CSC 2541: Bayesian Methods for Machine Learning

Radford M. Neal, University of Toronto, 2011

Lecture 3

More Markov Chain Monte Carlo Methods

The Metropolis algorithm isn't the only way to do MCMC. We'll look at two more methods next:

- *Gibbs Sampling* updates one (or some other subset) of variables at a time. It requires that the conditional distributions be tractable.
- *Slice Sampling* is a general framework, but one application also updates one variable at a time, but doesn't require a tractable conditional distribution.

Both of these methods eliminate or reduce the need for careful "tuning" of a proposal distribution, as is needed for Metropolis updates.

Gibbs Sampling

The *Gibbs sampling* algorithm updates each component of the state by sampling from its conditional distribution given other components.

Let the current state be $x = (x_1, \ldots, x_n)$. The next state, $x' = (x'_1, \ldots, x'_n)$, is produced as follows:

- Sample x'_1 from $x_1 \mid x_2, \ldots, x_n$.
- Sample x'_2 from $x_2 \mid x'_1, x_3, \ldots, x_n$.

  ...
- Sample x'_i from $x_i \mid x'_1, \ldots, x'_{i-1}, x_{i+1}, \ldots, x_n$.

  ...
- Sample x'_n from $x_n \mid x'_1, \ldots, x'_{n-1}$.

In many interesting problems we can easily sample from these conditional distributions, even though the joint distribution cannot easily be sampled from.

In many other problems, we can update some of the components this way, and use other updates (eg, Metropolis or slice sampling) for the others.

A Strategy for Gibbs Sampling

1) Write down an expression for the joint posterior density of all parameters, omitting constant factors as convenient.

   This is found as the product of the prior and likelihood.

2) For each parameter in turn, look at this joint density, and eliminate all factors that don't depend on this parameter.

   Try to view the remaining factors as the density for some standard (tractable) distribution, disregarding any constant factors in the density function for the standard distribution.

3) Write a program that samples from each of these standard conditional distributions in turn. The parameters of these standard distributions will in general depend on the variables conditioned on.

This works if the model and prior are *conditionally conjugate* — fixing values for all but one parameter gives a conjugate model and prior for the remaining parameter.

This covers a much larger class of problems than those where the model and prior are conjugate for all parameters at once.

Example: Linear Regression with Gaussian Noise

Gibbs sampling is possible for a simple linear regression model with Gaussian noise, if the regression coefficients have a Gaussian prior distribution and the noise precision (reciprocal of variance) has a Gamma prior.

For example, consider a model of n independent observations, (x_i, y_i), with x_i and y_i both being real:

$$\begin{aligned} y_i \mid x_i, \beta_0, \beta_1, \tau &\sim N(\beta_0 + \beta_1 x_i, 1/\tau) \\ \beta_0 &\sim N(\mu_0, 1/\tau_0) \\ \beta_1 &\sim N(\mu_1, 1/\tau_1) \\ \tau &\sim \text{Gamma}(a, b) \end{aligned}$$

Here, we'll assume that μ_0, μ_1, τ_0, τ_1, a, and b are known constants.

The Gamma(a, b) distribution for τ has density function $\frac{b^a}{\Gamma(a)} \tau^{a-1} \exp(-b\tau)$.

Gibbs Sampling for the Linear Regression Example — β_0

The $N(\mu_0, 1/\tau_0)$ prior for β_0 combines with the likelihood, with β_1 and τ fixed, to give a conditional posterior that is normal. This is like sampling for the mean of a normal distribution given independent data points, $(y_i - \beta_1 x_i)$, that all have variance $1/\tau$:

$$\beta_0 \,|\, x, y, \beta_1, \tau \quad \sim \quad N\Big(\frac{\tau_0\mu_0 + \tau\sum_i(y_i - \beta_1 x_i)}{\tau_0 + n\tau},\ \frac{1}{\tau_0 + n\tau}\Big)$$

We can see this by looking at the relevant part of the log joint posterior density:

$$-(\tau_0/2)(\beta_0 - \mu_0)^2 \ - \ \sum_i (\tau/2)(y_i - \beta_0 - \beta_1 x_i)^2$$

This is a quadratic function of β_0, corresponding to the log of a normal density, with the coefficient of β_0^2 being $-(\tau_0 + n\tau)/2$, giving the variance above. The coefficient of β_0 is $\tau_0\mu_0 + \tau\sum_i(y_i - \beta_1 x_i)$, giving the mean above.

