MCMC Review

- http://jackman.stanford.edu/mcmc/icpsr99.pdf
- http://students.washington.edu/fkrogsta/bayes/stat538.pdf
- http://www.stat.berkeley.edu/users/terry/Classes/s260.1998/Week9a/week9a/week9a.html
- http://omega.albany.edu:8008/cdocs/summer99/lecture3/l3.html
- http://www.maths.tcd.ie/~chas/node49.html
  - [Careful: His definition of aperiodicity is incorrect]
- [http://www.npac.syr.edu/users/gcf/CPS615NI95/p_NI.html] Start at slide 53.
- http://www.eecs.harvard.edu/~michaelm/CS222/MarkovChain.ps
- http://www.math.chalmers.se/Stat/Grundutb/Chalmers/TMS081/Lecture_notes/lecture3.ps

MCMC Review

- The basic idea behind MCMC (Markov chain Monte Carlo) is very simple: draw a sample from the full posterior distribution, and then make inferences using the sample as a representative of the posterior distribution
- Thus, if we are interested in the mean and variance of a parameter, we calculate the sample mean and sample variance of the parameter from the sample
- If we are interested in correlations between parameters, we calculate the sample correlations
- If we need the posterior distribution of a parameter, we display a histogram of the distribution on the parameter as given by the sample.

MCMC Review

- The ideas have been known for a long time
  - Metropolis-Hastings sampling was developed in the 1950s by physicists. The seminal paper was Metropolis, Teller, Teller, Rosenbluth and Rosenbluth (1953). A later paper by Hastings (1970) expanded on the technique
  - Gibbs sampling was invented later, and first described by Geman and Geman in 1984 in the context of image restoration
  - The methods have been used in statistics for approximately 15 years, and have revolutionized the practical application of Bayesian statistics

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, θ_1 and θ_2. Suppose we can sample from the *conditional* distributions

$$p(\theta_1 \mid \theta_2, D) \quad \text{and} \quad p(\theta_2 \mid \theta_1, D)$$

where D is the data. Then, starting at some initial point in parameter space, generate a *random walk*, a sequence as follows: $\{(\theta_1^{(0)},\theta_2^{(0)}),(\theta_1^{(1)},\theta_2^{(1)}),...,(\theta_1^{(k)},\theta_2^{(k)}),...\}$

Gibbs Sampling

- For k=1,…,n define

$$\theta_1^{(k)} \sim p(\theta_1 \mid \theta_2^{(k-1)}, D),$$
$$\theta_2^{(k)} \sim p(\theta_2 \mid \theta_1^{(k)}, D)$$

  where '~' means that we draw the value in question from the indicated distribution.
- Note that when a new value is generated, it immediately replaces the old value, and is not used again. The code should look something like this pseudocode:

```
repeat until done {
     th1 = sampleth1(th2,D)
     th2 = sampleth2(th1,D)
     savesamples(th1,th2)
}
```


Gibbs Sampling

- The resulting sequence of values is a *Markov chain*; the values at the (k+1)st step depend only on the values at the kth step and are independent of previous values
- The Markov chain will under mild conditions tend to a stationary distribution, and the stationary distribution will be the desired $p(\theta_1, \theta_2 \mid D)$
  - The chain must be able to reach any legal state in a finite number of steps
  - The chain must not have strictly periodic solutions
  - The chain must not have attractors or repellers

Gibbs Sampling

- The method generalizes to any number of variables, e.g.,

$$\theta_1^{(k)} \sim p(\theta_1 \mid \theta_2^{(k-1)}, \theta_3^{(k-1)}, \ldots, \theta_m^{(k-1)}, D),$$
$$\theta_2^{(k)} \sim p(\theta_2 \mid \theta_1^{(k)}, \theta_3^{(k-1)}, \ldots, \theta_m^{(k-1)}, D)$$
$$\vdots$$
$$\theta_m^{(k)} \sim p(\theta_m \mid \theta_1^{(k)}, \theta_2^{(k)}, \ldots, \theta_{m-1}^{(k)}, D)$$

- With 4 states, the code would look something like this:

```
repeat until done {
     th1 = sampleth1(th2,th3,th4,D)
     th2 = sampleth2(th1,th3,th4,D)
     th3 = sampleth3(th1,th2,th4,D)
     th4 = sampleth3(th1,th2,th3,D)
     savesamples(th1,th2,th3,th4)
}
```


