Post-Processing for MCMC

Edwin D. de Jong

Marco A. Wiering

Mădălina M. Drugan

institute of information and computing sciences, utrecht university

technical report UU-CS-2003-021

www.cs.uu.nl

Post-Processing for MCMC

Edwin D. de Jong Marco A. Wiering Mădălina M. Drugan

June 16, 2003

Abstract

Markov Chain Monte Carlo methods (MCMC) can sample from a target distribution and approximate this distribution. MCMC methods employ the a-priori known unnormalized target distribution to decide on the acceptance of new states, but not to approximate the final distribution. This paper investigates how the unnormalized target distribution can be used for approximation purposes. It is shown that this function can be used to obtain an unbiased estimate of the target distribution. Under certain circumstances, this approach can greatly reduce the variance of the resulting estimate.

A useful side-effect of our estimation method is that the sampling method no longer has to follow the target distribution. Based on this observation, a sampling method with a variable degree of exploration is described that still converges to the target function. Thus, exploration can be performed, thereby improving mixing, without discarding sample points. Both methods are demonstrated in experiments.

1 Introduction

Markov Chain Monte Carlo (MCMC) is a large and important class of sampling methods used in areas including statistics, econometrics, physics, and computer science [1, 3, 5]. MCMC provides a general theoretical framework for sampling from a target distribution when this cannot be done with other simpler methods (e.g. exhaustive enumeration), and for estimating the expectation of a function under this distribution. Methods from MCMC assume that the target distribution is known up to a normalization constant. In existing work on MCMC, the a-priori known unnormalized target distribution is used to decide on the acceptance of new states, but it is not used to compute the final distribution. This paper investigates how the unnormalized target distribution can be used for the purpose of estimation, and what effects this has on the efficiency of sampling.

The approach results in the opportunity to perform exploration by using a sampling distribution that is different from the target distribution without discarding samples. Standard methods for MCMC can take a long time to converge. Two popular methods to speed up convergence of MCMC are parallel tempering [2, 3] and simulated tempering [6]. These methods improve the exploration of the state space by a normal MCMC chain through the use of additional MCMC chains that perform more explorative sampling. However, samples from these additional chains must currently be discarded. In the second part of the paper, we generalize our algorithm to faster explore the state space of the target distribution. Unlike the tempering algorithms, our method for post-processing ensures that the MCMC chain converges to the target distribution even when exploration is used during sampling. Thus, our method is able to use the computational effort spent in sampling more efficiently than the tempering methods.

MCMC methods follow the following scheme, see [1] for an introduction. Given a current sample point, MCMC employs a *proposal* distribution to generate a candidate for the next sample point. The target distribution is then used to determine whether the generated point is accepted, or whether the current point will be maintained. This procedure results in a Markov chain. In the limit of an infinite number of samples, the distribution of the states in this chain approximates the target distribution.

2 Proportionate Sampling Algorithm

We will consider continuous distributions, where the target function is a probability density function. The most general and simple method to monitor the behavior of a MCMC chain in a continuous space is through the use of simple models such as frequency histograms [7]. Specifically, the goal here will be to estimate the target distribution by integrating over regions of the state space called *bins*. This yields an estimate in the form of a histogram, where the height of a bin multiplied with its width yields the probability that a state drawn from the distribution lies within the bin; this probability will be called the probability of the bin, $P(\text{bin})$. For a finite amount of samples, an estimate based on counting the number of points in a bin leads to variance in the height of the bins.

There is an apparent potential to reduce the variance of the estimates by basing this estimate on the target-function. An average of the target function for the sample points would lead to incorrect results however, as this average would be biased by the non-uniform sample. We describe a simple weighting method that addresses this problem. The result is a post-processing step that accepts a sample obtained using a MCMC method, and uses the target function values of the sampled points to produce an unbiased estimate of the distribution.

Our aim is to integrate the target function $f(x)$ over a bin to obtain the relative height for a bin, which after normalization yields the true height of the bin. We show that this relative height can be estimated based on the target-function values of the sample by using a weighting scheme that compensates for the sampling distribution.

