Session 4

Bayesian Methods

Yuri Balasanov

University of Chicago, MScA

For any inquiries contact the author, Yuri Balasanov, at ybalasan@uchicago.edu or yuri.balasanov@iLykei.com .

Outline of the Session

- Motivation for Monte Carlo methods in Bayesian analysis
- History of Monte Carlo and MCMC
- Metropolis algorithm
- Gibbs sampling
- Summary of Metropolis and Gibbs sampling methods

Limitations of Bayesian Approach without Monte Carlo Techniques

- We tried Bayesian analysis using analytical approach with conjugate distributions and grid approximation.
  Both methods have limitations:
  - Analytical approach is limited by the number of conjugate pairs
  - Grid approximation is limited by dimensionality of the parameter space
- Application of Markov Chain Monte Carlo (MCMC) allows us to extend Bayesian analysis to wide range of important practical applications. This has been a breakthrough of the last couple decades
- The main idea of MCMC is approximation of posterior distribution by sampling from it
- MCMC requires that prior distribution $p(\theta)$ can be easily computed for any θ and that likelihood function can also be efficiently calculated for any sample and any θ

Monte Carlo and MCMC

Who Started it?

Figure: Missing Pictures: Arianna Rosenbluth, Augusta H. Teller.

Motivation for MCMC

- Grid approximation stops working in multidimensional parameter space because it requires N^d points evaluated, where N is the number of grid points for one parameter and d is the number of parameters
- Monte Carlo is expected to reduce the number of sampling points. But if simulation is done independently then it also requires the number of calculations that grows exponentially with growing number of dimensions d
- The only solution that may work is if simulations are not independent. In this case random walk needs to find the most populated area quickly and then spend more time exploring it
- This leads to a random walk in the form of Markov process: next step depends on the current state, but independent from the past

Metropolis Algorithm: Simple Case

Let parameter θ take values $\{\theta_1, \ldots, \theta_k\}$ with probabilities $\{p_1 = p(\theta_1), \ldots, p_k = p(\theta_k)\}$.
The steps of Metropolis algorithm are:

1. Let the current simulated value be θ_c.
2. The next candidate for simulation is selected between θ_{c-1} and θ_{c+1} with probabilities 0.5. Let the next candidate be θ_n.
3. Then if $p(\theta_n) > p(\theta_c)$ the next simulated value is θ_n.
   If $p(\theta_n) \leq p(\theta_c)$ then there is another step of randomization: the next simulated value is θ_n with probability $p = \frac{p(\theta_n)}{p(\theta_c)}$ and θ_c repeats with probability $(1 - p)$.

After repeating such steps N times the probabilities of simulated values $\{\theta_1, \ldots, \theta_k\}$ will be very close to $\{p_1 = p(\theta_1), \ldots, p_k = p(\theta_k)\}$

Remarkable properties of MCMC?

Note that we needed to do three simple things in order to implement MCMC:

1. Generate the next proposed value θ_n
2. Calculate $p(\theta_n)$ and $\frac{p(\theta_n)}{p(\theta_c)}$
3. Simulate uniform random variable $\xi \in [0, 1]$

The real value of MCMC is realized when probability distribution $p(\theta|\, Data)$ is posterior distribution and it does not have to be normalized, meaning that we just can use $p(\theta|\, Data) \propto p(\, Data|\, \theta)\, p(\theta)$

Markov Chain in Metropolis Algorithm

Metropolis algorithm: Markov chain.

$p(i|j) = P\{\theta_i | \theta_j\}$ probability of moving from θ_j to θ_i. Note: $p(i|j) = 0$; $|i-j| > 1$

$p(i|i+1) = 0.5 \min\left[\frac{p(i+1)}{p(i)}, 1\right]$ $p(i+1|i) = 0.5 \min\left[\frac{p(i)}{p(i+1)}, 1\right]$

$$\frac{p(i|i+1)}{p(i+1|i)} = \frac{0.5 \min\left[\frac{p(i+1)}{p(i)}, 1\right]}{0.5 \min\left[\frac{p(i)}{p(i+1)}, 1\right]} = \begin{cases} \frac{1}{\frac{p(i)}{p(i+1)}}, & p(i+1) > p(i) \\ \frac{\frac{p(i+1)}{p(i)}}{1}, & p(i+1) < p(i) \end{cases} = \frac{p(i+1)}{p(i)}$$

Ratio of transition probabilities back and forth equals ratio of target probabilities.