Gibbs Sampling for the Linear Regression Example — β_1

Also, the $N(\mu_1, 1/\tau_1)$ prior for β_1 combines with the likelihood, with β_0 and τ fixed, to give a conditional posterior that is normal. This also is like sampling for the mean of a normal distribution given independent data points, but with data points $(y_i - \beta_0)/x_i$ that have different variances, equal to τx_i.

$$\beta_1 \,|\, x, y, \beta_0, \tau \quad \sim \quad N\left(\frac{\tau_1 \mu_1 + \tau \sum_i x_i (y_i - \beta_0)}{\tau_1 + \tau \sum_i x_i^2},\ \frac{1}{\tau_1 + \tau \sum_i x_i^2}\right)$$

Again, we can see this by looking at the relevant part of the log joint posterior density:

$$-(\tau_1/2)(\beta_1 - \mu_1)^2 \ - \ \sum_i (\tau/2)(y_i - \beta_0 - \beta_1 x_i)^2$$

This is a quadratic function of β_1, corresponding to the log of a normal density, with the coefficient of β_1^2 being $-(\tau_1 + \tau \sum_i x_i^2)/2$, giving the variance above. The coefficient of β_1 is $\tau_1 \mu_1 + \tau \sum_i x_i (y_i - \beta_0)$, giving the mean above.

Gibbs Sampling for the Linear Regression Example — τ

Finally, the Gamma prior for τ combines with the likelihood, with β_0 and β_1 fixed, to give a conditional posterior that is also Gamma:

$$\tau \,|\, x, y, \beta_0, \beta_1 \quad \sim \quad \text{Gamma}\Big(a + n/2,\ b + \sum_i (y_i - \beta_0 - \beta_1 x_i)^2/2\Big)$$