The distribution over the sampled points is proportional to the target-function: $p(x^{(i)}) \propto f(x^{(i)})$. To compensate for the sampling distribution, a weight of $\frac{1}{f(x^{(i)})}$ can be used for each sample point $x^{(i)}$. This weighted average results in the following estimate for the relative height of a non-empty bin:

$$h_{\text{bin}} = \frac{\sum_{x^{(i)} \in \text{bin}} f(x^{(i)}) \frac{1}{f(x^{(i)})}}{\sum_{x^{(i)} \in \text{bin}} \frac{1}{f(x^{(i)})}} = \frac{n_{\text{bin}}}{\sum_{x^{(i)} \in \text{bin}} \frac{1}{f(x^{(i)})}}$$

where $x^{(i)}$ is the i^{th} sample point in the bin, and n_{bin} is the number of sample points in the bin. Bins for which no samples are obtained receive a zero estimate. This leads to the following estimate for the relative height h_{bin} of a bin:

$$h_{\text{bin}} = \begin{cases} \frac{n_{\text{bin}}}{\sum_{x^{(i)} \in \text{bin}} \frac{1}{f(x^{(i)})}} & \text{for} \quad n_{\text{bin}} > 0 \\ 0 & \text{for} \quad n_{\text{bin}} = 0 \end{cases}$$

The main assumption we make is that within any bin containing mass, the function is bounded away from zero. Then in the limit of an infinite number of samples, the expectation of this estimate multiplied by the width w of the bin converges to the desired value of $\int_{\text{bin}} f(x)dx$, the relative amount of mass represented by the bin:

$$w\ E(h_{\text{bin}}) = \int_{\text{bin}} f(x)dx \tag{1}$$

This result follows from the proof given in appendix A, by considering the special case of $q = 1$ used by e.g. the Metropolis algorithm. If f was a normalized valid density function, eq. (1) equals the probability that a state drawn from the distribution will lie within the bin. If not, the resulting values can be normalized. In the limit of an infinite sample size, all bins will have been visited with probability one. Given furthermore that the estimate of each bin converges to the relative amount of mass represented by the bin, the normalized estimate converges in the limit to the true density function.

The proposed procedure is as follows, see Figure 1. First, we obtain a chain x by sampling with a MCMC algorithm, such as Metropolis-Hastings. Next, we randomly sample a set y of m points from x. Since the sample produced by MCMC methods converges to the target distribution, points in y are independently identically distributed (i.i.d.). This step is necessary for the proof, but may be omitted by choosing $y = x$. Finally, for each bin, we find the points y_{bin} in y that fall within the bin. The height of the bin is determined by taking a weighted average of the function values of these points.

The estimate obtained with the above procedure may yield improvement for two types of inaccuracies that occur for standard MCMC methods:

- Local scale effects: since estimates are based on bin counts, substantial variance in the height of individual bins may occur.

- Macro scale effects: in multi-modal distributions, difference in the time spent in each mode can lead to variance in the fraction of samples obtained for each mode. This phenomenon is known as 'bad mixing'.

2.1 Empirical Results

To demonstrate the potential benefit of the post-processing step described above, MCMC is compared to MCMC with post-processing on two example problems. The first problem consists of a single Gaussian. The sampling method used is the Metropolis algorithm, which employs a symmetric random walk proposal distribution, i.e. $q(x^*|x^{(i)}) = q(x^{(i)}|x^*)$. The standard method has substantial variance for the given number of sample points (5000), exemplifying a local scale effect, see figure 2. The post-processing step strongly reduces this variance.

Distribution-proportionate Algorithm

```
/*Metropolis-Hastings sampling algorithm*/
initialize $x_0$ s.t. $f(x_0) \neq 0$
for $i = 0 : n - 1$,
    Sample $u \sim U(0, 1)$
    Sample $x^* \sim q(x^*|x^{(i)})$
    if $u <$min$(1, \frac{f(x^*)q(x_{(i)}|x^*)}{f(x^{(i)})q(x^*|x^{(i)})}) \land f(x^*) \neq 0$
        $x_{i+1} = x^*$
    else
        $x_{i+1} = x^{(i)}$
end;

/*Post-processing step*/
for $i = 1 : m$,
     Sample $y^{(i)}$ from $x$ with $p(y^{(i)} = x_j) = \frac{1}{|x|}$
end

$\forall$ bin $\in$ bins
    $y_{\text{bin}} = \{y' \in y | y' \in \text{bin}\}$
    if $|y_{\text{bin}}| = 0$
        $h_{\text{bin}} = 0$
    else
        $h_{\text{bin}} = \frac{|y_{\text{bin}}|}{\sum_{y' \in y_{\text{bin}}} \frac{1}{f(y')}}$
end
```

Figure 1: Distribution-proportionate Sampling Algorithm: sample according to $f(x)$, weight according to $\frac{1}{f(x)}$.

