Bayesian Notes 14 & 15

He Li

Reading notes on PRML Chapter 11, Sampling Methods

November 13, 2018

1 Basic Sampling Algorithms

1.1 Standard distributions

Generate random numbers from simple nonuniform distributions given the fact that we can generate z from uniform distribution over the interval $(0, 1)$. Consider a function $f(\cdot)$ so that $y = f(z)$, Consider the transformation $z = F(y)$ where $F(\cdot) : \mathbb{R} \Rightarrow (0, 1)$ denotes the cdf of y. Then the cdf of z is:

$$P\{z < x\} = P\{F(y) < x\} = P\{y < F^{-1}(x)\} = F(F^{-1}(x)) = x \tag{1.1.1}$$

which is a uniform distribution. Therefore, $y = F^{-1}(z)$.

1.2 Rejection sampling

Sometimes it is hard to sample from a distribution $p(x)$ but we can calculate the value of $q(x)$. We can find a k as small as possible with limitation that $kq(x) \geq p(x)$ for all x. Then we can sample from $q(x)$, and sample $u \sim \text{uniform}[0, kq(x)]$. We reject the sample if $u > p(x)$.

Note that p(x) does not have to be a valid distribution, but a unnormalized one will also work.

The mathematical derivative is as follows:

$$p(\text{accept}) = \int \frac{p(x)}{kq(x)} q(x) \mathrm{d}x = \frac{1}{k} \int p(x)\, \mathrm{d}x = \frac{Z}{k} \tag{1.2.1}$$

and

$$p(x|\text{accept}) = \frac{p(x)}{kq(x)} q(x) \frac{k}{Z} = \frac{p(x)}{Z} \tag{1.2.2}$$

where $Z = \int p(x)\, \mathrm{d}x$ is the normalized constant.

1.3 Adaptive rejection sampling

Sometimes it proves difficult to determine a suitable analytic form for the envelope distribution $q(x)$. An alternative approach is to construct the envelope function on the fly based on measured values of the distribution $p(x)$.

Construction of an envelope function is particularly straightforward for cases in which $p(z)$ is log concave. The function $lnp(z)$and its gradient are evaluated at some initial set of grid points, and the intersections of the resulting tangent lines are used to construct the envelope

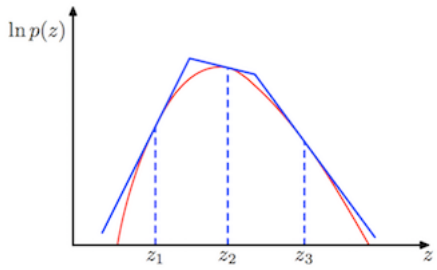

Figure 1: In the case of distributions that are log concave, an envelope function for use in rejection sampling can be constructed using the tangent lines computed at a set of grid points. If a sample point is rejected, it is added to the set of grid points and used to refine the envelope distribution.

function. Next a sample value is drawn from the envelope distribution. See the figure below for an illustration.

A variant of the algorithm exists that avoids the evaluation of derivatives (Gilks, 1992). The adaptive rejection sampling framework can also be extended to distributions that are not log concave, simply by following each rejection sampling step with a Metropolis-Hastings step (to be discussed in Section 11.2.2), giving rise to adaptive rejection Metropolis sampling.

See an example of log-concave rejection sampling from Wanlu Deng's Lecture Notes 8, *Bayesian Methods for Statistical Inference*.

1.4 Importance sampling

The core idea is expressed below:

$$\begin{aligned}\mathbb{E}[f] &= \int f(x)p(x)\mathrm{d}x \\ &= \int f(x)\frac{p(x)}{q(x)}q(x)\mathrm{d}x \\ &\approx \frac{1}{M}\sum_{i=1}^{M}\frac{p(x_i)}{q(x_i)}f(x_i)\end{aligned} \tag{1.4.1}$$

The quantities $\frac{p(x)}{q(x)}$ are known as importance weights, and they correct the bias introduced by sampling from the wrong distribution.