Gibbs Sampling

- Suppose we have normally distributed estimates X_i, i=1,…,N, of a parameter μ, with unknown variance σ^2. The likelihood is

$$p(X|\mu,\sigma^2) \propto \sigma^{-N}\exp(-\Sigma(X_i-\mu)^2/2\sigma^2)$$

- Assume a flat (uniform) prior for μ and a "Jeffreys" prior $1/\sigma^2$ for σ^2. The posterior is proportional to prior times likelihood:

$$p(\mu,\sigma^2|X) \propto \sigma^{-(N+2)}\exp(-\Sigma(X_i-\mu)^2/2\sigma^2)$$

(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 σ^2 instead of σ so we use $d\sigma^2/\sigma^2$)

Gibbs Sampling

- The posterior distribution can be simplified using the trick of "completing the square" (where $\bar{x}$ is the sample mean)

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

$$= \sigma^{-(N+2)} \exp\left(-\frac{\sum(X_i-\bar{x})^2}{2\sigma^2}\right)\exp\left(-\frac{\sum(\bar{x}-\mu)^2}{2\sigma^2}\right)$$

$$= \sigma^{-(N+2)} \exp\left(-\frac{S_{xx}}{2\sigma^2}\right)\exp\left(-\frac{N(\bar{x}-\mu)^2}{2\sigma^2}\right)$$

$$p(\mu \mid \sigma^2, X) = p(\mu,\sigma^2 \mid X)/p(\sigma^2 \mid X)$$

Depends only on σ^2

Gibbs Sampling

- When sampling μ, σ will be fixed, so we can ignore factors dependent only on σ. Let $\bar{x}$ be the sample mean. Then the conditional distribution of μ can be rewritten, apart from constant factors

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

- This is readily seen to be normal with mean $\bar{x}$ and variance σ^2/N. So that's the distribution we need to use for sampling μ in each Gibbs step. Here's a function to produce one sample

```
samplemu = function(xbar,sdev,N){
  rnorm(1,xbar,sdev/sqrt(N))
}
```


Gibbs Sampling

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

$$p(\chi^2 \mid \mu, X) \propto \left(\chi^2\right)^{N/2+1} \exp\left(-\chi^2/2\right)\frac{d\sigma^2}{d\chi^2}$$

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

$$\sigma^2 \propto \frac{1}{\chi^2} \quad \text{so} \quad \frac{d\sigma^2}{d\chi^2} \propto \frac{1}{\left(\chi^2\right)^2}$$

- 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 σ^2 by dividing $\Sigma(X_i-\mu)^2$ by the value of χ^2 that we sampled. That allows us to sample σ^2 at each Gibbs step. [What we've done is to sample from an *inverse chi-square* distribution].

Gibbs Sampling

- Here's R code to produce one sample from the inverse chi-square distribution, and then turn it into a sample on the standard deviation σ by taking the square root:

```
samplesd = function(X,mu,N){
  sqrt(sum((X-mu)^2)/rchisq(1,N))
}
```

- Comment: σ^2 has an inverse chi-square distribution, and σ has an inverse chi distribution (with scale parameter)

Gibbs Sampling

- Calculation: This code fragment performs *n* sampling steps

```
for(i in 1:n){
  mus[i] = mu = samplemu(xbar,sdev,N)
  sdevs[i] = sdev = samplesd(X,mu,N)
}
```


Gibbs Sampling

- This code fragment initializes the Markov chain and produces n samples from the chain

```
samplen = function(n=1000,mu=0,sdev=1){
  xbar = mean(X)
  N = length(X)
  mus = rep(NaN,n)
  sdevs = rep(NaN,n)
  for(i in 1:n) {
    mus[i] = mu = samplemu(xbar,sdev,N)
    sdevs[i] = sdev = samplesd(X,mu,N)
  }
  list(mu=mus,sd=sdevs) # result
}
```