Transition probabilities depend only on previous step: Markov property.

Transition Matrix in Metropolis Algorithm

Transition Matrix.

$$\begin{bmatrix} p(1|1) & p(2|1) & 0 & \dots & & & \\ p(1|2) & p(2|2) & p(3|2) & 0 & \dots & & \\ 0 & p(2|3) & p(3|3) & p(4|3) & 0 & \dots & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & \dots & 0 & p(k-1|k) & p(k|k) & p(k+1|k) & 0 & \dots \\ & & & & \dots & 0 & p(n-1|n) & p(n|n) & p(1|n) \end{bmatrix}$$

This is a matrix of transition probabilities of a Markov chain H.

Transitions over q steps are given by H^q.

Such Markov chain has stable distribution and stable distribution is the target distribution

Thus convergence of the process !!!

General Form of Metropolis Algorithm I

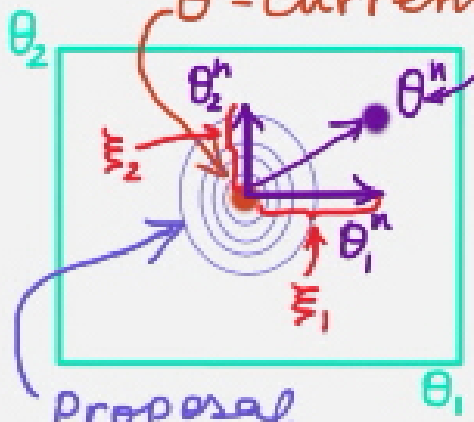
- Metropolis algorithm was published in 1953 by Metropolis, Rosenbluth, Rosenbluth Teller & Teller and got the name from the first author. Some sources give credit for the algorithm to his coauthors Marshall and Arianna Rosenbluth
- General algorithm applies to continuous multidimensional parameter space with posterior distribution $p\left(\theta \middle| D\right), \theta = \{\theta_1, \ldots, \theta_k\}$
- It starts from θ^0, for which $p\left(\theta^0 \middle| D\right) \neq 0, \theta^0 = \left\{\theta_1^0, \ldots, \theta_k^0\right\}$
- If current point is θ^c then next candidate θ^{new} is generated from some simple distribution. For example, following slide shows random walk in 2-parametric state space using 2-dimensional normal distribution with mean $(0, 0)$, standard deviation $(0.2, 0.2)$ and zero correlation
- Once the next candidate is selected the decision is made the same way as in multinomial Metropolis example: candidate θ^{new} is accepted with probability 1 if $p\left(\theta^{new} \middle| D\right) > p\left(\theta^c \middle| D\right)$; if $p\left(\theta^{new} \middle| D\right) \leq p\left(\theta^c \middle| D\right)$ then θ^{new} is accepted with probability $\frac{p(\theta^{new}|D)}{p(\theta^c|D)}$, otherwise the next sampling point is θ_c again

General Form of Metropolis Sampling II

Metropolis sampling

$\theta = (\theta_1, \theta_2)$ parameter space.