It will often be the case that the distribution p(z) can only be evaluated up to a normalization constant, so that $p(x) = \frac{\tilde{p}(x)}{Z_p}$ where $\tilde{p}(x)$ can be evaluated easily, whereas Z_p is unknown. Similarly, we may wish to use an importance sampling distribution $q(x) = \frac{\tilde{q}(x)}{Z_q}$, which has the same property. We then have

$$\begin{aligned}\mathbb{E}[f] &= \int f(x)p(x)\mathrm{d}x \\ &= \int f(x)\frac{\tilde{p}(x)}{\tilde{q}(x)}\frac{Z_q}{Z_p}q(x)\mathrm{d}x \\ &\approx \frac{1}{M}\frac{Z_q}{Z_p}\sum_{i=1}^{M}r_i f(x_i)\end{aligned} \tag{1.4.2}$$

where $r_i = \frac{\tilde{p}(x_i)}{\tilde{q}(x_i)}$. We can use the same sample set to evaluate the ratio $\frac{Z_p}{Z_q}$ with the result

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q}\int \tilde{p}(x)\mathrm{d}x = \int \frac{\tilde{p}(x)}{\tilde{q}(x)} q(x)\mathrm{d}x \approx \frac{1}{M}\sum_{i=1}^{M} r_i \tag{1.4.3}$$

Therefore, $\mathbb{E}[f] \approx \sum_{i=1}^{M} w_i f(x_i)$ where

$$w_j = \frac{r_j}{M}\frac{Z_q}{Z_p} = \frac{r_j}{\sum_{i=1}^{M} r_i} = \frac{\tilde{p}(x_j)/\tilde{q}(x_j)}{\sum_{i=1}^{M} \tilde{p}(x_i)/\tilde{q}(x_i)} \tag{1.4.4}$$

Note that this estimator is consistent but biased.

1.5 Sampling-importance-resampling

The rejection sampling method discussed in Section 11.1.2 depends in part for its success on the determination of a suitable value for the constant k where sometimes it is impractical. As in the case of rejection sampling, the sampling-importance-resampling (SIR) approach also makes use of a sampling distribution $q(x)$ but avoids having to determine the constant k.

There are two stages to the scheme. In the first stage, M samples $x_1, ..., x_M$ are drawn from $q(x)$. Then in the second stage, weights $w_1, ..., w_M$ are constructed using (6). Finally, a second set of M samples is drawn from the discrete distribution $(x_1, ..., x_M)$ with probabilities given by the weights $(w_1, ..., w_M)$. So our approximation distribution is

$$P(x \le a) = \sum_{i:x_i \le a} w_i = \frac{\sum_{i=1}^{M} \mathrm{I}(x \le a)\tilde{p}(x_i)/q(x_i)}{\sum_{i=1}^{M} \tilde{p}(x_i)/q(x_i)} \tag{1.5.1}$$

The resulting L samples are only approximately distributed according to p(z), but the distribution becomes correct in the limit $L \to \infty$. When $L \to \infty$,

$$\begin{aligned} P(x \le a) &= \frac{\int \mathrm{I}(x \le a)\{\tilde{p}(x)/q(x)\}q(x)\mathrm{d}x}{\int \{\tilde{p}(x)/q(x)\}q(x)\mathrm{d}x} \\ &= \frac{\int \mathrm{I}(x \le a)\tilde{p}(x)\mathrm{d}x}{\int \tilde{p}(x)\mathrm{d}x} \\ &= \int \mathrm{I}(x \le a)\tilde{p}(x)\mathrm{d}x \end{aligned} \tag{1.5.2}$$

which is the cdf of $p(x)$. Again, we see that the normalization of $p(z)$ is not required.

1.6 Sampling and the EM algorithm

Sampling methods can be used to approximate the E step of the EM algorithm for models in which the E step cannot be performed analytically. The Q function can be expressed approximately with samples drawn from the current estimate for the posterior distribution. A particular instance of the Monte Carlo EM algorithm, called stochastic EM, arises if we consider a finite mixture model, and draw just one sample at each E step.

2 Markov Chain Monte Carlo

Rejection sampling and importance sampling suffer from severe limitations particularly in spaces of high dimensionality. However, MCMC allows sampling from a large class of distributions, and which scales well with the dimensionality of the sample space.

