Doing the Integrals

• For webpage: Some articles
• http://en.wikipedia.org/wiki/Markov_chain_Monte_Carlo
• http://www.stat.berkeley.edu/users/terry/Classes/s260.1998/Week9a/week9a/week9a.html

Until recently, the greatest practical difficulty in Bayesian inference was just doing the integrals—you have to do them eventually in order to compute the marginal likelihoods and marginal distributions

• Some exact results are known, for some useful cases (e.g., normally distributed data) but many realistic problems cannot be handled exactly
• Special cases of conjugate priors could be handled, but were of no help if the real priors did not resemble the available conjugate priors, also only low-dimensional
• Numerical integration can be used, but only for low-dimension problems
• Laplace approximation (approximation of posterior by normal) can be used if there’s lots of data

Ironically, this amounts to using frequentist estimators on the sample to estimate Bayesian quantities!
Doing the Integrals

- There are two main approaches to MCMC, with variations. These are described in many places. A good source of information is Gelman, Carlin, Stern and Rubin, *Bayesian Data Analysis*, Chapters 10-11.
- The two main methods are Gibbs sampling and Metropolis-Hastings sampling; the first is actually a special case of the second.
- Software is available that enables many practical problems to be solved using MCMC. A useful packages is BUGS (Bayesian inference Using Gibbs Sampling) available for the PC and various UNIX flavors (including FreeBSD, but not Linux or PowerPC Macintosh).
- The BUGS website is http://www.mrc-bsu.cam.ac.uk/bugs/welcome.shtml

Doing the Integrals

- Unfortunately, BUGS and other packages are not powerful enough for all problems though they are very flexible and can be used for many problems. So sometimes it is necessary to code the MCMC for oneself. This is not difficult, for the concepts are actually quite straightforward. However, it must be said that getting a program that works efficiently is as much art as science.
- So, we need to understand how MCMC works in practice.

Gibbs Sampling

- Gibbs sampling is the simplest and most easily implemented sampling method for MCMC. However, the problem has to have a particular form in order for it to work.
- The idea is as follows. Consider a problem with two parameters, $\theta_1$ and $\theta_2$. Suppose we can sample from the conditional distributions $p(\theta_1 | \theta_2, D)$ and $p(\theta_2 | \theta_1, D)$ where $D$ is the data (not needed). Then, starting at some initial point $(\theta_1^{(0)}, \theta_2^{(0)})$ in parameter space, generate a random walk, a sequence $(\theta_1^{(k)}, \theta_2^{(k)})$ as follows:
Gibbs Sampling

- For \( k = 1, \ldots, n \) define
  
  \[
  \theta_i^{(k)} \sim p(\theta_i | \theta_{i-1}^{(k-1)}, D), \\
  \theta_j^{(k)} \sim p(\theta_j | \theta_{j-1}^{(k-1)}, D)
  \]

  where ‘\( \sim \)’ means here that we draw the value in question from the indicated distribution.

- The resulting sequence of values is a Markov chain; the values at the \( (k+1) \)st step depend only on the values at the \( k \)th step and are independent of previous values.

- The Markov chain will in general tend to a stationary distribution, and the stationary distribution will be the desired \( p(\theta_1, \theta_2 | D) \).

<table>
<thead>
<tr>
<th>( \theta_1 \setminus \theta_2 )</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

- Example: Suppose the distribution is on a finite state space
  
  \( \theta_1 = \{1, 2\}, \ \theta_2 = \{1, 2, 3\} \)

  with the joint distribution

  \[
  \begin{array}{c|ccc}
  \theta_1 \setminus \theta_2 & 1 & 2 & 3 \\
  \hline
  1 & 0.1 & 0.2 & 0.3 \\
  2 & 0.2 & 0.1 & 0.1 \\
  \end{array}
  \]