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:

```
z=samplen(1000)
quartz("mu histogram") #Mac only!
hist(z$mu,50) #mu marginal
quartz("mu plot")
plot(z$mu) #look at sequence
quartz("correlation plot")
plot(z$mu,z$sd) #look at correlations
cat("mubar = ",mean(z$mu),"\n")
cat("sdbar = ",mean(z$sd),"\n")
```


Markov Chains

- The aim of MCMC is to construct a Markov Chain which has a stationary limiting distribution π which is the same as the distribution we wish to simulate
- There are many ways to do this, but one often 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,ϕ) with transition matrix M and distribution ϕ satisfies the detailed balance condition *iff* $M^i_j \phi_i = M^j_i \phi_j$.
- Astronomers are familiar with the principle of detailed balance, which arises in stellar atmosphere and radiative transfer problems, for example.
- If detailed balance holds, then $M\phi = \phi$, and $\phi = \pi$, the stationary distribution

Metropolis-Hastings Algorithm

- 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

$$\alpha^i_j = \min\left\{1, \frac{q(i \mid j)\phi_j}{q(j \mid i)\phi_i}\right\}$$

  - generate a random variable u which is uniform on [0,1]
    - » If $u<\alpha^i_j$ accept the proposal and move to state j.
    - » Otherwise, reject the proposal and remain in state i. The new sample equals the old one.
  - Repeat until a large enough sample has been generated

Metropolis-Hastings Algorithm

- The one-dimensional Gibbs sampler is a special case of Metropolis-Hastings, where the proposal distribution $q(j|i)=\phi(j)$. For then, $\alpha^i_j=1$ 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. Any constant factor will cancel when calculating α^i_j.
- *Very Important*: It is advisable to calculate the *logarithm* of the ratio α^i_j since in real problems with even moderately large amounts of data the products of likelihoods quickly exceed the limits of machine arithmetic. Always work with log likelihoods and priors, and compare the resulting log of the Metropolis-Hastings factor with the log of a uniform random variable.

Example of Metropolis-Hastings

- We show how to draw a sample from the previous example using Metropolis-Hastings steps instead of exact sampling
- We devise functions to do a M-H step on each variable (mu, sg) separately
- We then put them together into a sampler that initializes and repeatedly samples sequentially

Example of Metropolis-Hastings

- Compute log of posterior probability

```
logpost = function(X,mu,sg){
  lf = sum(dnorm(X,mu,sg,log=T))
  return(lf - log(sg))
}
```

- Note that we do this by actually computing the log of each term in the original likelihood and summing them (equivalent to taking the product of the individual likelihoods)
  - We don't compute the total likelihood and then take the log, which would defeat the purpose
- Also, we will sample on σ rather than σ^2

Example of Metropolis-Hastings

- Sampling on mu:

```
samplemu=function(xbar,mu,sg,N,a=2) {
  mustar = mu + runif(1,-a,a)
  accept = 0
  alpha = logpost(X,mustar,sg)-
    logpost(X,mu,sg)
  u = log(runif(1,0,1))
  if(u<alpha){
    mu = mustar  #accept proposal
    accept = 1 #useful to keep track
  }
  list(mu=mu,accept=accept)
}
```


Example of Metropolis-Hastings

- Sampling on σ:

```
samplesg=function(X,mu,sg,N,a=2) {
  sgstar = abs(sg+runif(1,-a,a))
  accept = 0
  alpha = logpost(X,mu,sgstar)-
    logpost(X,mu,sg)
  u = log(runif(1,0,1))
  if(u<alpha){
    sg = sgstar  #accept proposal
    accept = 1 #useful
  }
  list(sg=sg,accept=accept)
}
```


Example of Metropolis-Hastings

- Initialize and sample

```
samplen = function(n=1000,mu=0,sg=1,
  a=2,b=2) {
  # Initialization phase

  mus = rep(NaN,n)
  sgs = rep(NaN,n)
  acceptmu = 0
  acceptsg = 0

# …Continues on next chart
```


Example of Metropolis-Hastings

- Initialization routine

```
# Sampling phase
  for(i in 1:n) {
    mul = samplemu(xbar,mu,sg,N,a)
    mus[i] = mu = mul$mu
    acceptmu = acceptmu + mul$accept
    sgl = samplesg(X,mu,sg,N,b)
    sgs[i] = sg = sgl$sg
    acceptsg = acceptsg + sgl$accept
  }
  list(mus=mus,acceptmu=acceptmu,
    sds=sgs,acceptsd=acceptsg)
}
```

