ORIE 7791: Spring 2009
Monte Carlo Methods

Guozhang Wang

May 2, 2009

1 Motivation

1.1 Bayesian Statistics

First, the normalizing denominator in the Bayes's Rule often has a form of integral which is typically intractable. Second, we often want an estimate of the posterior, for example the mean, the standard deviation, etc. And this estimation can be reduced to calculating integrals of the form. Monte Carlo methods can be applied to this integral computation $g(\theta)f(\theta)\lambda(d\theta)$ with high-dimensional parameter theta and unevaluated density function f.

1.2 Statistical Physics

Probability models are used to describe a physical system at equilibrium, whose properties is of interest of the physicists. The property often has the form of integral, which is also typically intractable.

1.3 Theoretical Computer Science

One example is Volume Approximation: given a hypercube A that contains K and a hypercube B that is contained in K, try to approximate the volume of a convex set K.

If only use A to approximate the volume by sampling in $|A|$ and check if it is in K, the estimator is very inefficient if $|K| << |A|$, which is normal in real life applications. Then we need to use B.

Therefore the key point of Monte Carlo method in approximate estimating is 1) algorithm efficiency (the number of sampling is expected to be small, which depend on the speed of converge) and 2) estimate accuracy.

1.4 Deterministic Methods for Integration

To calculate integrals, deterministic methods such as $numericalquadrature$ can be used, but this method needs exponentially increasing computational effort as a function of the dimension N, in order to achieve a fixed error. Therefore some approximation methods (which are still deterministic) such as Laplace Approximation can be used, which set max value of the distribution the same and approximating the distribution function as being proportional to a normal density function, as suggested by the Bayesian Central Limit Theorem. However this method also not generalize to many specific distributions. Therefore calls for random methods, for example, Monte Carlo Integration.

2 Monte Carlo Integration

Following the Law of Large Numbers: for a sequence of i.i.d random variables with expectation μ the average converges almost surely to μ, consider the random variable $g(x)$ which is a function of variable x which has the distribution $f(x)$. If we have i.i.d samples X_i from $f(x)$ then $\frac{1}{n}\sum_{i=1}^{n} g(X_i) \rightarrow \int g(x)f(x)\lambda(dx)asn \rightarrow \infty$. This is called Monte Carlo Integration.

This estimator is an unbiased estimator, and furthermore, using the Central Limit Theorem:

Theorem 2.1 (Central Limit Theorem) *Let $X_1, X_2, X_3, ...X_n$ be a sequence of n independent and identically distributed (i.i.d) random variables each having finite values of expectation μ and variance $\sigma^2 > 0$. As the sample size n increases, the distribution of the sample average of these random variables approaches the normal distribution with a mean μ and variance σ^2/n irrespective of the shape of the original distribution.*

We can get the limiting standard error of the estimator: $\sigma/\sqrt{n} = O(n^{\frac{-1}{2}})$. One note this that this error does not depend on the dimension of x, compared with numerical quadrature in Section 1.4 (actually numerical quadrature in 1 dimension has error $O(\frac{1}{n})$, which is more efficient; but in high dimensions we do much better with Monte Carlo).

Monte Carlo methods not only gives us a good value estimate, it also provides a way to compute the confidential interval estimate. The difficulty of this method is in obtaining the samples from the distribution that is i.i.d. High-dimensional distribution can make this even more challenging. We will show later how Monte Carlo overcomes this problem by simulating these sample processes. In fact, computing the expectation of certain functions over variables (Monte Carlo Integration) relies on sampling from the distribution of the variable (Monte Carlo Sampling).

3 Monte Carlo Sampling: Basic Methods

3.1 Random Number Generation

For a special class of distribution whose inverse $CDFF^{-1}$ can be calculated in closed form, we can generate its samples using Probability Integral Transform.

First of all, we assume that we can generate i.i.d uniform (0, 1) samples (this can be achieved using pseudo-random generators). Simply the discrete-valued random variables can be generated by dividing the space of the possible values according to its distribution. Then we can draw a uniform (0, 1) random sample, and contribute it to value k if it falls in the k^{th} interval of the space. However, if the state space is very large, this is inefficient in dividing the space and checking which intervals the sample falls in.