- From the joint distribution we get the two full conditional distributions:

  \[
  \begin{array}{c|c|c|c}
  \theta_1 \setminus \theta_2 & 1 & 2 & 3 \\
  \hline
  1 \ | \theta_2 & 0.333 & 0.667 & 0.750 \\
  2 \ | \theta_2 & 0.667 & 0.333 & 0.250 \\
  \end{array}
  \]

  \[
  \begin{array}{c|c|c|c}
  \theta_1 \setminus \theta_2 & 1 & 2 & 3 \\
  \hline
  1 \ | \theta_2 & 0.166 & 0.333 & 0.500 \\
  2 \ | \theta_2 & 0.500 & 0.250 & 0.250 \\
  \end{array}
  \]

  (One would never use the Gibbs sampler in a case like this; where the joint distribution is so tractable! This is just for illustration.)
Gibbs Sampling

- We start anywhere, for example, start at the point (1,1)

<table>
<thead>
<tr>
<th>$\theta_1</th>
<th>\theta_2$</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\theta_2$</td>
<td>0.333</td>
<td>0.667</td>
<td>0.750</td>
</tr>
<tr>
<td>2</td>
<td>$\theta_2$</td>
<td>0.667</td>
<td>0.333</td>
<td>0.250</td>
</tr>
</tbody>
</table>

$\theta_1 \backslash \theta_2$ | $\theta_2$ | 1 | 2 | 3 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\theta_1$</td>
<td>0.166</td>
<td>0.333</td>
<td>0.500</td>
</tr>
<tr>
<td>2</td>
<td>$\theta_1$</td>
<td>0.500</td>
<td>0.250</td>
<td>0.250</td>
</tr>
</tbody>
</table>

- We will now display an R program that does this

```r
# The first fragment of the program does a single Gibbs sampling round, first picking alh and then bla from the conditional distribution (joint/marginal)
sample1 = function(a,b) {
  a = sample(c(1,2),1,prob=joint,1)
  b = sample(c(1,2,3),1,prob=joint[a,])
  invisible(list(a=a,b=b))
}
```

```r
# The second fragment calls sample1() repeatedly and summarizes the results
samplen = function(n=1000) {
  a = 1
  b = 1
  aa = rep(NaN,n)
  bb = rep(NaN,n)
  myjoint = matrix(0,nrow=2,ncol=3)
  # continued on next chart

  for(i in 1:n) {
    z = sample1(a,b)
    aa[i] = a = z$a
    bb[i] = b = z$b
    myjoint[a,b] = myjoint[a,b]+1
  }

  print(myjoint/sum(myjoint))
  print(joint)
  invisible(list(aa=aa,bb=bb))
```

Gibbs Sampling

- The second fragment calls `sample1()` repeatedly and summarizes the results

```r
#...continued from previous chart
```

```r
for(i in 1:n) {
  z = sample1(a,b)
  aa[i] = a = z$a
  bb[i] = b = z$b
  myjoint[a,b] = myjoint[a,b]+1
}
print(myjoint/sum(myjoint))
print(joint)
invisible(list(aa=aa,bb=bb))
```
Gibbs Sampling

- We have to create the array with the joint distribution
  \[
  \text{joint} = c(.1,.2,.2,.1,.3,.1) \\
  \text{dim(joint)} = c(2,3)
  \]
  ...and calculate and inspect the results
  \[
  z = \text{samplen}(1000) \\
  \text{hist(z$aa)} \ #a \ \text{marginal} \\
  \text{hist(z$bb)} \ #b \ \text{marginal}
  \]
- Note that R stores arrays in "column-major" order (like FORTRAN), not "row-major" order (like C). This means that the first index varies most rapidly in memory, e.g.,
  \[
  c[1,1], c[2,1], c[1,2], c[2,2], ... \\
  \]

Bayesian Inference 10/21/08 17

Gibbs Sampling

- The first example was much too simple (for one thing, we already knew the explicit joint and marginal distributions, so we didn’t learn anything new!)
- A more interesting example is to look at normally distributed data, where the data are measurements of some quantity. This is just about the simplest practical problem we could find.
  - This problem has an exact solution for the priors we will use; but the point is to see how we can get our results using MCMC.