Further insight into the nature of Markov chain Monte Carlo algorithms can be gleaned by looking at the properties of a specific example, namely a simple random walk. After τ steps, the random walk has only travelled a distance that on average is proportional to the square root of τ. This square root dependence is typical of random walk behaviour and shows that random walks are very inefficient in exploring the state space. As we shall see, a central goal in designing Markov chain Monte Carlo methods is to avoid random walk behaviour.

2.1 Markov chains

A first-order Markov chain is defined to be a series of random variables $z^{(1)}, ..., z^{(M)}$ such that the following conditional independence property holds for $m \in 1, ..., M1$

$$p(z^{(m+1)} \mid z^{(1)}, ..., z^{(m)}) = p(z^{(m+1)} \mid z^{(m)}) \tag{2.1.1}$$

This of course can be represented as a directed graph in the form of a chain. The *transition probabilities* is $T_m(z^{(m)}, z^{(m+1)}) = p(z^{(m+1)} \mid z^{(m)})$. The marginal probability for a particular variable can be expressed as:

$$p(z^{(m+1)}) = \sum_{z^{(m)}} p(z^{(m+1)} \mid z^{(m)}) p(z^{(m)}) \tag{2.1.2}$$

A Markov chain is called *homogeneous* if the transition probabilities are the same for all m.

A distribution is said to be invariant, or stationary, with respect to a Markov chain if each step in the chain leaves that distribution invariant. Here we can see that invariant means that for each state z, the probability to get out is the same as getting in.

$$p^*(z) = \sum_{z'} T(z', z) p^*(z') \tag{2.1.3}$$

A sufficient (but not necessary) condition for ensuring that the required distribution $p(z)$ is invariant is to choose the transition probabilities to satisfy the property of *detailed balance*, defined by

$$p^*(z) T(z, z') = p^*(z') T(z', z) \tag{2.1.4}$$

It is easily seen that a transition probability that satisfies detailed balance with respect to a particular distribution will leave that distribution invariant, because

$$\sum_{z'} p^*(z') T(z', z) = \sum_{z'} p^*(z) T(z, z') = p^*(z) \sum_{z'} p(z' \mid z) = p^*(z) \tag{2.1.5}$$

We can achieve this if we set up a Markov chain such that the desired distribution is invariant. However, we must also require that for $m \to \infty$, the distribution $p(z(m))$ converges to the required invariant distribution $p^*(z)$, irrespective of the choice of initial distribution $p(z^{(0)})$. This property is called ergodicity, and the invariant distribution is then called the equilibrium distribution.

In practice we often construct the transition probabilities from a set of 'base' transitions $B_1, ..., B_K$. This can be achieved through a mixture distribution of the form

$$T(z', z) = \sum_{k=1}^{K} \alpha_k B_k(z', z) \tag{2.1.6}$$

2.2 The Metropolis-Hastings algorithm

Inspiration of Markov Chain: if we can find a transformation matrix P, such that our target sampling distribution $p(x)$ is its invariant distribution. Given the information above, we just need to satisfy **detailed balanced condition**.

Given a matrix Q, $p(i)Q(i,j) \neq p(j)Q(j,i)$. But we can add acceptance rate:

$$p(i)Q(i,j)\alpha(i,j) = p(j)Q(j,i)\alpha(j,i) \tag{2.2.1}$$

Noted that only the ratio of $\alpha(i,j), \alpha(j,i)$ matters. We can magnify each side until one of them reach 1. Therefore,

$$\alpha(i,j) = min\left(1, \frac{p(j)Q(j,i)}{p(i)Q(i,j)}\right) \tag{2.2.2}$$

Now we get the *Metropolis-Hastings algorithm.*

Metropolis-Hastings algorithm: In particular at step τ of the algorithm, in which the current state is $z(\tau)$, we draw a sample z^* from the distribution $q_k(z \mid z(\tau))$ and then accept it with probability $A_k(z^*, z_\tau)$ where

$$A_k(z^*, z_\tau) = min\left(1, \frac{\tilde{p}(z^*)q_k(z^{(\tau)} \mid z^*)}{\tilde{p}(z^\tau)q_k(z^* \mid z^{(\tau)})}\right) \tag{2.2.3}$$

Here k labels the members of the set of possible transitions being considered.