Hence we can use Probability Integral Transform to generate samples from continuous random variables or discrete variables with large space. In a word, if we can integrate f analytically and invert the result, then its samples can be generated from random variable $F^{-}(U)$ where U is from uniform (0, 1). Using this method, *Exponential, Chi-square, Gamma, Beta, Poisson* distribution can be sampled. Note that since Normal distribution's PDF cannot be integrated, a special transformation method called *Box-Muller* is used to randomly sample r and θ and then convert them to Euclidean coordinates x and y.

3.2 Rejection Sampling

When the inverse $CDFF^{-1}$ cannot be calculated in closed form, transformation methods cannot be used. Therefore more general sampling methods need to be applied. Rejection Sampling is the first Monte Carlo sampling method to solve this problem. The *key* is to find a constant c and a distribution $h(x)$ which we already know how to sample from such that $c \times h(x) \geq f(x)$.

The accept ratio is $\frac{1}{c}$, hence finding c is a very important procedure: if it is too large, then the algorithm will need too many samples and thus inefficient; on the other hand, choosing a small bound c is challenging. Furthermore, for some $h(x)$, there is no such constant that can bound $f(x)$. For example, when $h(x)$ is normal distribution and $f(x)$ is Cauchy distribution.

3.3 Adaptive Rejection Sampling

We often don't know too much about f when we choose h. Therefore the chosen h might not be a good approximation of f, and as a consequences the bound $h(x)/f(x)$ might be very large and the rejection ratio will be large. This would result in the slow convergence problem.

To automatically choose a good function f we can learn from previous rejections to improve the function h. On requirement for Adaptive Rejection Sampling is that the target f must be log-concave. If it is not, a variant Adaptive Rejection Metropolis Sampling can be used.

3.4 Importance Sampling

Instead of rejecting some samples, we can also "weight" the samples in order to compensate for sampling from the wrong density. This method is called *Importance Sampling.* Using Law of Large Numbers, if $w(x) = f(x)/h(x)$ (which means, we assume the support of h includes the support of f) we can get the unbiased estimator:

$$\int g(x)f(x)\lambda(dx) = \int g(x)w(x)h(x)\lambda(dx)$$

$$\approx \frac{1}{n}\sum_{i=1}^{n} g(X_i)w(X_i)$$

One problem for importance sampling is that in practice we cannot calculate the weights $w(X_i)$ since we do not know the normalizing constant c of f, and as a result cannot directly evaluate $f(x)$ to compute $f(x)/h(x)$. However, we can resolve this problem by alternatively represent the weight as $\frac{cw(X_i)}{\sum cw(X_i)}$. Since we can directly evaluate $cf(x)$ we can also evaluate $cf(x)/h(x)$. Also, from LLN we know that:

$$\frac{\sum cg(X_i)w(X_i)}{\sum cw(X_i)} \rightarrow \int\int g(x)w(x)h(x)\lambda(dx)$$

Therefore we can also use $\frac{\sum cg(X_i)w(X_i)}{\sum cw(X_i)}$ as the importance sampling to get an unbiased estimator.
Similar to rejection sampling, another important part for importance sampling is choosing h. Since we can compute the variance of the estimator to be related to $E[g^2(x)\frac{f^2(x)}{h^2(x)}]$, if $f(x)/h(x)$ is unbounded the variance of the estimator will be infinite for many g.

Since minimizing the variance of the estimator corresponds to minimizing $E[g^2(x)\frac{f^2(x)}{h^2(x)}]$, we can get its lower bound using *Jensen's Inequality* (note x^2 is a convex function):

$$E[g^2(x)\frac{f^2(x)}{h^2(x)}] \geq E[g(x)\frac{f(x)}{h(x)}]^2$$

This lower bound is attained when $h(x) = \frac{1}{c}|g(x)|f(x)$.

However, in practice we do not know the normalizing constant $c = \int g(x)f(x)\lambda(dx)$ in order to get $h(x) = \frac{1}{c}|g(x)|f(x)$. Therefore sometimes we have to make a "guess" when choosing h. Under high dimensionality, finding an appropriate h is very difficult since the probability is often concentrated in a small region of the space. Therefore another class of sampling methods, called *Markov Chain Monte Carlo* is proposed to address this high-dimension curse.

4 Markov Chain Monte Carlo Sampling

From the Law of Large Numbers we have the following principle of Monte Carlo:

$\int g(x)\pi(dx) \approx \frac{1}{n}\sum_{i=1}^{n} g(X_i)$, where $X_i \sim^{iid} \pi(dx)$

We can loose the constraint that $X_i \sim^{iid} \pi(dx)$ but X_i approaches π as $i \to \infty$ and still get the above convergence. However, if we loose another constraint to allow X_i to be dependent, the convergence does not hold anymore (a counter example is that all X_i are equal). We will propose *Markov Chain* to address this problem.