Bayesian Inference 10/21/08 18

Gibbs Sampling

- Suppose we have normally distributed observations \(X_i\), i=1,...,N, with mean \(\mu\), and variance \(\sigma^2\). The likelihood is
  \[
p(X|\mu,\sigma^2) \propto \sigma^{-N}\exp(-\sum(X_i-\mu)^2/2\sigma^2)\]
- Assume a flat (uniform) prior for \(\mu\) and a “Jeffreys” prior \(1/\sigma^2\) for \(\sigma^2\). The posterior is proportional to prior times likelihood:
  \[
p(\mu, \sigma^2|X) \propto \sigma^{-(N+2)}\exp(-\sum(X_i-\mu)^2/2\sigma^2)\]
  (The Jeffreys prior is \(d\sigma/\sigma \propto d\sigma^2/\sigma^2\); it is commonly used as a prior for scale variables. For technical reasons having to do with the distributions available in R it’s best to think about sampling \(\sigma^2\) instead of \(\sigma\) so we use \(d\sigma^2/\sigma^2\))

Bayesian Inference 10/21/08 19

Gibbs Sampling

- The posterior distribution can be simplified using the trick of "completing the square"
  \[
p(\mu, \sigma^2 | X) \propto \sigma^{-(N+2)} \exp\left(-\frac{\sum(X_i - \bar{x} + \bar{x} - \mu)^2}{2\sigma^2}\right)
  \]
  \[
  = \sigma^{-(N+2)} \exp\left(-\frac{\sum(X_i - \bar{x})^2}{2\sigma^2}\right) \exp\left(-\frac{\sum(\bar{x} - \mu)^2}{2\sigma^2}\right)
  \]
  \[
  = \sigma^{-(N+2)} \exp\left(-\frac{S_{xx}}{2\sigma^2}\right) \exp\left(-\frac{N(\bar{x} - \mu)^2}{2\sigma^2}\right)
  \]
  \[
  p(\mu | \sigma^2, X) = p(\mu, \sigma^2 | X)/p(\sigma^2 | X)
  \]
  \(\text{Depends only on } \sigma^2\)
Gibbs Sampling

• When sampling $\mu$, $\sigma$ will be fixed, so we can ignore factors dependent only on $\sigma$. Let $\bar{x}$ be the sample mean as we defined it before. Then the conditional distribution of $\mu$ can be rewritten, apart from constant factors

$$p(\mu | \sigma^2, X) \propto \exp \left( -\frac{N(\bar{x} - \mu)^2}{2\sigma^2} \right)$$

• This is readily seen to be normal with mean $\bar{x}$ and variance $\sigma^2/N$. So that’s the distribution we need to use for sampling $\mu$ in each Gibbs step.

Gibbs Sampling

• The conditional distribution for $\chi^2 = \sum(X - \mu)^2/\sigma^2$ is even easier, going back to the original posterior (3 charts back):

$$p(\chi^2 | \mu, X) \propto (\chi^2)^{N/2+1} \exp(-\chi^2/2)$$

$$\propto (\chi^2)^{N/2-1} \exp(-\chi^2/2)$$

$$\sigma^2 \propto \frac{1}{\chi^2} \text{ so } \frac{d\sigma^2}{d\chi^2} \propto \frac{1}{\chi^2}$$

• This is a standard chi-square distribution on $N$ degrees of freedom, and R has a function for drawing samples from that distribution. We can then get a value of $\sigma^2$ by dividing $\sum(X - \mu)^2$ by the value of $\chi^2$ that we sampled. That allows us to sample $\sigma^2$ at each Gibbs step.