Note that:

(1) Can use $\tilde{P} \propto P(x)$; normalizer cancels in acceptance ratio

(2) The specific choice of proposal distribution can have a marked effect on the performance of the algorithm. For continuous state spaces, a common choice is a Gaussian centred on the current state, leading to an important trade-off in determining the variance parameter of this distribution. If the variance is small, then the proportion of accepted transitions will be high, but progress through the state space takes the form of a slow random walk leading to long correlation times. However, if the variance parameter is large, then the rejection rate will be high because, in the kind of complex problems we are considering, many of the proposed steps will be to states for which the probability $p(z)$ is low.

(3) Satisfies detailed balance, so that $p(z)$ is an invariant distribution of Markov Chain.

$$p(z)q_k(z \mid z')A_k(z', z) = \min\big(p(z)q_k(z \mid z'), p(z')q_k(z' \mid z)\big) = p(z')q_k(z' \mid z)A_k(z, z') \tag{2.2.4}$$

2.3 Gibbs Sampling

Gibbs Sampling is usually used on high-dimension distribution. And we will see below that Gibbs sampling is a *natural fit for probabilistic graphic model.*

We will use 2-d as an example to verify this algorithm. Consider $(x,y),(x,y')$. We can just verify the correctness of Metropolis-Hastings.

$$\begin{aligned} p(x,y)p(y' \mid x) &= p(x)p(y \mid x)p(y' \mid x) \\ p(x,y')p(y \mid x) &= p(x)p(y' \mid x)p(y \mid x) \end{aligned} \tag{2.3.1}$$

Gibbs Sampling

1. Initialize $\{z_i : i = 1, \ldots, M\}$
2. For $\tau = 1, \ldots, T$:
   - Sample $z_1^{(\tau+1)} \sim p(z_1 | z_2^{(\tau)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - Sample $z_2^{(\tau+1)} \sim p(z_2 | z_1^{(\tau+1)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - $\vdots$
   - Sample $z_j^{(\tau+1)} \sim p(z_j | z_1^{(\tau+1)}, \ldots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - $\vdots$
   - Sample $z_M^{(\tau+1)} \sim p(z_M | z_1^{(\tau+1)}, z_2^{(\tau+1)}, \ldots, z_{M-1}^{(\tau+1)})$.

Figure 2: Gibbs Sampling Algorithm

Therefore, we can use this conditional probability as transition matrix. This is a Metropolis-Hastings with acceptance ratio = 1.

$$p(x, y)p(y' \mid x) = p(x, y')p(y \mid x) \tag{2.3.2}$$

Specically, the lower bound on the number of iterations required to generate an independent sample is $O((L/l)^2)$ where L the marginal variance and l the conditional variance.

As with the Metropolis algorithm, we can gain some insight into the behaviour of Gibbs sampling by investigating its application to a Gaussian distribution. Consider a correlated Gaussian in two variables, as illustrated in Figure 3, having conditional distributions of width l (we can see this as the conditional varinace) and marginal distributions of width L. The typical step size is governed by the conditional distributions and will be of order l. Because the state evolves according to a random walk, the number of steps needed to obtain independent samples from the distribution will be of order $(L/l)^2$. Of course if the Gaussian distribution were uncorrelated, then the Gibbs sampling procedure would be optimally efficient. For this simple problem, we could rotate the coordinate system in order to decorrelate the variables. However, in practical applications it will generally be infeasible to find such transformations.

One approach to reducing random walk behaviour in Gibbs sampling is called *over-relaxation*. In its original form, this applies to problems for which the conditional distributions are Gaussian, which represents a more general class of distributions than the multivariate Gaussian. At each step of the Gibbs sampling algorithm, the conditional distribution for a particular component z_i has some mean μ_i and some variance σ^2. In the over-relaxation framework, the value of z_i is replaced:

$$z_i' = \mu_i + \alpha(z_i - \mu_i) + \sigma_i(1 - \sigma_i^2)^{1/2}\epsilon \tag{2.3.3}$$

where $\epsilon \sim \mathcal{N}(0, 1)$. For $\alpha = 0$, the method is equivalent to standard Gibbs sampling, and for $\alpha < 0$ the step is biased to the opposite side of the mean. This step leaves the desired distribution invariant because if z_i has mean μ_i and variance σ_i^2, then so too does z_i.