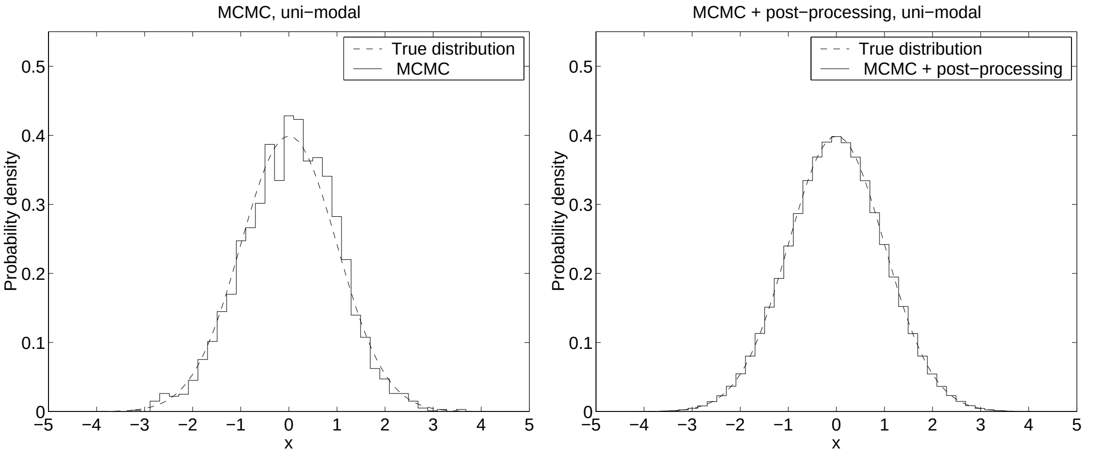

Figure 2: MCMC and MCMC + post-processing on a simple uni-modal distribution.

The second problem consists of two Gaussians. In multi-modal distributions such as these, the amount of sample points spent in each mode may not be proportional to the mass represented by the modes, due to limited sample sizes. This problem is known as bad mixing, and can result in macro-scale inaccuracies. If all modes are visited, standard MCMC plus the post-processing step may reduce this form of variance, as shown in figure 3.

3 Exponentiated Sampling Algorithm

The second problem of the previous section exemplifies the possible effects of bad mixing. When the proposed post-processing step is used to estimate a distribution however, it is no longer necessary to sample according to the target distribution. Thus, we may choose to perform exploration by sampling according to $f^q(x)$, where q is a constant governing the degree to which higher probability states are preferred. The weights required to compensate for the sampling distribution are given by $f^{-q}(x)$. For $q = 0$ for example, this results in random sampling over the states containing mass. This leads to the algorithm shown in Figure 4.

The resulting estimate converges to the optimal values:

$$h_{\text{bin}} = \begin{cases} \frac{\sum_{x^{(i)} \in \text{bin}} f(x^{(i)}) f^{-q}(x^{(i)})}{\sum_{x^{(i)} \in \text{bin}} f^{-q}(x^{(i)})} & \text{for} \quad n_{\text{bin}} > 0 \\ 0 & \text{for} \quad n_{\text{bin}} = 0 \end{cases}$$

$$w\ E(h_{\text{bin}}) = \int_{\text{bin}} f(x) dx$$

A proof of this is given in Appendix A. To demonstrate the potential benefit of exploration, the third test problem consists of two distant Gaussians, connected by a 'tunnel' of low probability density. For sparse, multi-modal problems such as this example, standard MCMC is likely to visit only part of the modes of the distribution.