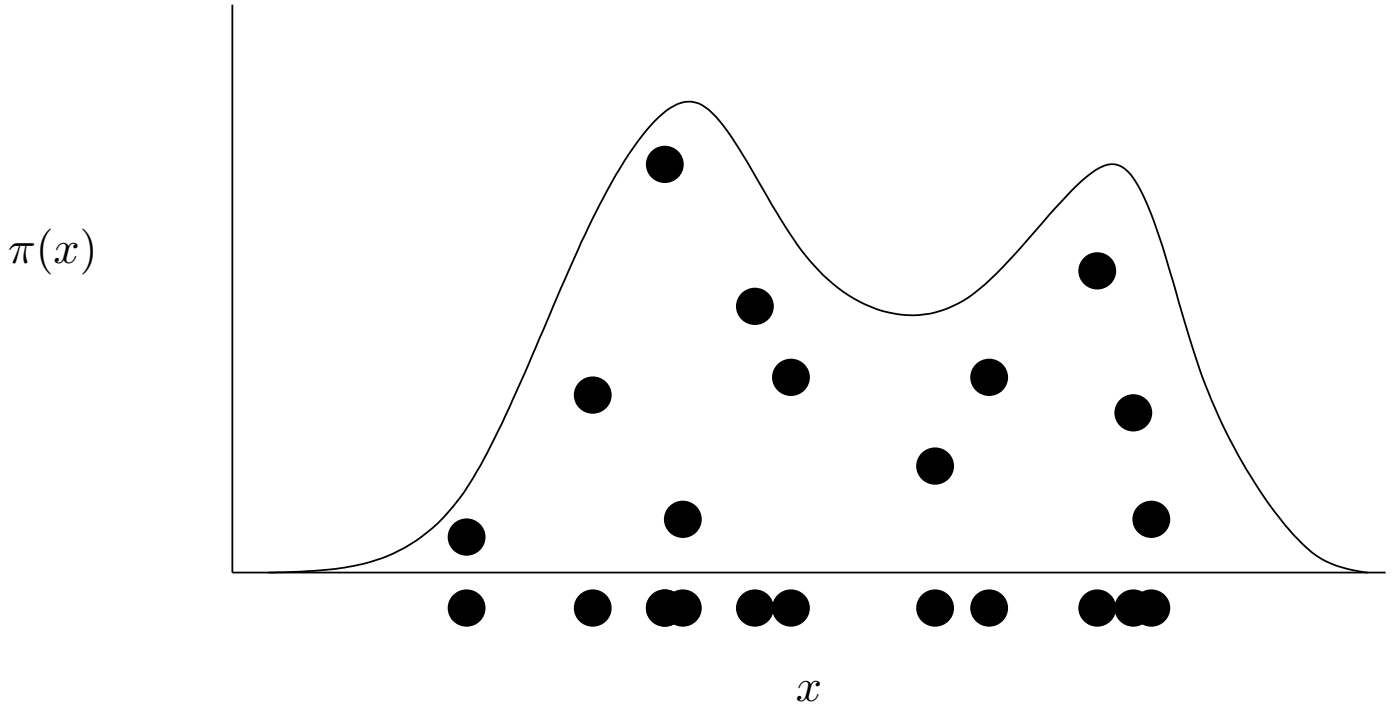
The relevant part of the log joint posterior density is

$$(a-1)\log\tau - b\tau \ + \ (n/2)\log\tau \ - \ \sum_i (\tau/2)(y_i - \beta_0 - \beta_1 x_i)^2$$

This combines into the sum of $\log\tau$ times $a + (n/2) - 1$ and $-\tau$ times $b + \sum_i (y_i - \beta_0 - \beta_1 x_i)^2/2$, giving a Gamma distribution with parameters as above.

The Idea of Slice Sampling (I)

Slice sampling operates by sampling uniformly from under the curve of a density function $\pi(x)$. Ignoring the vertical coordinate then yields a sample of values for x drawn from this density:



However, we won't try to pick points drawn independently from under $\pi(x)$. We will instead define a Markov chain that leaves this uniform distribution invariant.

Note that we can just as easily use any constant multiple of $\pi(x)$, so we have no need to calculate the normalizing constant for π.

The Idea of Slice Sampling (II)

Let's write a point under the curve $\pi(x)$ as (x, y), with y being the height above the x-axis.

The uniform distribution under $\pi(x)$ is left invariant by a Markov chain that alternately