θ^c - current state



New proposal

Proposal distribution

Proposal

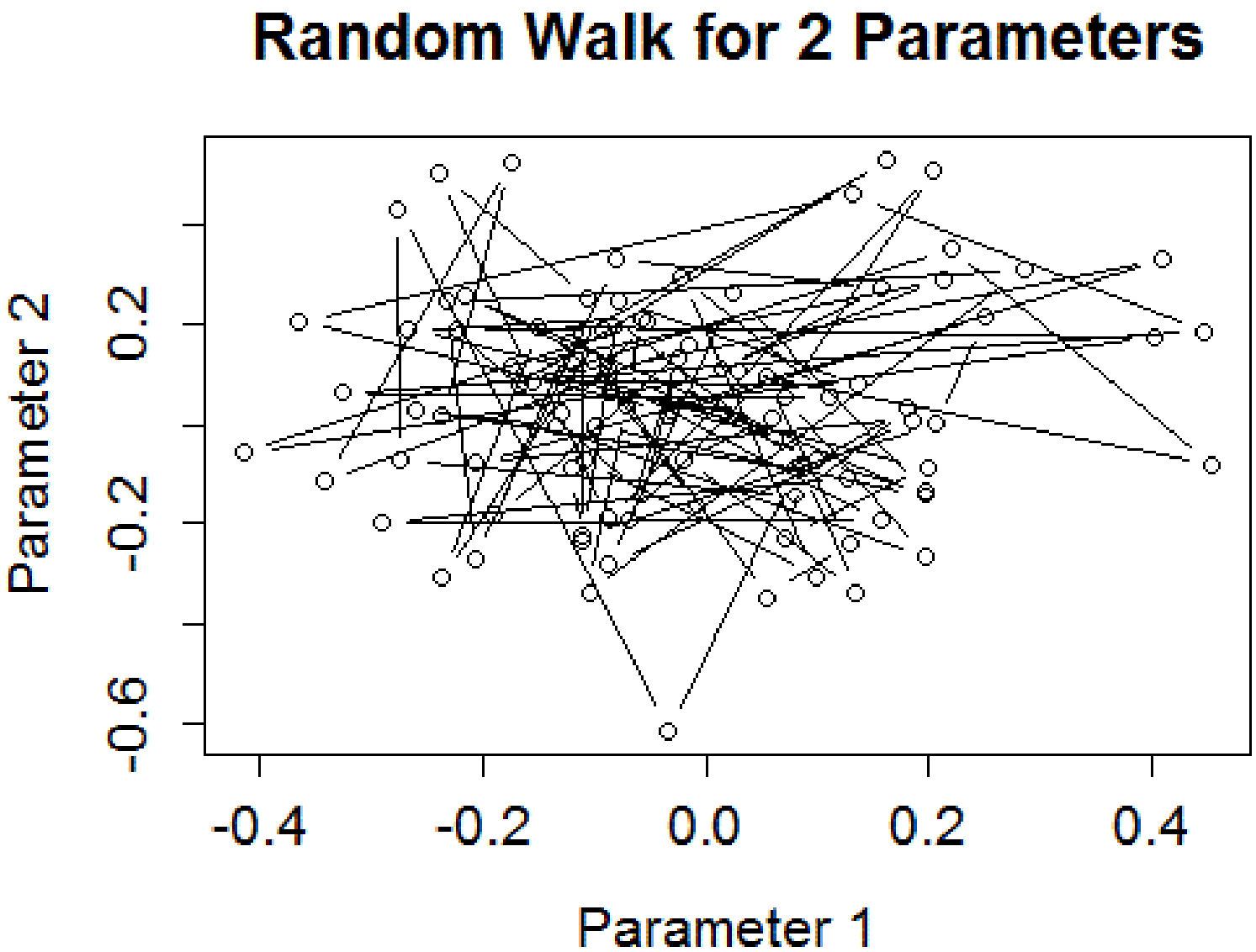
(ξ_1, ξ_2) - independent sample

$(\xi_1, \xi_2) \sim N(\mu = (0,0), \sigma = (\sigma_1, \sigma_2), \rho = 0)$

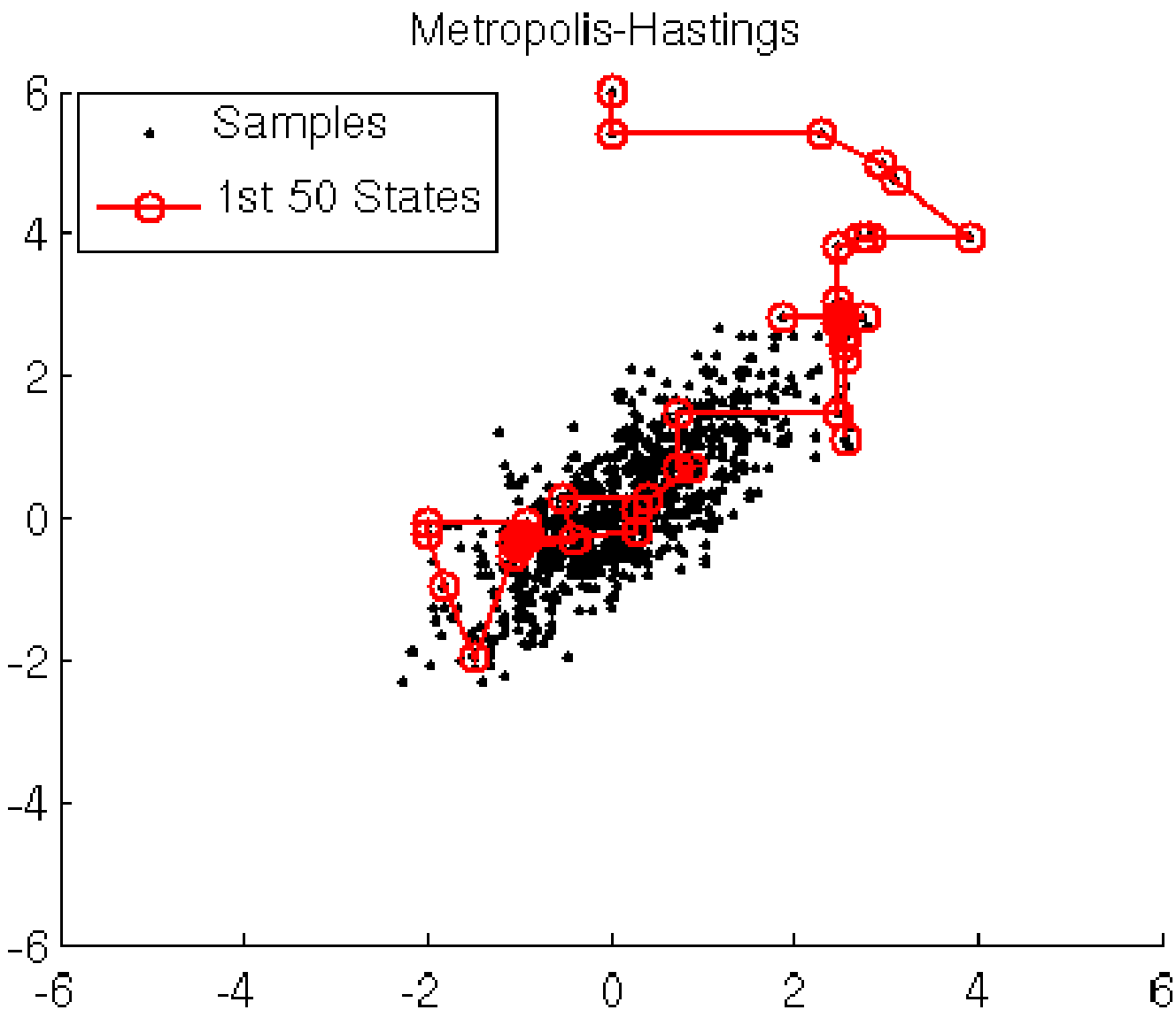
Acceptance: $p(\theta^n|\mathfrak{D}) > p(\theta^c|\mathfrak{D}) \Rightarrow P\{\theta^n\} = 1$

$$p(\theta^n|\mathfrak{D}) < p(\theta^c|\mathfrak{D}) \Rightarrow \begin{cases} P\{\theta^n\} = \dfrac{p(\theta^n|\mathfrak{D})}{p(\theta^c|\mathfrak{D})} \\ P\{\theta^c\} = 1 - \dfrac{p(\theta^n|\mathfrak{D})}{p(\theta^c|\mathfrak{D})} \end{cases}$$