4.1 Markov Chains

If the X_i are the iterations of a Markov chain with certain properties, then the convergence stated in the last section does hold. The transition kernel T of a Markov Chain is called:

- *Time-homogeneous* if $PrX_{n+1}|X_0, ..., X_n$, which means it is only dependent on the current state n, not previous states.
- Has *stationary distribution* if $\int \pi(dx)T(X, A) = \pi(A)$.
- *Aperiodic* if there is no cycles that have transition probability $= 1$.
- *Harris Recurrent* if there is no states that is measured 0 under π (those are call transient states). This implies that the Markov chain converges to π for all starting states rather than "π-almost" all starting states.
- *Irreducible* if $\forall dx, A, T(x, A) > 0$. In the general case this means for all x and measurable A s.t. $\pi(A) > 0$: $\exists n \in Ns.t.T^n(x, A) > 0$.

If the transition kernel is time-homogeneous, has the target distribution π as its stationary distribution, and is irreducible and aperiodic, it has π as its *limiting distribution.* Which means $\frac{1}{n}\sum^n i = 1g(X_i) \to \int g(x)\pi(dx)$ "almost surely". Furthermore, if the chain is Harris recurrent then this convergence occurs for all starting states.

Reversibility/detailed balance is a useful property of the transition kernel:

$\pi(dx)T(x, dy) = \pi(dy)T(y, dx)$

since it can guarantee that π is the stationary distribution of T if the above reversibility holds.

4.2 Metropolis-Hastings Sampling

The key idea of M-H sampling is the design of the accepting probability:

$p(x, y) = \min\{1, \frac{f(y)q(y,x)}{f(x)q(x,y)}\}$

so that reversibility (here is simplified as symmetry) holds: $f(x)T(x, y) = f(y)T(y, x) = \min\{f(x)q(x, y), f(x)q(y, x)\}$, and it is not dependent on the unknown normalizing constant of f.

The M-H Independence proposals with $h(x) \geq f(x)/c$, the expected acceptance rate is lower bounded by $1/c$. Although it is better than accept-reject sampling, we have correlations between the samples. So it is not a clear cut which one is better.

4.2.1 Local Proposals

In high dimension space, the choice of local proposal density T centered at the current state is very important to the converge rate of the markov chain. When the target density π is unimodal choice of multivariate normal proposal kernel works well. However, if two parameters are highly correlated in π but not in the proposal density, or if the marginal distribution of one of the parameters in multimodal under π, this proposal density may not guarantee high convergence rate. We give the reasons as follows.

For unimodal graph space, even when the number of nodes in the graph is large, exploring the space using M-H random walk proposal kernel may be done in polynomial number of moves. That is because this local proposal kernel guaranteed to move quickly into thid "crucial region" of high probability of the space (which is the peak region of the unimodal density). This efficient exploring is called "rapid mixing", which means that the number of iterations required for an "accurate" Monte Carlo estimate increases only polynomially in the dimension of the space even if the nodes of the space increases exponentially with the dimension.

However, for multimodal density, the chain would be very difficult to leave from one mode (peak region) to another, but stuck in that mode. On the other hand, the exploring cannot get the "whole picture" of the density until it explores all the crucial regions. Therefore the chain is slowly mixing, which means it requires an exponentially large number of iterations of the chain to get "accurate" Monte Carlo estimates. Actually there is a theorem that formalizes this conclusion, but the result is intuitive.

4.2.2 Scaling Ratio

Another problem is, if our random-walk proposal kernel is multi-variate normal, or multi-variate-t based on the current state, how to choose the correlation matrix $\sum$, given we do not know the correlation of the parameters in π ahead of time? One solution is that we can run a short Markov chain, choose $\sum$ to be the empirical correlation matrix of those samples. Then run a long chain with this proposal covariance. This is called a running period of the chain. We **have** to throw out the samples from the tuning period because the later Markov chain samples depend on them, violating the Markov property.