Because the basic Gibbs sampling technique considers one variable at a time, there are strong dependencies between successive samples. This is achieved in the blocking Gibbs sampling algorithm by choosing blocks of variables, not necessarily disjoint, and then sampling jointly from the variables in each block in turn, conditioned on the remaining variables.

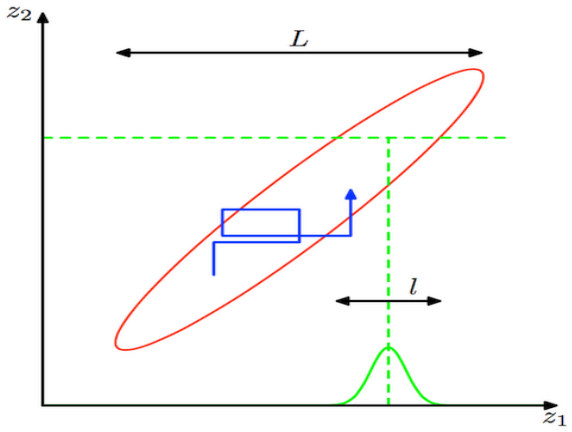

Figure 3: Illustration of Gibbs sampling by alternate updates of two variables whose distribution is a correlated Gaussian. The step size is governed by the standard deviation of the conditional distribution (green curve), and is $O(l)$, leading to slow progress in the direction of elongation of the joint distribution (red ellipse). The number of steps needed to obtain an independent sample from the distribution is $O((L/l)^2)$.

2.4 Slice Sampling

Metropolis: sensitive to step size. If too small, slow decorrelation due to random walk. If too large, inefficiency due to high rejection rate. We instead use *slice sampling*, which require to evaluate the unnormalized distribution $\tilde{p}(z)$.

Consider first the univariate case. Slice sampling involves augmenting z with an additional variable u and then drawing samples from the joint (z, u) space. We shall see another example of this approach when we discuss hybrid Monte Carlo in Section 11.5. The goal is to sample uniformly from the area under the distribution

Consider the univariate case, we add an additional variable u and drawing samples from the joint (z, u) space. The joint distribution is given by

$$\hat{p}(z,u) = \begin{cases} 1/Z_p & \text{if } 0 \le u \le \tilde{p}(z) \\ 0 & \text{otherwise} \end{cases} \tag{2.4.1}$$

where $Z_p = \int \tilde{p}(z)\mathrm{d}z$. The marginal distribution is

$$\int \hat{p}(z,u)\mathrm{d}u = \int_0^{\tilde{p}(z)} \frac{1}{Z_p}\mathrm{d}u = \frac{\tilde{p}(z)}{Z_p} = p(z) \tag{2.4.2}$$

Given the value of z we evaluate p(z) and then sample u uniformly in the range $0 \le u \le \tilde{p}(z)$, which is straightforward. Then we fix u and sample z uniformly from the 'slice' through the distribution defined by $\{z : \tilde{p}(z) > u\}$.

In practice, it can be difficult to sample directly from a slice through the distribution and so instead we define a sampling scheme that leaves the uniform distribution under $\hat{p}(z, u)$ invariant, which can be achieved by ensuring that detailed balance is satisfied. Suppose the current value of z is denoted $z(\tau)$ and that we have obtained a corresponding sample u. The next value of z is obtained by considering a region $z_{min} \le z \le z_{max}$ that contains $z(\tau)$. It is in the choice of this region that the adaptation to the characteristic length scales of the distribution takes place. We want the region to encompass as much of the slice as possible so as to allow large moves in z space while having as little as possible of this region lying outside the slice, because this makes the sampling less efficient. See below a proof for slice sampler.

Proof for invariance of slice sampler

The presentation of the theoretical parts of the slice sampler in the lecture contained some errors. This note gives a more formal (and hopefully correct) presentation.

Assume $x \sim f(x)$ for $x \in \mathcal{X}$.