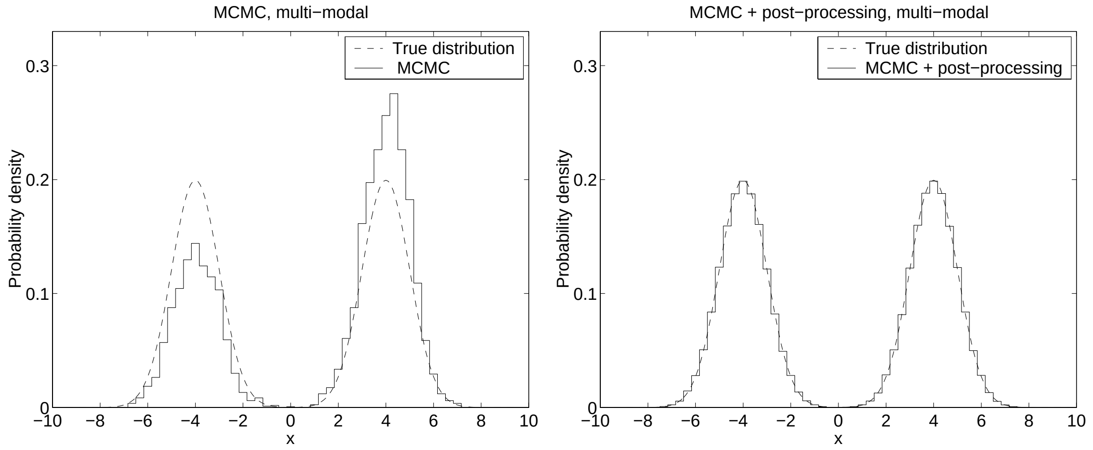

Figure 3: MCMC and MCMC + post-processing on a bi-modal distribution.

By lowering the influence of high probability while still taking the target function into account, exploration can improve the chances of visiting the different modes. The subsequent post-processing step ensures that the resulting estimates still represent the target function.

Figure 5 and 6 show the results for this third test problem. While standard MCMC has only visited one of the modes after 5000 samples, the exponentiated method (with q = 0.1) has substantially explored the space, visiting all states with substantial mass multiple times. Using the proposed post-processing step for estimation, this results in an accurate estimate of the distribution.

4 Advantages and Limitations

The proposed method for estimating the mass represented by a bin is based on values of the target function for sampled points within the bin. This estimate is bounded by the minimum and maximum of value of the density function over the bin. As demonstrated, the estimate can have substantially lower variance than an estimate based on counts. The method can only be accurate however if sampled target function values in a bin are representative of the bin. If the target function is highly variable over the range of a bin, this may not be the case. To ensure representative samples, the bin-size may be chosen small, such that the variability of the target function within a bin is limited. On the other hand, an overly small bin size may result in empty bins, for which no informed estimate can be given, and thus the choice of the bin size is an important factor. Ways to overcome this problem are detecting high-variance bins and then splitting them. It is also possible to use self-organizing networks such as Kohonen maps [4] to partition the input-space. We have to explore these issues further.

Since MCMC often converges slowly for practical purposes, there are techniques to speed-up the MCMC sampler, e.g. parallel tempering and simulated tempering.

Exponentiated Sampling Algorithm

```
/*Metropolis-Hastings sampling algorithm using an exponentiated distribution*/
initialize $x_0$ s.t. $f(x_0) \neq 0$
for $i = 0 : n - 1$,
    Sample $u \sim U(0,1)$
    Sample $x^* \sim q(x^*|x^{(i)})$
    if $u <\min(1, \frac{f^q(x^*)q(x_{(i)}|x^*)}{f^q(x^{(i)})q(x^*|x^{(i)})}) \wedge f(x^*) \neq 0$        $x_{i+1} = x^*$
    else
        $x_{i+1} = x^{(i)}$
end;

/*Post-processing step*/
for $i = 1 : m$,
    Sample $y^{(i)}$ from $x$ with $p(y^{(i)} = x_j) = \frac{1}{|x|}$
end

$\forall$ bin $\in$ bins
    $y_{\text{bin}} = \{y' \in y | y' \in \text{bin}\}$
    if $|y_{\text{bin}}| = 0$
        $h_{\text{bin}} = 0$
    else
        $h_{\text{bin}} = \frac{\sum_{y' \in y_{\text{bin}}} f(y') f^{-q}(y')}{\sum_{y' \in y_{\text{bin}}} f^{-q}(y')}$
end
```

Figure 4: Exponentiated Sampling Algorithm: sample according to $f^q(x)$, weight according to $f^{-q}(x)$.