Example of Metropolis-Hastings

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

Example of Metropolis-Hastings

- We explore the sampler for this problem for various characteristics common to many M-H samplers
  - Where do we start?
    - » 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)

Example of Metropolis-Hastings

- We explore the sampler for this problem for various characteristics common to many M-H samplers
  - How large a move?
    - » 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

Example of Metropolis-Hastings

- We explore the sampler for this problem for various characteristics common to many M-H samplers
  - How many steps to calculate?
    - » 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 any given states, but finite number does the job
    - » Symmetry of proposal on μ simplifies the calculation

$$q(\mu^* \mid \mu) = q(\mu \mid \mu^*) = f(\mu - \mu^*)$$

for f an even function

$$\therefore \frac{q(\mu \mid \mu^*)p(\mu^*)}{q(\mu^* \mid \mu)p(\mu)} = \frac{p(\mu^*)}{p(\mu)}$$

Example of Metropolis-Hastings

- f is no longer an even function for some values of σ when sampling on σ, because of the absolute value when calculating σ^*. However, it is still the case that

$$q(\sigma^* \mid \sigma) = q(\sigma \mid \sigma^*)$$

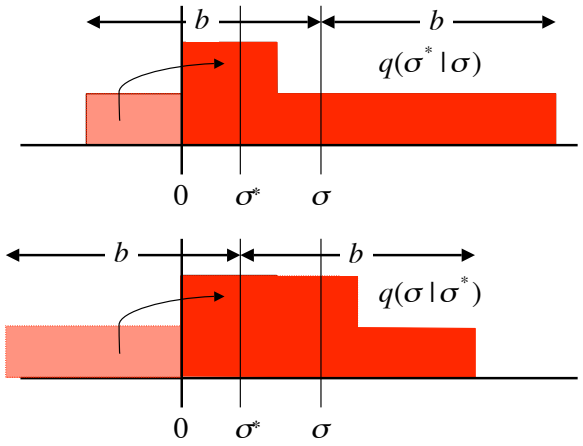
so that

$$\therefore \frac{q(\sigma \mid \sigma^*)p(\sigma^*)}{q(\sigma^* \mid \sigma)p(\sigma)} = \frac{p(\sigma^*)}{p(\sigma)}$$

This is because if both σ and σ^* are within b of 0, then there are two equally probable ways to get from σ to σ^*, and two equally probable ways to get back, so the the q's will still cancel.

Example of Metropolis-Hastings

- Here we show the q functions when $\sigma < b$



Metropolis-within-Gibbs

- The distinguishing feature of the exact 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 exact Gibbs
  - There are techniques for doing this under some circumstances, such as importance sampling and slice sampling, which I won't discuss

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
  - Mathematically this is equivalent
- That is, 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 α 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

Poisson Data With Background

- Tom Loredo presented the case of Poisson data with a background
- His posterior distribution was proportional to

$$p(s,b \mid N_{on}, N_{off}) \propto (s+b)^{N_{on}} e^{-(s+b)T_{on}} b^{N_{off}} e^{-bT_{off}} [s \geq 0 \,\&\, b \geq 0]$$

- Using variables $u=b+s$, b, this can be rewritten

$$p(u,b \mid N_{on}, N_{off}) \propto u^{N_{on}} e^{-uT_{on}} b^{N_{off}} e^{-bT_{off}} [u \geq b \,\&\, b \geq 0]$$

- where $u \geq b$ so that the prior is 0 whenever $u<b$. This is a product of *independent* gammas but truncated to satisfy $u \geq b$.

Poisson Data With Background

- We can simulate by drawing simultaneous proposals u^*, b^* from the obvious independent gamma distributions and accepting the proposal if $u^* \geq b^*$, keeping the old values if we do not accept.
- This works because the ratio p/q is a constant whenever we will accept (so the M-H factor is exactly 1 then); we only reject unphysical proposals

$$\frac{p(u^*,b^* \mid D) q(u,b \mid u^*,b^*)}{p(u,b \mid D) q(u^*,b^* \mid u,b)} = 1 \text{ when } u^* \geq b^*$$