Defube the extended distribution $f(x,u)$ to be uniform over the set

$$\mathcal{A} = \{(x,u) : x \in \mathcal{X}, 0 \leq u \leq f(x)\}$$

Note first that

$$\int_{(x,u)\in\mathcal{A}} dudx = \int_{x\in\mathcal{X}} \int_0^{f(x)} dudx = \int_{x\in\mathcal{X}} f(x)dx = 1$$

showing that $f(x,u) = 1$ for $(x,u) \in \mathcal{A}$, or $f(x,u) = I(0 \leq u \leq f(x)$ for $x \in \mathcal{X}$.

Further, for $x \in \mathcal{X}$,

$$\int_u f(x,u)du = \int_0^{f(x)} du = f(x)$$

showing that $f(x,u)$ has $f(x)$ as marginal distribution.

The slice sampler is now defined as follows: Iteratively, go through the following steps:

- Sample $u^i \sim \text{uniform}\{u : 0 \leq u \leq f(x^{i-1})\}$
- Sample $x^i \sim \text{uniform}\{x : u^i \leq f(x)\}$

Note first that $\{(x^i, u^i)\}$ is a Markov chain. By defining $\mathcal{A}(u) = \{x : u \leq f(x)\}$ and $|\mathcal{A}(u)|$ the size of $\mathcal{A}(u)$, we have that $K((x,u),(x',u'))$, the kernel for a transition from (x,u) to (x',u') can be written as

$$\begin{aligned} K((x,u),(x',u')) =& K_1((x,u),(x,u'))K_2((x,u'),(x',u')) \\ =& \frac{1}{f(x)} I[0 \leq u' \leq f(x)] \frac{1}{|A(u')|} I[u' \leq f(x')] \end{aligned}$$

We want to show that

$$f(x',u') = \int_{(x,u)\in\mathcal{A}} f(x,u)K((x,u),(x',u'))dudx \qquad (*)$$

which implies that $f(x,u)$ is the invariant distribution for the Markov chain.

Now

$$\begin{aligned}
&\int_{(x,u)\in\mathcal{A}} f(x,u)K((x,u),(x',u'))dudx \\
&\quad = \int_{(x,u)\in\mathcal{A}} \frac{1}{f(x)} I[0 \le u' \le f(x)] \frac{1}{|A(u')|} I[u' \le f(x')] dudx \\
&\quad = \frac{1}{|A(u')|} I[0 \le u' \le f(x')] \int_{x\in\mathcal{X}} \frac{1}{f(x)} I[u' \le f(x)] \int_0^{f(x)} dudx \\
&\quad = \frac{1}{|A(u')|} I[0 \le u' \le f(x')] \int_{x\in\mathcal{X}} I[u' \le f(x)] dx \\
&\quad = \frac{1}{|A(u')|} |A(u')| = I[0 \le u' \le f(x')] = f(x',u')
\end{aligned}$$

showing (*).

One can further show that if $f(x)$ is bounded and the support $\mathcal{X}$ of $f(x)$ is bounded, the slice sampler is aperiodic, irreducible and uniformly ergodic.

2.5 The Hybrid Monte Carlo Algorithm

2.5.1 Dynamical systems

The dynamics that we consider corresponds to the evolution of the state variable $z = \{z_i\}$ under continuous time, which we denote by τ. Classical dynamics is described by Newton's second law of motion in which the acceleration of an object is proportional to the applied force, corresponding to a second-order differential equation over time. We can decompose a second-order equation into two coupled first-order equations by introducing intermediate momentum variables r, corresponding to the rate of change of the state variables z, having components

$$r_i = \frac{\mathrm{d}z_i}{\mathrm{d}\tau} \tag{2.5.1}$$

where the zi can be regarded as position variables in this dynamics perspective. Thus for each position variable there is a corresponding momentum variable, and the joint space of position and momentum variables is called phase space. We can rewrite our probability distribution as

$$p(z) = \frac{1}{Z_p} exp(-E(z)) \tag{2.5.2}$$

where $E(z)$ is interpreted as the potential energy of the system when in state z. The system acceleration is the rate of change of momentum and is given by the applied force, which itself is the negative gradient of the potential energy

$$\frac{\mathrm{d}r_i}{\mathrm{d}\tau} = -\frac{\partial E(z)}{\partial z_i} \tag{2.5.3}$$