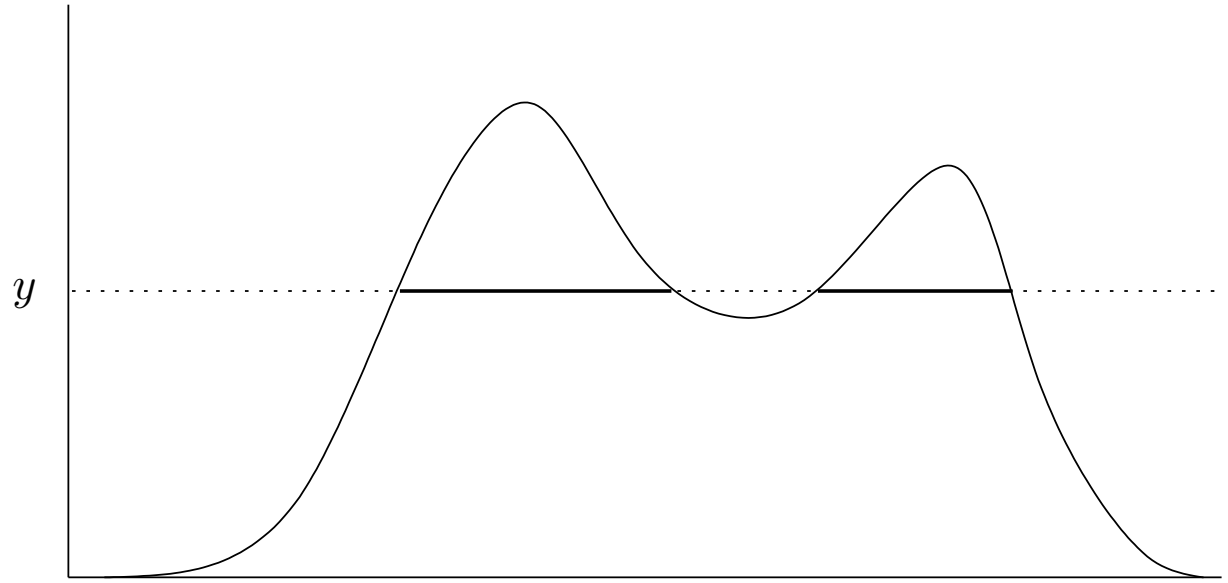
1) Picks a new y uniformly from $(0, \pi(x))$.

2) Performs some update on x alone that leaves invariant the uniform distribution on the "slice" through the density at height y.

If in step (2) we picked a point from the slice independently of the current x, this would just be Gibbs sampling on the (x, y) space.

A Picture of a Slice

For one-dimensional x, the "slice distribution" that we sample from in step (2) will be uniform over the union of some collection of intervals:



If π is unimodal, the slice is a single interval.

Slice Sampling for One Variable

We can use simple slice sampling for one variable at a time as an alternative to Gibbs sampling or single-variable Metropolis updates.

We are probably best off doing just a single slice sampling update for each variable before going on to the next. (But note that this is not quite the same thing as Gibbs sampling.)

On our visit to variable i, we first draw a value y uniformly from the interval $(0,\ \pi(x_i \,|\, x_j : j \neq i))$. Note that π needn't be normalized.

We then update x_i in a way that leaves the slice distribution defined by y invariant.

The big question: How can we efficiently update x_i in such a way?

Slice Sampling for One Variable: Finding an Interval By Stepping Out

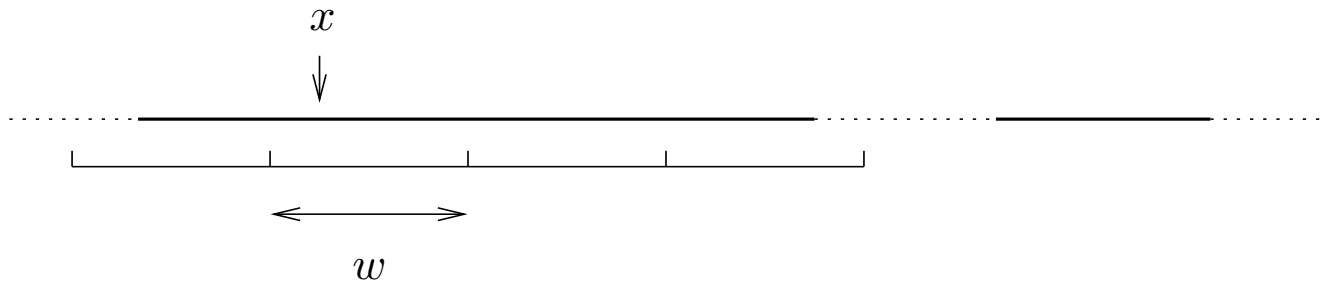
Given a current point x, and a height y, we start by finding an interval (x_0, x_1) around x for which both x_0 and x_1 are outside the slice.

As our first attempt, we try the interval

$$(x - wu,\ x + w(1-u))$$

where u is drawn uniformly from $(0, 1)$.