The reason we do not directly use covariance is due to the scaling factor: we want to normalize our covariance matrix so that we can get an appropriate scaling ratio, and a good acceptance rate as a result (too large scaling ratio causes low acceptance rate, too small scaling ratio has small step size). There is only one theoretical conclusion that, under certain situation (MUN target, MUN proposal, $\sum = \sigma I$, dimension $\to \infty$), best convergence is when acceptance rate = 23.4%. This becomes a rule of thumb that the acceptance rate is flexibility between 20% and 40%.

4.2.3 Burn-In and Thinning

It takes some number of iterations before our chain is close to stationary distribution. We might discard these samples and not use them for Monte Carlo estimation to decrease the variance. This period is called *Burn-In* period.

Because the chain has autocorrelation, we may only keep every k^{th} sample for use in Monte Carlo estimation. Since it is a Markov chain, *thinning* reduces autocorrelation of the samples. However, there has been a theorem stating that using the m samples after thinning always gives use $higher-variance$ estimator than using all the km samples. So the reason of using thinning is mainly memory and computation efficiency concern: you need only to store m samples instead of mk, and you will only compute (probably expensive) $g(x)$ k times also.

4.3 Convergence Diagnosis

Often there are no theoretical bounds derived for the number of iterations required to approximately reach stationarity distribution π. Thus we usually have to empirical methods for assessing convergence: they are absolutely critical, although sometimes misleading.

An analogy: think of some iterative maximization method (e.g. EM), how is convergence verified? And does empirical convergence always mean convergence to the true max? The answer is NO (e.g. when the difference between successive values of x gets small, you stop the EM. But you may at the local maximum, not global maximum).

On the other hand, assess whether the chain has converged is very hard. Proving that the chain has converge is impossible unless we evaluate π at all states in the chain. It is as difficult as our original integration problem. However, proving lack of convergence is often quite easy. One method of these is *Time Series Plots.* One note is that one should create a trace plot for *every* parameter, plus important functions of the parameters, and if one parameter shows lack of convergence, the chain cannot be used for inference.

Furthermore, autocorrelation plots are also helpful for deciding the parameter of the thinning, and the effective independent sample sizes.

4.3.1 Geweke Diagnostic

Take the estimate g_A that uses the first n_A iterations of the chain and the estimate g_B that uses the last n_B iterations of the chain. Their difference divided by the asymptotic standard error of it should have a standard normal distribution if the chain converges before we started saving samples. Calculate z-scores for each parameter and important functions of them. If they are more extreme than expected, the chain may not have converged.

4.3.2 Gelman-Rubin Diagnostic

Recognizing the fact that a chain that is stuck in a single mode of a multimodal distribution appears to have converged, this method chooses to run multiple chains from "overdispersed" starting states. Compare the distributions to which they converge – if they converge to different distributions then the chains may be trapped in different modes. Checking whether they converge to different modes can be done using "shrink factor" (the ratio between within-chain-variance and between-chain-variance) for each single parameter. Factor closer to 1 is better.

One problem with this method is that under high dimension, choosing "overdispersed" starting states is difficult. Some resolutions can be found in [2].

4.4 Gibbs Sampler - Random Scan

Let π defined on $(\theta_1, .., \theta_p) \in X$. Define the M-H chain:

- Sample an index i $\in Z_p$ uniformly
- Update θ_i using the proposal $\pi(\theta_i|\theta_{[-i]})$

The accept probability is actually 1 (inferred from the formula of H-M accept probability).

Systematic-scan Gibbs sampler omit the first step of uniformly randomness but choose $i = 1, 2, ..., p$ iteratively. This is not a M-H chain any more since it is not reversible any more. But at least it can be proved to have the correct stationary distribution $\pi(\theta_1, ..., \theta_p)$. Gibbs sampler can be used when when the full conditional distributions $\pi(\theta_i|\theta_{[-i]})$ are known and can be sampled directly.

4.4.1 Some Advanced Topics

Gibbs sampler can fix the difficulty that M-H had when π has highly correlated parameters by updating a block of parameters each time where these blocks are chosen to contain highly correlated parameters. Updating a group of parameters together can dramatically improve convergence and mixing.

Sometimes the full conditional distribution for one of the parameters is not closed-form, so we can update this parameter via adaptive rejection sampling, treating the other parameters as fixed. That is, for every iteration of the chain we start the adaptation again, and adapt until accepting a sample. On the other hand, it is also valid to update this parameter via a Metropolis-Hastings step, treating all other parameters as fixed. Note the resulting Markov chain is not reversible.

