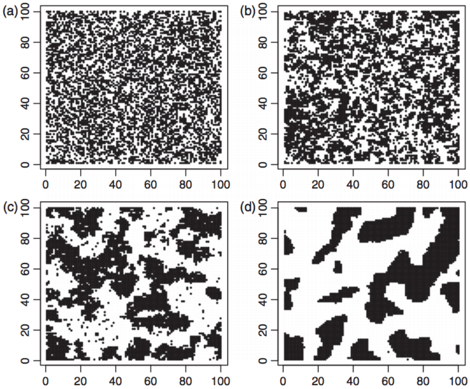
Lecture 10
Gibbs Sampling and Bayesian Computations

Dennis Sun
Stanford University
Stats 253

July 15, 2015

A Puzzle

How were these plots from Lecture 8 generated?



These are simulations of the Ising model

$$p(y_i | y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n) = \frac{e^{y_i \phi \sum_{j \in N(i)} y_j}}{1 + e^{\phi \sum_{j \in N(i)} y_j}},$$

but we can't even compute the likelihood $p(\mathbf{y})$!

Gibbs Sampling

Sometimes, it is easy to sample from the conditionals

$$p(y_i | y_1, ..., y_{i-1}, y_{i+1}, ..., y_n),$$

but not the joint distribution $p(\mathbf{y})$.

Gibbs sampling starts at a random point $\mathbf{y}^{(0)}$ and recursively generates

$$\begin{aligned} y_1^{(k)} &\sim p(y_1 | y_2^{(k-1)}, y_3^{(k-1)}, ..., y_n^{(k-1)}) \\ y_2^{(k)} &\sim p(y_2 | y_1^{(k)}, y_3^{(k-1)}, ..., y_n^{(k-1)}) \\ &\vdots \\ y_i^{(k)} &\sim p(y_i | y_1^{(k)}, ..., y_{i-1}^{(k)}, y_{i+1}^{(k-1)}, ..., y_n^{(k-1)}). \end{aligned}$$

In this way, we obtain $\mathbf{y}^{(k)}$. As $k \to \infty$, the distribution of $\mathbf{y}^{(k)}$ approaches $p(\mathbf{y})$.

Gibbs Sampler for the Bivariate Normal

Let's try this for an example where we know the answer:

$$\mathbf{y} \sim N\left(\mathbf{0}, \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix}\right).$$

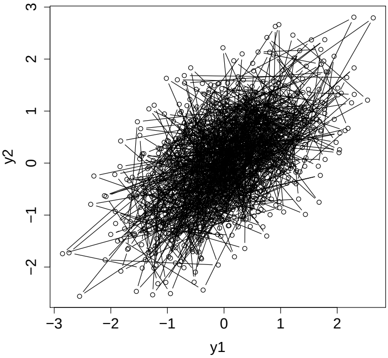
The Gibbs sampler generates

$$y_1^{(k)} \sim N(.5y_2^{(k-1)}, 1 - (.5)^2)$$
$$y_2^{(k)} \sim N(.5y_1^{(k)}, 1 - (.5)^2)$$

Gibbs Sampler for the Bivariate Normal

Here's some R code:

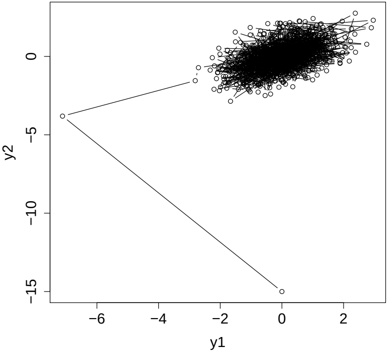
```
y1 <- 0
y2 <- 0
for(i in 1:1000) {
    y1[i+1] <- rnorm(1, .5*y2[i], .75)
    y2[i+1] <- rnorm(1, .5*y1[i+1], .75)
}
```



Gibbs Sampler for the Bivariate Normal

Now let's try some absurd initialization:

```
y1 <- 0
y2 <- -15
for(i in 1:1000) {
    y1[i+1] <- rnorm(1, .5*y2[i], .75)
    y2[i+1] <- rnorm(1, .5*y1[i+1], .75)
}
```



It's common practice to discard the first "few" samples. This is called the **adaptation period** (or burn-in period).