Gibbs Sampling

• Calculation: This code fragment performs one sampling step:

```r
sample1 = function(X,mu,sg) {
  N = length(X)
  mu = rnorm(1,mean(X),sg/sqrt(N))
  sg = sqrt(sum((X-mu)^2)/rchisq(1,N))
  list(mu=mu,sg=sg)
}
```

Gibbs Sampling

• This code fragment samples $n$ times by calling `sample1()` repeatedly. It also initializes the Markov chain:

```r
samplen = function(X,n=1000,mu=0,sg=1) {
  mus = rep(NaN,n)
  sgs = rep(NaN,n)
  for(i in 1:n) {
    z = sample1(X,mu,sg)
    mus[i] = mu = z$mu
    sgs[i] = sg = z$sg
  }
  results(mus,sgs) # display results
  invisible(list(mus=mus,sgs=sgs))
}
```
Gibbs Sampling

- We generate a data set
  \[ X = rnorm(15, 3, 5) \] 
  #mean 3, sd 5

  ...and sample 1000 times and then look at the marginal distributions and other posterior statistics, for example in the function `results()`:
  \[
  z = sample(x, 1000)
  \]
  `hist(z$mus, 50)` #mu marginal
  `hist(z$sds, 50)` #sd marginal
  `mean(z$mus)` #posterior mean
  `plot(z$mus)` #look at sequence
  `mean(z$sds)` #posterior mean of sd
  `plot(z$sds)` #look at sequence

Markov Chains

- The matrix \[ M_{ij} = P(X_{n+1}=j | X_n=i) \] of a stationary Markov chain is called the Markov matrix or transition matrix. The entries satisfy \( 0 \leq M_{ij} \leq 1 \) and \( \Sigma_{j} M_{ij} = 1 \). Also, \( [M^n]_{ij} = P(X_n=j | X_0=i) \).
- A stationary distribution of a Markov chain is a distribution \( \pi \) such that \( \pi_i > 0 \) for all \( i \), \( \Sigma_{i} \pi_i = 1 \), and \( \pi = M \pi \). It is also referred to as the invariant distribution or equilibrium distribution of the MC.
- If \( l_j = \lim_{n \to \infty} \{ [M^n]_{ij} \} \) exists for all \( j \) (independent of \( k \)) then \( l_j \) is the limiting distribution of the Markov chain.

A stochastic process (random walk) is a collection of random variables, \( X_i \), where \( i \in I \) for some indexing set \( I \), each \( X_i \) taking values in a state space, \( S \).

- A Markov chain is a stochastic process with a discrete indexing set \( I \) (e.g., the integers) such that the conditional distribution of \( X_{i+1} \) depends only on \( X_i \), that is \( P(X_{i+1}=j | X_1, X_2, \ldots, X_i) = P(X_{i+1}=j | X_i) \). That is, the transition probabilities \( P(X_{i+1}=j | X_i=k) \) are independent of \( i \).

The aim of MCMC is to construct a Markov chain which has a stationary limiting distribution \( \pi \) which is the same as the distribution we wish to simulate.

- There are many ways to do this, but the one most commonly used in practice is to use the principle of detailed balance such that the transitions between any pair of states maintains the equilibrium distribution. We say that a MC \( (M, \phi) \) with transition matrix \( M \) and distribution \( \phi \) satisfies the detailed balance condition iff \( M' \phi = M \phi \).
- Astronomers are familiar with the principle of detailed balance, which arises in stellar atmosphere and radiative transfer problems, for example.
Bayesian Inference 10/21/08 31

Metropolis-Hastings Method

- We construct the transition matrix $M$ as follows: We assume that at any step we are in the state $i$ and wish to construct the element of the transition matrix to state $j$. We specify an arbitrary (within limits) proposal distribution $q(j|i)$ and construct a quantity $\alpha_j'$ so that the matrix $M_j'=q(j|i)\alpha_j'$ is a Markov matrix (with constant entries, so that the resulting Markov chain will be stationary).
- The proposal distribution is sometimes called the jumping distribution.