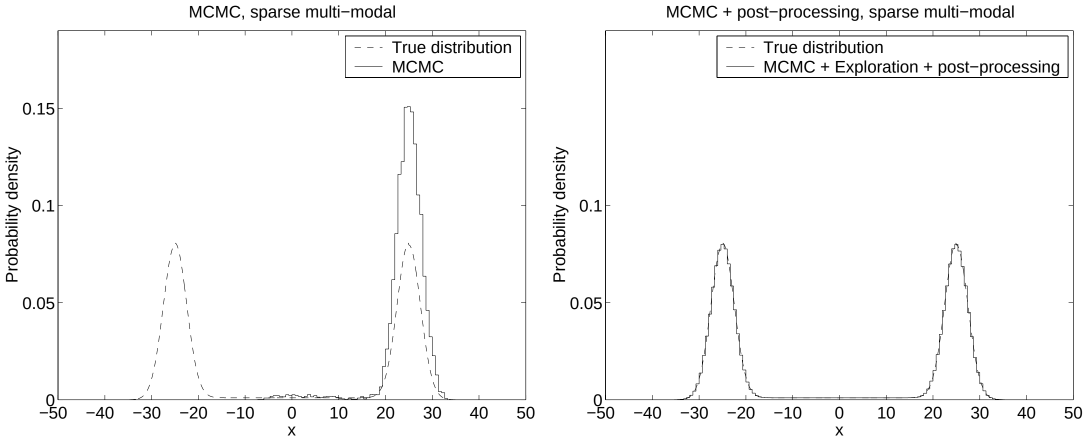

Figure 5: MCMC and Exponentiated MCMC + post-processing on a sparse multi-modal distribution.

Parallel tempering [2, 3] proposes running in parallel N MCMC chains with different stationary distributions one of which is equal to the target distribution. Each such MCMC chain i converges to a target distribution $f(x)^{\frac{1}{Temp[i]}}$ with various values for the temperature $Temp[i]$. Every step, the parallel chains interact by exchanging states between them to improve convergence of the first MCMC chain: better states of a warmer chain (e.g. with higher temperature) could be inserted in a colder chain (e.g. with lower temperature) that is moving slower through the search space than the warmer one. Simulated tempering [6] only uses one long chain which changes its temperature from time to time. Previous states generated with higher temperature can be reinserted in the chain. In parallel tempering, states from a warmer chain are inserted in a colder chain to improve mixing. Note that, normally, only the samples from the (part of) chain which samples from the target distribution are used, the rest of the samples are discarded. By choosing $q[i] = \frac{1}{Temp[i]}$ as the constant from the exponentiated sampling post-processing method however, all samples from all chains can be used to approximate the target distribution.

5 Conclusions

A post-processing method for MCMC has been proposed that provides estimates based on the target function. The procedure can substantially reduce variance on both local and macro scales, as demonstrated in experiments. Since the estimates are not based on counts, it opens up a potential to sample from more explorative distributions while retaining unbiased estimates of the target distribution. One class of such sampling methods, based on exponentiation, was presented. It was seen that mixing can be improved by sampling according to an exponentiated version of the distribution, thereby obtaining substantially better estimates of the distribution. The proposed method is expected to be of benefit in practical applications for which

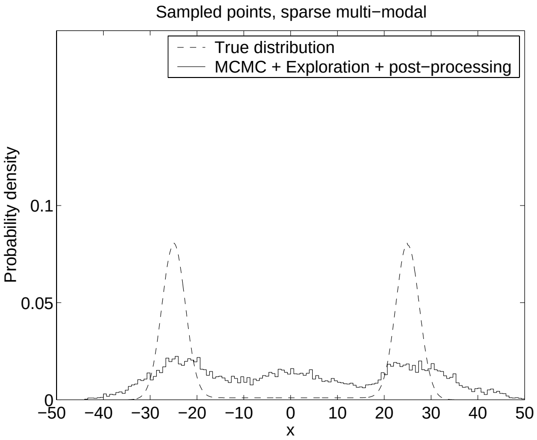

Figure 6: Exponentiated MCMC + post-processing on a sparse multi-modal distribution: frequencies of the sampled points.

the bin-size can be chosen such that the variability of the target function within bins is limited. Furthermore, this work may serve as a starting point for the development of other exploration schemes and corresponding weightings for different MCMC algorithms.