We then define the kinetic energy by

$$K(r) = \frac{1}{2}\|r\|^2 = \frac{1}{2}\sum_i r_i^2 \tag{2.5.4}$$

The total energy of Hamilton system is then the sum of its potential and kinetic energies

$$H(z, r) = E(z) + K(r) \tag{2.5.5}$$

We can now express the dynamics of the system in terms of the Hamiltonian equations given by

$$\begin{aligned} \frac{\mathrm{d}z_i}{\mathrm{d}\tau} &= \frac{\partial H}{\partial r_i} \\ \frac{\mathrm{d}r_i}{\mathrm{d}\tau} &= -\frac{\partial H}{\partial z_i} \end{aligned} \tag{2.5.6}$$

During the evolution of this dynamical system, the value of the Hamiltonian H is constant, as is easily seen by differentiation

$$\begin{aligned} \frac{\mathrm{d}H}{\mathrm{d}\tau} &= \sum_i \left\{ \frac{\partial H}{\partial z_i}\frac{\mathrm{d}z_i}{\mathrm{d}\tau} + \frac{\partial H}{\partial r_i}\frac{\mathrm{d}r_i}{\mathrm{d}\tau} \right\} \\ &= \sum_i \left\{ \frac{\partial H}{\partial z_i}\frac{\partial H}{\partial r_i} - \frac{\partial H}{\partial r_i}\frac{\partial H}{\partial z_i} \right\} = 0 \end{aligned} \tag{2.5.7}$$

A second important property of Hamiltonian dynamical systems, known as *Liouville's Theorem*, is that they preserve volume in phase space. In other words, if we consider a region within

the space of variables (z, r), then as this region evolves under the equations of Hamiltonian dynamics, its shape may change but its volume will not. This can be seen by noting that the flow field (rate of change of location in phase space) is given by

$$\mathbb{V} = (\frac{\mathrm{d}z}{\mathrm{d}\tau}, \frac{\mathrm{d}r}{\mathrm{d}\tau}) \tag{2.5.8}$$

and that the divergence of this field vanishes

$$\begin{aligned} \mathrm{div}\mathbb{V} &= \sum_i \left\{ \frac{\partial}{\partial z_i}\frac{\mathrm{d}z_i}{\mathrm{d}\tau} + \frac{\partial}{\partial r_i}\frac{\mathrm{d}r_i}{\mathrm{d}\tau} \right\} \\ &= \sum_i \left\{ -\frac{\partial}{\partial z_i}\frac{\partial H}{\partial r_i} + \frac{\partial}{\partial r_i}\frac{\partial H}{\partial z_i} \right\} = 0 \end{aligned} \tag{2.5.9}$$

Now consider the joint distribution over phase space whose total energy is the Hamiltonian, i.e., the distribution given by

$$p(z, r) = \frac{1}{Z_H}\exp(-H(z, r)) \tag{2.5.10}$$

Using the two results of conservation of volume and conservation of H, it follows that the Hamiltonian dynamics will leave $p(z, r)$ invariant. This can be seen by considering a small region of phase space over which H is approximately constant. If we follow the evolution of the Hamiltonian equations for a finite time, then the volume of this region will remain unchanged as will the value of H in this region, and hence the probability density, which is a function only of H, will also be unchanged. Although H is invariant, the values of z and r will vary, and so by integrating the Hamiltonian dynamics over a finite time duration it becomes possible to make large changes to z in a systematic way that avoids random walk behaviour.

The Markov Chain Steps

(1) Gibbs sample velocity

(2) Simulate Hamiltonian dynamics then flip sign of velocity

   (a) Hamiltonian 'proposal' is deterministic and reversible $q(x', v'; x, v) = q(x, v; x', v')$

   (b) Conservation of energy means $P(x, v) = P(x, v)$

   (c) Metropolis acceptance probability is 1

Evolution under the Hamiltonian dynamics will not, however, sample ergodically from $p(z, r)$ because the value of H is constant. Note that this is because in the phase space, the hamiltonian dynamics is actually a deterministic path given energy constant. In order to arrive at an ergodic sampling scheme, we can introduce additional moves in phase space that change the value of H while also leaving the distribution $p(z, r)$ invariant. The simplest way to achieve this is to replace the value of r with one drawn from its distribution conditioned on z. This can be regarded as a Gibbs sampling step, and hence from Section 11.3 we see that this also leaves the desired distribution invariant. Noting that z and r are independent in the distribution $p(z, r)$, we see that the conditional distribution $p(r \mid z)$ is a Gaussian from which it is straightforward to sample.