4.5 Convergence of MCMC, Rapid Mixing

We have informally analyze the convergence of MCMC, now we give one theoretical summary that:

$$||\mu_0 T^n - \mu||_2 = ||\mu_0 T^n - \mu T_n||_2$$

$$= ||(\mu_0 - \mu)T_n||_2$$

$$= ||(f - 1)T_n||_2$$

$$\leq ||(f - 1)||_2 ||T_n||_2$$

$$= ||(\mu_0 - \mu)||_2 (1 - Gap(T))^n$$

which means the n-step L-2 distance from stationary is bounded above by the initial distance from stationarity, times $(1 - Gap(T))^n$. Thus in order to guarantee that $||\mu_0 T^n - \mu||_2 < \epsilon$, for smaller ϵ we need a bigger n, for larger initial distance we need a bigger n, for smaller spectral gap we need bigger n. However, in many cases it is hard to get the value of the spectral gap of T, so the convergence diagnosis is still useful.

Next we study the worse case mixing time τ_ϵ, which is the maximum number of iterations required for the chain to be within distance ϵ of stationarity. If τ_ϵ grows at most polynomially as a function of the dimension of the state space, we call it is rapid mixing.

To prove rapid mixing: 1. Show that initial distance $||\mu_0 T^n - \mu||_2||$ is bounded above by M, where M grows at most polynomially; 2. Show that $Gap(T)^{-1}$ grows polynomially.

4.6 A Polynomial-Time Volume Approximation Algorithm

Recall at the beginning of the course we want to estimate the volume of a convex set. Suppose we have an oracle who tells us for any $x \in \mathbb{R}^n$, whether $x \in K$; we also have hypercubes A, B s.t. $B \subset K \subset A$. We want an accurate approximation using polynomial (in n) number of oracle queries. The difficulty is, under high-dimension space, Vol(B) can be exponentially smaller than Vol(K) based on the side length $length(B)$, and Vol(K) can be exponentially smaller than Vol(A) based on the side length $length(A)$, so a naive Monte Carlo estimate is very inefficient.

Instead, we can use *affine transformation* for which we can adjust our estimators to achieve efficiency. Suppose the space dimension is n, we firstly divide the space between B and A into n "circles": $B_i = (length(A)/length(B))^{i/n}$, so $A = B_n, B = B_0$, and $Vol(K) = Vol(B_0)\prod_{i=1}^{n} \frac{Vol(K \cap B_i)}{Vol(K \cap B_{i-1})}$.

If we can get an approximation of $[\frac{Vol(K \cap B_i)}{Vol(K \cap B_{i-1})}]$ to within factors of $1 - \epsilon/n$ and $1 + \epsilon/n$ for each i then we have an approximation of Vol(K) to within factors of $1 - \epsilon$ and $1 + \epsilon$. We can use Metropolis chain to sample approximately uniformly from $Vol(K \cap B_i)$. Since we only need to do n times of MCMC, and each step we only do m times, the algorithm is polynomial.

On the other hand, since $\frac{Vol(K \cap B_i)}{Vol(K \cap B_{i-1})} \in [1/2, 1]$, it is good enough to get an estimate of the ratio to within $[-\epsilon/2n, \epsilon/2n]$.

5 Advanced Techniques

5.1 Swendsen-Wang Algorithm

A single parameter M-H updating schemes such as the Gibbs sampler can be very slow mixing when there is strong dependence between the parameters. An alternative to the usual single parameter updating schemes is the Swendsen-Wang algorithm, in which auxiliary variables are introduced which conditionally remove interactions among parameters. [5] For each pair of the parameters a new variable is introduced, when it is 1 the two parameters are constraint to be equal. So the auxiliary variables define clusters of parameters with the same value.

5.2 Parallel Tempering

Typically a Markov Chain consists of a single stochastic process that accepts/rejects the current states. When the sampling distribution fluctuates with high peaks and deep shallows (high and low energy barriers), traditional MCMC can be dramatically inefficient because the local moves will be very slow in these barrier space, which do not allow the chain to explore all of configuration space..

The parallel tempering algorithm solves this problem by supplementing local Metropolis moves with global "swaps" that update an entire set of configurations. [4] Several MC simulations are run in parallel at a series of different distribution energies (temperatures). The simulation at higher temperatures (in other words, with lower peaks and higher shallows) will be able to explore configuration space more freely. The Parallel Tempering algorithm takes advantage of this by exchanging these higher-temperature configurations at the low temperature of interest. By carefully defining the acceptance probability of the proposed swap, we can preserve the properties of the Metropolis chain.