Appendix A

We prove that the estimate produced by the Exponentiated Sampling Algorithm (Figure 4) converges to the optimal values, i.e. for each bin:

$$w\ E(h_{\text{bin}}) = \int_{\text{bin}} f(x)dx$$

where

$$h_{\text{bin}} = \begin{cases} \frac{\sum_{x^{(i)}\in\text{bin}} f(x^{(i)})f^{-q}(x^{(i)})}{\sum_{x^{(i)}\in\text{bin}} f^{-q}(x^{(i)})} & \text{for} \quad n_{\text{bin}} > 0 \\ 0 & \text{for} \quad n_{\text{bin}} = 0 \end{cases}$$

and n_{bin} is the number of sample points in the bin. For bins not containing mass, the algorithm cannot accept any sample points, as the function necessarily equals zero over the domain of such bins. Thus $n_{\text{bin}} = 0$, and the resulting estimate correctly equals zero. For bins containing mass, it was assumed that the function is bounded away from zero over the domain of the bin. For these bins, the probability that one or more sample points are received ($n_{\text{bin}} > 0$) converges to one in the limit of infinite sample size.

$$w\ E(h_{\text{bin}}) = w\ E\frac{\sum_{x^{(i)}\in\text{bin}} f(x^{(i)})f^{-q}(x^{(i)})}{\sum_{x^{(i)}\in\text{bin}} f^{-q}(x^{(i)})} =$$

$$E\frac{\sum_{x^{(i)}\in\text{bin}} f^{1-q}(x^{(i)})w}{\sum_{x^{(i)}\in\text{bin}} f^{-q}(x^{(i)})} =$$

$$\sum_{x^{(i)}\in\text{bin}} E\frac{f^{1-q}(x^{(i)})w}{\sum_{x^{(i)}\in\text{bin}} f^{-q}(x^{(i)})}$$

using the property that these points are i.i.d. (see text), this equals

$$n_{\text{bin}}E\frac{f^{1-q}(x^{(i)})w}{\sum_{x^{(i)}\in\text{bin}} f^{-q}(x^{(i)})} = \frac{n_{\text{bin}}}{n_{\text{bin}}}\frac{Ef^{1-q}(x^{(i)})w}{Ef^{-q}(x^{(i)})} = \frac{Ef^{1-q}(x^{(i)})w}{Ef^{-q}(x^{(i)})} \tag{2}$$

$$Ef^{1-q}(x^{(i)}) = \int_{\text{bin}} \frac{f^q(x)}{\int_X f^q(y)dy}\frac{\int_X f^q(y)dy}{\int_{\text{bin}} f^q(y)dy} f^{1-q}(x)dx =$$

$$\int_{\text{bin}} \frac{f^q(x)}{\int_{\text{bin}} f^q(y)dy} f^{1-q}(x)dx =$$

for $c = \frac{1}{\int_{\text{bin}} f^q(y)dy}$, this is

$$c\int_{\text{bin}} f^q(x)f^{1-q}(x)dx = c\int_{\text{bin}} f(x)dx$$

Furthermore,

$$Ef^{-q}(x^{(i)}) =$$

$$\int_{\text{bin}} \frac{f^q(x)}{\int_X f^q(y)dy}\frac{\int_X f^q(y)dy}{\int_{\text{bin}} f^q(y)dy} f^{-q}(x)dx =$$

$$\int_{\text{bin}} \frac{f^q(x)}{\int_{\text{bin}} f^q(y)dy} f^{-q}(x)dx$$

for c as given above, this equals

$$c\int_{\text{bin}} f^q(x)f^{-q}(x)dx = c\int_{\text{bin}} 1dx = cw$$

Substituting these expectations in eq. 2 yields

$$\frac{Ef^{1-q}(x^{(i)})w}{Ef^{-q}(x^{(i)})} = \frac{c\int_{\text{bin}} f(x)dx \quad w}{cw} = \int_{\text{bin}} f(x)dx$$

References

[1] C. Andrieu, N. De Freitas, A. Doucet, and M.I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

[2] C. J. Geyer. Markov Chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pages 156–163, 1991.

[3] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. Introducting Markov chain Monte Carlo. In W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, editors, *Markov Chain Monte Carlo in practice*. Chapman & Hall, 1996.

[4] T. Kohonen. *Self-Organization and Associative Memory*. Springer, second edition, 1988.

[5] D.J.C. MacKay. *Information theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge, UK, 2002. Draft.

[6] E. Marinari and G. Parisi. Simulated Tempering: A new Monte Carlo scheme. Technical report, Universita di Roma Tor Vergata and Syracuse University, USA, 1992.

[7] K.L. Mengersen, M. Robert, and C. Guihenneuc-Jouyaux. MCMC convergence diagnostics: a review, 1998.