In a practical application of this approach, we have to address the problem of performing a numerical integration of the Hamiltonian equations. This will necessarily introduce numerical errors and so we should devise a scheme that minimizes the impact of such errors. One scheme

for achieving this is called the leapfrog discretization and involves alternately updating discrete-time approximations $\hat{z}$ and $\hat{r}$ to the position and momentum variables using

$$\hat{r}_i(\tau + \epsilon/2) = \hat{r}_i(\tau) - \frac{\epsilon}{2}\frac{\partial E}{\partial z_i}(\hat{z}(\tau)) \tag{2.5.11}$$

$$\hat{z}_i(\tau + \epsilon) = \hat{z}_i(\tau) + \epsilon\hat{r}_i(\tau + \epsilon/2) \tag{2.5.12}$$

$$\hat{r}_i(\tau + \epsilon) = \hat{r}_i(\tau + \epsilon/2) - \frac{\epsilon}{2}\frac{\partial E}{\partial z_i}(\hat{z}(\tau + \epsilon)) \tag{2.5.13}$$

Note that the Hamiltonian dynamics method, unlike the basic Metropolis algorithm, is able to make use of information about the gradient of the log probability distribution as well as about the distribution itself. An analogous situation is familiar from the domain of function optimization. In most cases where gradient information is available, it is highly advantageous to make use of it. Informally, this follows from the fact that in a space of dimension D, the additional computational cost of evaluating a gradient compared with evaluating the function itself will typically be a fixed factor independent of D, whereas the D-dimensional gradient vector conveys D pieces of information compared with the one piece of information given by the function itself.

2.5.2 Hybrid Monte Carlo

As we discussed in the previous section, for a nonzero step size , the discretiza- tion of the leapfrog algorithm will introduce errors into the integration of the Hamil- tonian dynamical equations. Hybrid Monte Carlo (Duane et al., 1987; Neal, 1996) combines Hamiltonian dynamics with the Metropolis algorithm and thereby removes any bias associated with the discretization.

Specifically, the algorithm uses a Markov chain consisting of alternate stochastic updates of the momentum variable r and Hamiltonian dynamical updates using the leapfrog algorithm. After each application of the leapfrog algorithm, the resulting candidate state is accepted or rejected according to the Metropolis criterion based on the value of the Hamiltonian H. Thus if (z, r) is the initial state and (z^*, r^*) is the state after the leapfrog integration, then this candidate state is accepted with probability

$$\min\left(1, \exp\{H(z, r) - H(z^*, r^*)\}\right) \tag{2.5.14}$$

If the leapfrog integration were to simulate the Hamiltonian dynamics perfectly, then every such candidate step would automatically be accepted because the value of H would be unchanged. Due to numerical errors, the value of H may sometimes decrease, and we would like the Metropolis criterion to remove any bias due to this effect and ensure that the resulting samples are indeed drawn from the required distribution. In order for this to be the case, we need to ensure that the update equations corresponding to the leapfrog integration satisfy detailed balance.

3 Estimating the Partition Function

Most of the sampling algorithms considered in this chapter require only the functional form of the probability distribution up to a multiplicative constant. The value of the normalization constant Z_E, also known as the partition function, is not needed in order to draw samples from $p(z)$. However, knowledge of the value of Z_E can be useful for Bayesian model comparison since it represents the model evidence (i.e., the probability of the observed data given the model), and so it is of interest to consider how its value might be obtained.

One way to estimate a ratio of partition functions is to use importance sampling from a distribution with energy function.

An alternative approach is therefore to use the samples obtained from a Markov chain to define the importance-sampling distribution.

Resource

- Bishop book: PRML, Chapter 11 Sampling Methods.
- Markov Chain Monte Carlo – Iain Murray's introduction at the 2009 Machine Learning Summer School