Bayesian Inference 10/21/08 29

Markov Chains

- Theorem: If $(M, \phi)$ satisfies the detailed balance condition, then $\phi$ is the stationary distribution for $M$.
- Proof: Sum the detailed balance condition over $j$:
  \[ \sum_j M_j^i\phi_j = \sum_j M_j^i\phi_i = \phi_i \sum_j M_j^i = \phi_i \]
  \[ \therefore M\phi = \phi \]

Bayesian Inference 10/21/08 30

Markov Chains

- Some side conditions need to be satisfied for this to work, but in practical situations these can usually be satisfied
  - Irreducibility condition: There is a nonzero probability of going from any state $j$ to any state $k$ in a finite number $n$ of steps
  - Aperiodicity condition: A state $j$ in a MC is periodic if there exists an integer $d>1$ such that $[M^n]_{j,k}=0$ unless $n$ is divisible by $d$. Otherwise it is aperiodic.
  - Recurrency condition: A state $j$ is recurrent iff $\sum_{i=1}^{\infty} \{M^n\}_j>0$, otherwise it is transient.
  - Ergodic theorem: An irreducible, aperiodic (no periodic states), positively recurrent (recurrent in the future) MC has a unique limiting distribution which is the invariant distribution of the MC.

Bayesian Inference 10/21/08 31

Metropolis-Hastings Method

- We do this as follows:
  - If $q(j|i)\phi_j=q(i|j)\phi_j$ set $\alpha_j'=\alpha_j=1$
  - Otherwise assume WLG that $q(j|i)\phi_j>q(i|j)\phi_j$. Set $\alpha_j'=\frac{q(j|i)\phi_j}{q(i|j)\phi_j}$
  - Then we verify that $M'_j\phi_i=M_j\phi_i$.
    » Show this!
  - Thus the desired solution is
    \[ \alpha'_j = \min \left\{ 1, \frac{q(i|j)\phi_j}{q(j|i)\phi_i} \right\} \]
**Example of Metropolis-Hastings**

- We show how to draw a sample from the standard normal distribution using Metropolis-Hastings.

```r
# Proposal moves a random distance +/-
sample1 = function(x,a) {
  xprop = x+runif(1,-a,a)
  # Proposal moves a random distance +/-
  alpha = dnorm(xprop,log=T)-dnorm(x,log=T) # use logs!!!
  # log of log(p(xstar)/p(x))
  u = log(runif(1,0,1))
  accept=u<alpha
  x = accept*xprop + (1-accept)*x
  return(list(x=x,accept=accept))
}
```

**Metropolis-Hastings Method**

- The Metropolis-Hastings algorithm proceeds this way:
  - Start at an arbitrary point \( i \) in the state space \( S \).
  - Generate a random variable \( j \) from an arbitrary but fixed proposal distribution \( q(j|i) \) representing a proposed move from state \( i \) to state \( j \).
  - Calculate \( \alpha_j \) and generate a random variable \( u \) which is uniform on \([0,1]\)
    - If \( u<\alpha_j \) accept the proposal and move to state \( j \).
    - Otherwise, reject the proposal and remain in state \( i \).
  - Repeat until a large enough sample has been generated.

- The Gibbs sampler is a special case of Metropolis-Hastings, where the proposal distribution \( q(j|i) = \phi(j) \). For then, \( \alpha_j = 1 \) and the proposal is always accepted.
- Very important: we do not need to know the normalized target distribution; it is sufficient to know it up to a constant factor. The reason is that any constant factor will cancel when calculating \( \phi \).
- It is advisable always to calculate the logarithm of the ratio \( \phi \), since in real problems with even moderately large amounts of data the products of likelihoods quickly exceed the limits of machine arithmetic. Therefore we work with log likelihoods and priors, and compare the resulting log of the Metropolis-Hastings factor with the log of a uniform random variable.

