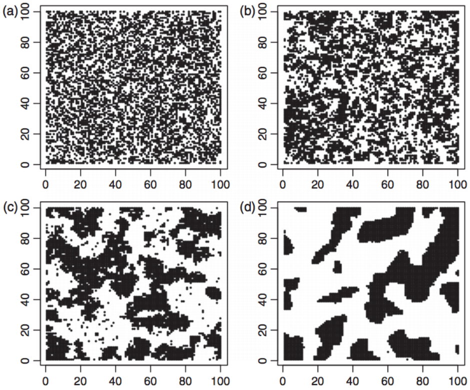
Lecture 10
Gibbs Sampling and Bayesian Computations

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:

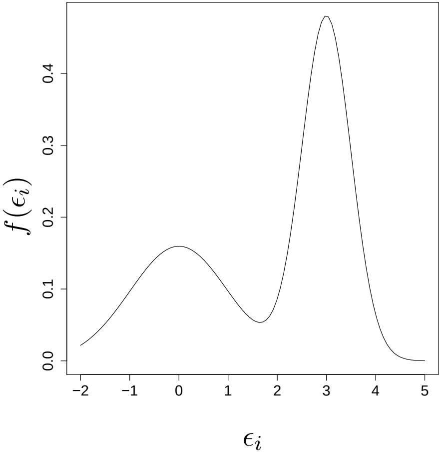
$$\mathbf{y} \sim N\left(\mathbf{0}, \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix}\right).$$

The Gibbs sampler generates

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

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 $p(\epsilon_i|y_i, \boldsymbol{\theta}) = \frac{p(\epsilon_i|\boldsymbol{\theta})p(y_i|\epsilon_i)}{\int p(\epsilon_i|\boldsymbol{\theta})p(y_i|\epsilon_i)\, d\epsilon_i}$.