Why does Gibbs Sampling work?

We analyze a modification of the Gibbs sampler: a coordinate i is chosen uniformly from $\{1, \ldots, n\}$ and at iteration ℓ, we update

$$y_i^{(\ell)} \sim p(y_i | y_1^{(\ell-1)}, \ldots, y_{i-1}^{(\ell-1)}, y_{i+1}^{(\ell-1)}, \ldots, y_n^{(\ell-1)}),$$

holding all other coordinates fixed.

- $\{\mathbf{y}^{(\ell)}\}$ is a Markov chain with transition matrix

$$Q(\mathbf{y}, \mathbf{y}') = \begin{cases} \frac{1}{n} p(y_i' | \mathbf{y}_{-i}) & \text{if } y_j = y_j' \text{ for all } j \neq i \\ 0 & \text{otherwise} \end{cases}$$

- It is **reversible** with respect to $p(\mathbf{y})$:

$$p(\mathbf{y}) Q(\mathbf{y}, \mathbf{y}') = p(\mathbf{y}') Q(\mathbf{y}', \mathbf{y}).$$

- This implies that p is a stationary distribution of this chain:

$$\sum_{\mathbf{y}} p(\mathbf{y}) Q(\mathbf{y}, \mathbf{y}') = \sum_{\mathbf{y}} p(\mathbf{y}') Q(\mathbf{y}', \mathbf{y}) = p(\mathbf{y}').$$

- For “well-behaved” Markov chains, the chain will converge to the stationary distribution.

Application to the Ising Model

```
m <- 50
y <- matrix(rbinom(m^2, 1, .5), nrow=m, ncol=m)
phi <- 1

for(iter in 1:1000) {
    for(i in 1:m) {
        for(j in 1:m) {
            nb <- c()
            if(i > 1) nb <- c(nb, y[i-1,j])
            if(i < m) nb <- c(nb, y[i+1,j])
            if(j > 1) nb <- c(nb, y[i,j-1])
            if(j < m) nb <- c(nb, y[i,j+1])
            y[i,j] <- rbinom(1, 1, 1 / (1 + exp(-phi*mean(nb))))
        }
    }
}

image(y)
```

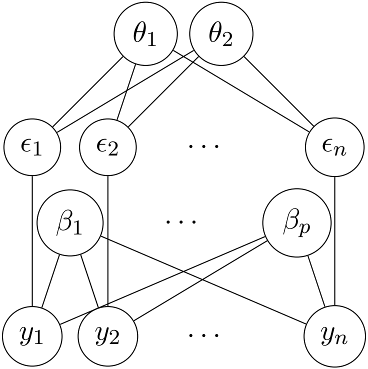
A Mystery

This is all really cool, but what does any of this have to do with Bayesian inference?

In fact, the Ising model is an example of a model that cannot be fit in BUGS or JAGS (because it's a cyclic graph).

Bayesian Models

Last time, we looked at models like the Bayesian kriging model:

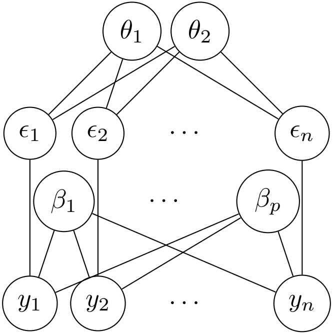


Remember that the goal was to obtain the posterior

$$p(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\epsilon} | \mathbf{y}).$$

We can use Gibbs sampling to obtain samples from this posterior.

Why is Gibbs sampling easy?



Gibbs sampling would require that we sample from conditional distributions, like $p(\epsilon_i|\mathbf{y}, \boldsymbol{\epsilon}_{-i}, \boldsymbol{\theta}, \boldsymbol{\beta})$. Why is this easy?