If both ends of this interval are outside the slice ($\pi(x_0) < y$ and $\pi(x_1) < y$), we stop. Otherwise, we step outwards from the ends, a distance w each step, until the ends are outside:



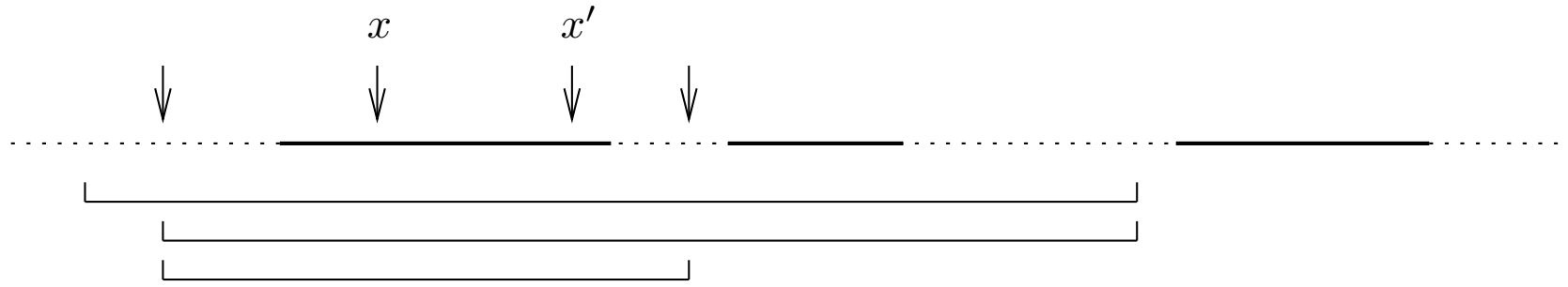
A crucial point: We have the same probability of arriving at this final interval if we start at any point within it that is inside the slice.

Slice Sampling for One Variable: Sampling from the Interval Found

Once we have an interval (x_0, x_1) around the current point, x, with both ends outside the slice, we select a point inside this interval in a way that leaves the slice distribution invariant.

We proceed as follows:

1) Draw x^* uniformly from (x_0, x_1).
2) If x^* is inside the slice, it's our new point, x'. We test this by checking whether $\pi(x^*) \geq y$.
3) Otherwise, we narrow the interval by setting either x_0 or x_1 to x^*, keeping the current point, x, inside the new interval, and then repeat.



Detailed balance holds for $T(x, x')$ when the stepping procedure for finding the interval around x is followed by the above procedure for drawing x'.

Requirements and Characteristics of One-Variable Slice Sampling

- There are no restrictions on distributions we can sample from.
- We need to be able to calculate only $\pi(x)$, up to a constant factor.
- We must decide on a width, w_i, for the initial interval and for any subsequent steps outward. Ideally, w_i will be a few times the standard deviation for x_i.

  If the distribution for x_i is unimodal, we can use data from past iterations to chooseæ good value for w_i, since in this case the distribution of x' does not depend on w_i.
- The x_i' we get is not an independent point from the conditional distribution for x_i. The penalty from this could be large for heavy-tailed distributions.

Accuracy of MCMC estimates

Recall that when estimating $E_\pi[a(x)]$ by the average of $a(x)$ at n *independent* points drawn according to π, the estimate has standard error (the square root of an estimate of the variance in repetitions of the method) of $s_a/\sqrt{n}$, where s_a is the sample standard deviation of $a(x)$.

Under mild conditions (finite variance of $a(x)$ and "geometric" ergodicity of the Markov chain), the standard error for an MCMC estimate based on n *dependent* points from π has the same form, except that n is replaced by an *effective sample size*.

We write this effective sample size for n points as n/τ, where τ is called the *autocorrelation time*.

Note: This all assumes that the early part of the chain (before near convergence) has been discarded, with only later points being used.