Random Walk by Metropolis Algorithm for 2 Parameters



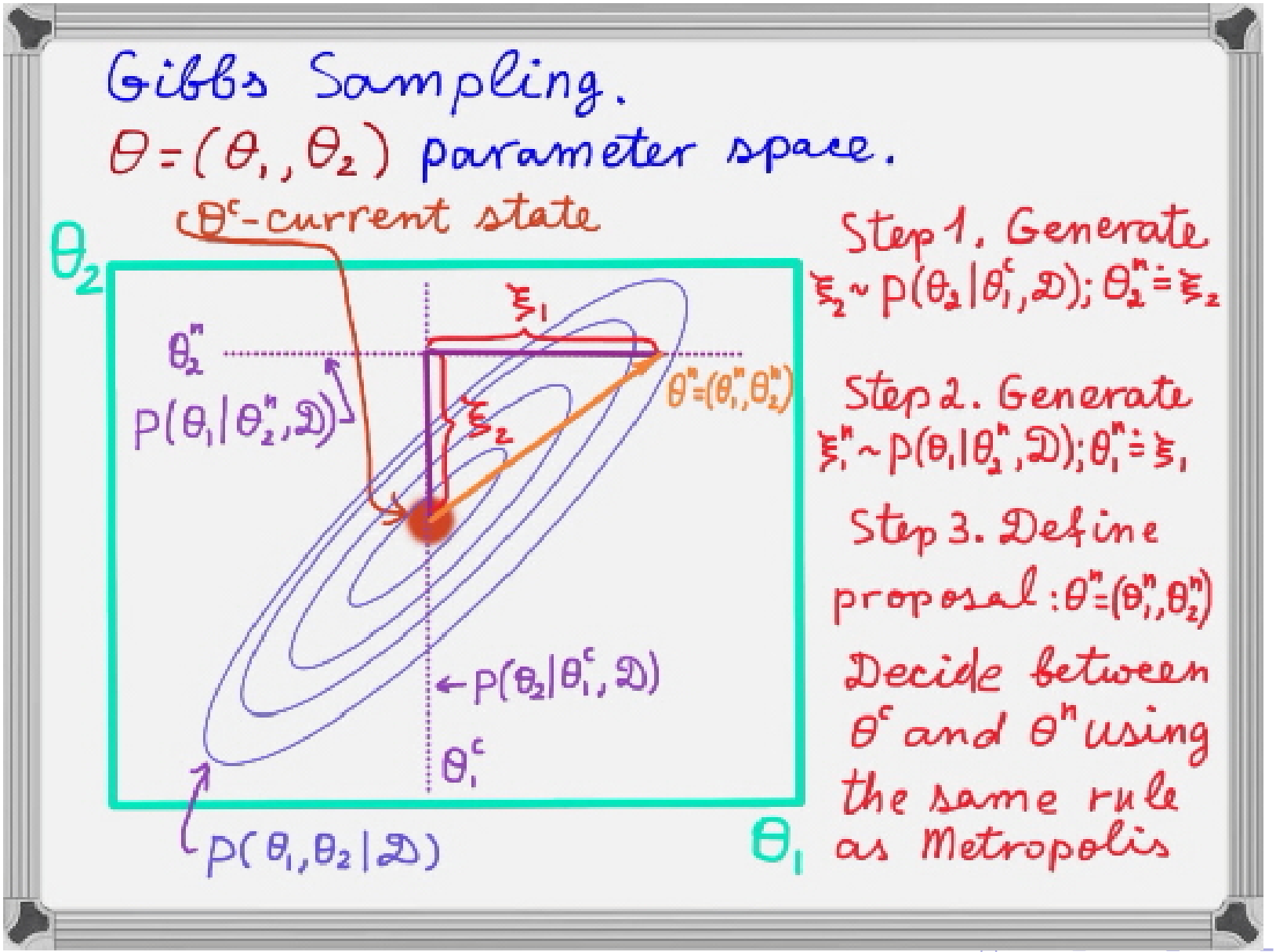
Metropolis-Hastings Random Walk: Exploration of Distribution



Gibbs Sampling

- Metropolis sampling is a very general method replacing straight Monte Carlo with simulations of a Markov chain process that is efficient numerically and converges to the target distribution faster
- A significant limitation of Metropolis MCMC appears in multidimensional parameters space: next candidate is generated by an arbitrary proposal distribution
- If the proposal distribution is very different from the target distribution then significant number of proposed candidates is rejected reducing efficiency of the algorithm
- A modification of the Metropolis MCMC algorithm samples from one-dimensional posterior marginal distribution directly solving the problem of tuning proposal distribution to the posterior distribution. This modification is called **Gibbs sampling**

Gibbs Sampling II



Summary of the Two Approaches

- Metropolis algorithm is a general algorithm for sampling from a given distribution. Distribution does not have to be normalized.
- In multidimensional state space proposal distribution needs to be tuned to the posterior distribution to avoid multiple rejections
- Gibbs sampling is a particular case of Metropolis algorithm in which generation of a new proposal consists of steps of sampling from marginal distributions for each of the parameters
- Gibbs sampling does not require tuning to posterior distribution, it samples directly from it. But it requires knowing the posterior distribution
- Both distributions are very efficient for analysis of hierarchical models