Because it is a local computation on the graph—it only depends on the parents and children of ϵ_i, not the whole graph!

$$p(\epsilon_i|\mathbf{y}, \boldsymbol{\epsilon}_{-i}, \boldsymbol{\theta}, \boldsymbol{\beta}) = p(\epsilon_i|y_i, \boldsymbol{\theta}) \propto p(\epsilon_i|\boldsymbol{\theta})p(y_i|\epsilon_i).$$

Further Simplifications: Conjugate Priors

$$p(\epsilon_i|y_i, \boldsymbol{\theta}) \propto \underbrace{p(\epsilon_i|\boldsymbol{\theta})}_{\text{prior}} \cdot \underbrace{p(y_i|\epsilon_i)}_{\text{likelihood}}.$$

We can think of $p(\epsilon_i|y_i, \boldsymbol{\theta})$ as just the posterior of ϵ_i given y_i.

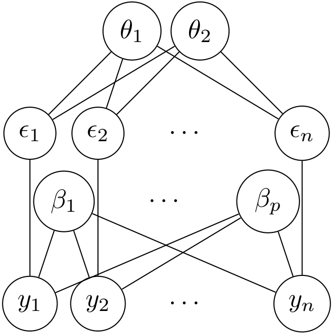
In many cases, the posterior is a familiar distribution—when the prior is the **conjugate prior** for the likelihood.

Example: normal prior $N(0, \tau^2)$, normal likelihood $N(\epsilon, \sigma^2)$:

$$\begin{aligned} p(\epsilon|y) \propto p(\epsilon)p(y|\epsilon) &\propto \exp\left\{-\frac{\epsilon^2}{2\tau^2}\right\} \exp\left\{-\frac{(y-\epsilon)^2}{2\sigma^2}\right\} \\ &\propto \exp\left\{-\frac{1}{2}\frac{\sigma^2+\tau^2}{\sigma^2\tau^2}(\epsilon - \frac{\tau^2}{\sigma^2+\tau^2}y)^2\right\}, \end{aligned}$$

so we see that $\epsilon|y \sim N\left(\frac{\tau^2}{\sigma^2+\tau^2}y, \frac{\sigma^2\tau^2}{\sigma^2+\tau^2}\right)$. Easy to sample!

Gibbs Sampling in Bayesian Kriging



Gibbs sampling in Gaussian kriging is straightforward because we choose most distributions to be normal to exploit conjugacy:

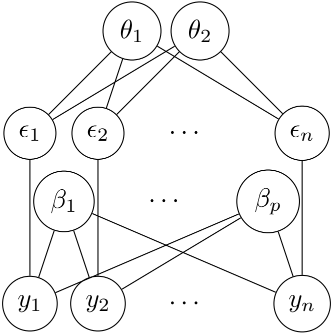
$$\boldsymbol{\beta} \sim N(\mathbf{0}, \nu^2 I)$$

$$\boldsymbol{\epsilon}|\boldsymbol{\theta} \sim N(\mathbf{0}, \Sigma(\boldsymbol{\theta}))$$

$$\mathbf{y}|\boldsymbol{\epsilon}, \boldsymbol{\beta} \sim N(X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tau^2 I)$$

(Only challenge is $\boldsymbol{\theta}$.)

Gibbs Sampling in Bayesian Kriging



Gibbs sampling in binomial kriging is *not* straightforward because the binomial is not conjugate to the normal:

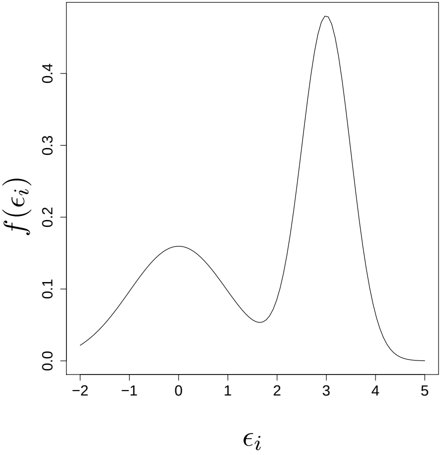
$$\boldsymbol{\beta} \sim N(\mathbf{0}, \nu^2 I)$$
$$\boldsymbol{\epsilon}|\boldsymbol{\theta} \sim N(\mathbf{0}, \Sigma(\boldsymbol{\theta}))$$
$$y_i|\boldsymbol{\epsilon}, \boldsymbol{\beta} \sim \text{Binom}(1, f(X\boldsymbol{\beta} + \boldsymbol{\epsilon}))$$

Other Conjugate Priors

prior	likelihood
normal	normal (mean)
Gamma	normal (variance)
beta	binomial
Gamma	Poisson

Sampling from General Distributions

The distribution $p(\epsilon_i|y_i, \boldsymbol{\theta}) \propto p(\epsilon_i|\boldsymbol{\theta})p(y_i|\epsilon_i)$ might be some weird distribution, like



How do we sample from a distribution like this?