- The example code above can be used to generate a sample of size 1000 from the standard normal distribution using Metropolis-Hastings.

```r
sampleen = function(n=1000,a=2,x=0) {
  xs = rep(NaN,n)
  accepts = 0
  for(i in 1:n){
    z = sample1(x,a)
    xs[i] = x = z$x
    accepts = accepts+z$accept
  }
  cat("Accepts = ",accepts/n,"\n")
  # code to display results
  invisible(list(x=xs,acc=accepts))
}
Example of Metropolis-Hastings

- We explore the sampler for this problem for various characteristics common to many M-H samplers
  - How large a move?
  - Where do we start?
  - How many steps to calculate?
  - Characteristics of this particular sampler

- We can start from various points
  - See that there may be need for a “burn-in” period where statistics are not gathered
  - Again, experience and knowledge of the problem can help decide on how long a “burn-in” period is needed
  - There is software to assist this decision (e.g., CODA, available at the BUGS website)

- It’s important not to make too large or too small a move. If the move is too large, it will be rejected with high probability and it will take a long time to sample the space. If too small, you will spend a lot of time getting anywhere because the moves are small
  - There is no hard and fast rule, but acceptances in the 30-60% range seem to work well
  - If proposals can be made that approximate the underlying distribution well, good behavior can be obtained

- The more steps calculated the better; and since the sampling is basically a frequentist exercise (to obtain Bayesian answers) we can generally expect the precision of our estimates to improve as \(1/\sqrt{N}\)
  - Sampling schemes that better sample the available sample space \(S\) will generally yield better performance with fewer samples
Example of Metropolis-Hastings

- We explore the sampler for this problem to understand characteristics common to many M-H samplers
- Characteristics of this particular sampler
  - Takes more than one step to access some states from other states, but finite number does the job
  - Symmetry of proposal simplifies the calculation
  \[
  q(x^* | x) = q(x | x^*) = f(x - x^*)
  \]
  for \( f \) an even function
  \[
  \therefore \frac{q(x | x^*)p(x^*)}{q(x^* | x)p(x)} = \frac{p(x^*)}{p(x)}
  \]

Metropolis-within-Gibbs

- The Metropolis-within-Gibbs idea retains the idea of sequential sampling, but uses a Metropolis step on some or all variables rather than attempting to sample from the exact conditional distribution.
- In other words, we propose a move of the variable in question from its current position to a new position in the state space (keeping all the other variables fixed); calculate an \( \alpha \) using the conditional distribution of that variable; decide whether to accept or reject, and then move onto the next variable
- We can mix “pure” Gibbs steps where we sample from the desired conditional distribution of some variables, with Metropolis steps on other variables

Metropolis-within-Gibbs

- The distinguishing feature of the Gibbs sampler was that we could sample first one variable conditioned on all the others, then a second variable, and so on, always conditioning on the most current values of the other variables.
- However, it has a drawback. You need to be able to draw a sample from each of the conditional distributions. If that can’t be done, you can’t use Gibbs
  - There are techniques for doing this, such as importance sampling (we may discuss this at some point)

Example: Consider the normal inference problem that we discussed, where we used Gibbs sampling on \( x \) and \( \sigma^2 \). We could have used a Metropolis-Hastings scheme on one or both of these variables, e.g. M-H on \( x \) and Gibbs on \( \sigma^2 \).
- We could do M-H on both variables, but we’d have to be careful since our prior on \( \sigma^2 \) says that \( \sigma^2 \geq 0 \). This means that if we use the “move by some random amount” idea that we used earlier, we would have to be careful near \( \sigma^2 = 0 \), since the proposal distribution \( q(*|\sigma^2) \) cannot take us to \( \sigma^2 \leq 0 \); alternatively we can just reject such steps if they are proposed (which will happen automatically if we are careful to define the posterior calculation to have value 0 when \( \sigma^2 \leq 0 \))—but be careful if using logs!!!