, no. 2, pp. 185–200, 2013.

[9] F. Liang, C. Liu, and R. Caroll, *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*, Wiley Series in Computational Statistics, England, 2010.