Derivation of the Variance of $\overline{a}$

Here is the variance of the sample mean, $\overline{a}$, of n dependent points, $a^{(1)}, \ldots, a^{(n)}$, all from the same distribution, when the true mean is μ, and the autocovariance function for the (stationary) series is $\gamma(k)$:

$$\begin{aligned}
\mathrm{Var}[\overline{a}] &= E[(\overline{a}-\mu)^2] &&= E\Big[\Big(\frac{1}{n}\sum_{t=0}^{n-1}(a^{(t)}-\mu)\Big)^2\Big] \\
&&&= \frac{1}{n^2}\sum_{t,t'=0}^{n-1} E[(a^{(t)}-\mu)(a^{(t')}-\mu)] \\
&&&= \frac{1}{n^2}\sum_{t,t'=0}^{n-1} \gamma(t'-t) \\
&&&= \frac{1}{n}\sum_{-n<s<n} (1-|s|/n)\gamma(s)
\end{aligned}$$

As $n \to \infty$, we get the following:

$$\mathrm{Var}[\overline{a}] \;\to\; \frac{1}{n}\sum_{-n<s<n}\gamma(s) \;=\; \frac{1}{n}\Big[\gamma(0)+2\sum_{s=1}^{\infty}\gamma(s)\Big] \;=\; \frac{\sigma^2}{n}\Big[1+2\sum_{s=1}^{\infty}\rho(s)\Big]$$

$\sigma^2 = \gamma(0)$ is the variance of a, and $\rho(s) = \gamma(s)/\sigma^2$ is the autocorrelation at lag s.

Estimating the Autocorrelation Time

The effective sample size is therefore n/τ with $\tau = 1 + 2\sum_{s=1}^{\infty} \rho(s)$.

How can we estimate this? The standard estimate of $\rho(s)$ for $0 \le s < n$ is $\widehat{\rho}(s) = \widehat{\gamma}(s)/\widehat{\gamma}(0)$, with

$$\widehat{\gamma}(s) = \frac{1}{n}\sum_{i=1}^{n-s}(a^{(i)} - \overline{a})(a^{(i+s)} - \overline{a})$$

The obvious estimate of the autocorrelation time is

$$\widehat{\tau} = 1 + 2\sum_{s=1}^{n-1}\widehat{\rho}(s)$$

but this doesn't work. Even as $n \to \infty$, the estimate doesn't converge to the true value, because more and more noise gets included in the sum of $\widehat{\rho}(s)$.

Instead, one needs to cut off the sum of $\widehat{\rho}(s)$ at some lag above which $\rho(s)$ seems to be nearly zero.

Running Multiple Chains

We can use more than one realization of the Markov chain for MCMC, with different seeds, and different initial state.

We find estimates by averaging values from all chains, after discarding "burn-in" from each chain.

If the same number of iterations are used from each chain, the estimate from C chains is

$$\overline{a}_* \;=\; (1/C)\sum_{c=1}^{C}\overline{a}_c$$

where $\overline{a}_c$ is the sample mean from chain c.

(We could instead use multiple chains of different lengths, but then the analysis is a bit more complicated.)

Advantages and Disadvantages of Multiple Chains

Using $C > 1$ chains has some advantages:

- The needed burn-in period can be assessed more reliably.
- A direct estimate of the standad error can be obtained as the standard deviation of the $\overline{a}_c$ divided by $\sqrt{C}$. (Needs C to be at least ten or so.)
- Autocovariances can be estimated more reliably using $\overline{a}_*$.
- The C chains can be run in parallel if C processors are available.

There are also disadvantages:

- For a given number of total iterations, more are wasted discarding burn-in periods if C burn-in periods are discarded rather than one.
- If the time actually needed to reach equilibrium is very long, we may need to spend all our available time on one chain to reach equilibrium.

For relatively simple problems, using about ten chains may be best. For difficult problems, for which the required burn-in period is long and uncertain, it is possible that one chain may be best.