Sampling from General Distributions

Metropolis algorithm: To sample from f, start at $\epsilon^{(0)}$. At iteration k,

1. Propose a new ϵ according to a jump distribution $J(\epsilon|\epsilon^{(k-1)})$.
2. Set $\epsilon^{(k)} = \epsilon$ with probability $\min\left(1, \frac{f(\epsilon)}{f(\epsilon^{(k-1)})}\right)$. Otherwise, stay put.

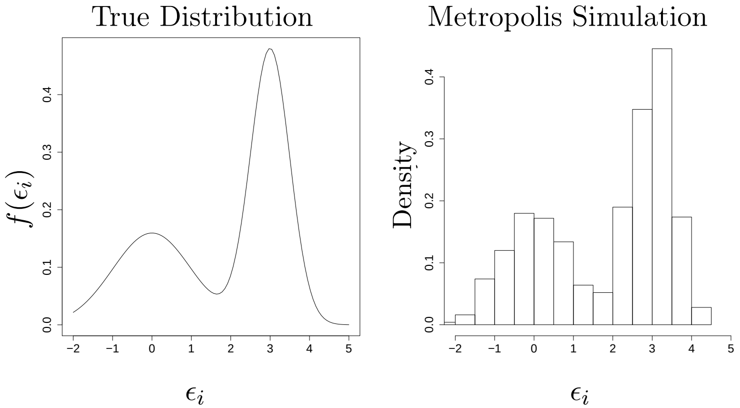
The distribution of $\epsilon^{(k)}$ approaches f as $k \to \infty$.

Why it works: Much like Gibbs sampling, it defines a Markov chain whose stationary distribution is the target distribution. Collectively, these methods are known as **Markov Chain Monte Carlo** (MCMC).

No need for normalizing constants! Notice that the Metropolis algorithm only depends on the ratio of f at two points. So we just need to know f up to a constant. This means we can just plug in $p(\epsilon_i|\boldsymbol{\theta})p(y_i|\epsilon_i)$ for f, rather than have to calculate $p(\epsilon_i|y_i, \boldsymbol{\theta}) = \frac{p(\epsilon_i|\boldsymbol{\theta})p(y_i|\epsilon_i)}{\int p(\epsilon_i|\boldsymbol{\theta})p(y_i|\epsilon_i)\, d\epsilon_i}$.

Sampling from General Distribution

```
eps <- 0
for(i in 1:1000) {
    eps.propose <- rnorm(1, eps[i], 1)
    if(runif(1) < p(eps.propose) / p(eps[i]))
        eps[i+1] <- eps.propose
    else eps[i+1] <- eps[i]
}
```

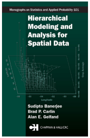


How JAGS Works

1. It forms a directed acyclic graph from the model you specify.
2. The overarching algorithm is Gibbs sampling. It goes through each node and samples from the conditional distribution at each node.
3. If there is a conjugate relationship at that node, then the conditional distribution is a known distribution, and JAGS can sample directly from it.
4. If the conditional distribution is not a known distribution, then JAGS uses the Metropolis algorithm (or other algorithms) to sample from it.

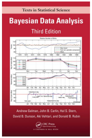
References

I have added the following reference to the course website:

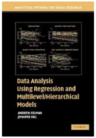


S. Banerjee, B. P. Carlin, and A. E. Gelfand. *Hierarchical Modeling and Analysis for Spatial Data*. Chapman and Hall 2003.

These are great references for Bayesian and hierarchical modeling.



A. Gelman *et al. Bayesian Data Analysis*. Third Edition. Chapman and Hall 2013.



A. Gelman and J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press 2006.