5.3 validation of MCMC Output for Bayesian Analysis

In the context of Bayesian statistics, there is a simple way to detect errors in the code for the MCMC algorithm. 1) Sample the parameter vector θ from the prior $\pi(\theta)$; 2) Sample data Y from the model $\pi(Y|\theta)$; 3) Obtain a single sample θ' from the posterior distribution $\pi(\theta|Y)$ by simulating the Markov chain; 4) Do this many time. The samples θ' should be marginally distributed according to the prior; 5) For any 1 - D function $g(\theta)$ of the parameters, do a hypothesis test to check whether the samples $g(\theta')$ are distributed according to the known prior distribution of $g(\theta)$.

6 Summary

6.1 The problem

The aims of Monte Carlo methods are to solve one or both of the following problems:

Problem 6.1 *Generate samples from a given probability distribution P(X).*

Problem 6.2 *Estimate expectations of functions under this distribution.*

Note the distribution usually have the following properties: 1) high dimensional, 2) hard to evaluate directly, but can be evaluated within a multiplicative normalizing constant, which we do not know.

If we can solve the first problem, then we can solve the second one by using the random samples to give the estimator as the average of function values. Limit Central Theorem and Law of Large Numbers ensures us that the estimator is unbiased, and the variance of the estimator will decrease as $\frac{\sigma^2}{R}$. This is one of the important properties of Monte Carlo methods: the accuracy of its estimator is independent of the dimensionality of the space sampled.

6.2 Difficulty of High-Dimensional Sampling

The first class of sampling methods is by firstly get random, independent samples from approximation distribution and then adjust to the target distribution. Typical examples are importance sampling and reject sampling. However, they cannot totally solve the problems generated from high dimensional sampling.

The difficulty of these samplings comes from the high dimension, which makes the number of evaluations (in a discretized space) growing exponentially. Furthermore, a high dimensional distribution is often concentrated in a small region of the state space known as its *typical set* (for example, Gaussian) which dominate the value of the expected function value. Therefore if the approximation distribution is not very close to the target sampling, it will take exponentially long time to get enough samples from this typical set regions as the dimension increases (for importance sampling exponentially many samples have to be drawn before we can get one sample from this region, for rejection sampling the rejection ratio is exponentially high). What is worse, it is hard to estimate how reliable the estimator is, since the true variance is difficult to compute from empirical variance. [3]

The reason that this class of sampling fails to overcome this curse of dimensionality is that they use total independent random sampling, which is analogous to random walks. Since random walk is very slowly in exploring the state space, it takes a long time to get to the typical set region of a highly concentrated distribution. Therefore, the goal of the second class of sampling is to eliminate this random walk behavior. [1]

6.3 Markov Chain Monte Carlo

The Markov Chain Monte Carlo (MCMC) draw samples dependent on the last drawn sample instead of independent sampling, in order to "guide" the semi-random walk in the space (for example, Metropolis sampling and Gibbs sampling). The advantage of MCMC is that it is much faster to converge compared with random walked samplings. And further, its performance is not catastrophic dependent on the dimensionality. On the other hand, since the successive samples are correlated with each other, the Markov chain may have to be run for a considerable time to generate samples that are effective independent.

To prove a MCMC will converge with certain proposal density, we need the target distribution an *invariant/stationary distribution* of the chains, and the chain itself be *irreducible and ergoidc*. To prove the first property *reversibility/detailed balance* of the transition probabilities of the Markov chain is often useful.

The length scale (or the step) of the proposal density, which is measured as the variance, determines how fast the estimator is to converge. If the step is large, then it would converge rapidly, but the reject rate would be high; if the step is small, the walk is cautious with low reject rate, but it is slow to converge.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer-Verlag, 2006.

[2] Mary Kathryn Cowles and Bradley P. Carlin. Markov chain Monte Carlo convergence diagnostics: a comparative review. *J. of the American Statistical Association*, 91:883–904, 1996.

[3] D.J.C. MacKay. Introduction to Monte Carlo methods. *Learning in Graphical Models*, 1999.

[4] R. H. Swendsen and J. S. Wang. Replica Monte Carlo simulation of spin glasses. *Phys. Rev. Lett.*, 57:2607–2609, 1986.

[5] R. H. Swendsen and J. S. Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Phys. Rev. Lett.*, 58(2):86–88, 1987.