Markov Chain Monte Carlo (MCMC): Introduction

Dimitris Fouskakis

Department of Mathematics
School of Applied Mathematical and Physical Sciences
National Technical University of Athens

fouskakis@math.ntua.gr

Spring Semester

Markov Chains

- A Markov Chain $\{X^{(t)}\}$ is a sequence of dependent random variables

$$X^{(0)}, X^{(1)}, X^{(2)}, \ldots, X^{(t)}, \ldots$$

such that the probability distribution of $X^{(t)}$ given the past variables depends only on $X^{(t-1)}$. This conditional probability is called a **transition kernel** K. Thus

$$X^{(t+1)}|X^{(0)}, X^{(1)}, X^{(2)}, \ldots, X^{(t)} \sim K(x^{(t)}, x^{(t+1)}),$$

with $x^{(i)}$ denoting the observed value of $X^{(i)}$.

- For example, in a *simple random walk*

$$X^{(t+1)} = X^{(t)} + \epsilon_t, \ \epsilon_t \sim N(0, 1)$$

the kernel $K(x^{(t)}, x^{(t+1)})$ is the pdf of $N(x^{(t+1)}|\mu = x^{(t)}, \sigma^2 = 1)$.

- The *State Space* $\mathcal{X}$ of a Markov chain, is the set of values that each $X^{(t)}$ can take. The state space can be discrete or continuous; here we will assume is continuous.
- The index t take values in the *Index Set* $\mathcal{T}$ (usually time) which can also be either discrete or continuous. Here we will assume is discrete.

Markov Chains - Properties

- A Markov Chain is called **(ϕ -) Irreducible** if the sequence $\{X^{(t)}\}$ has positive probability of eventually reaching any region A of the state space (with $\phi(A) > 0$), no matter the starting value $X^{(0)}$ is.
- The pdf $f(y)$, $y \in \mathcal{X}$, is called **stationary probability distribution** of a Markov Chain $\{X^{(t)}\}$ if $X^{(t)} \sim f$, then $X^{(t+1)} \sim f$. Formally

$$\int_{\mathcal{X}} K(x, y) f(x) = f(y)$$

- A Markov Chain is called **reversible** with respect to a pdf f on $\mathcal{X}$, if the **detailed balance equation** holds:

$$f(x)K(x, y) = f(y)K(y, x), \ x, y \in \mathcal{X}$$

  If a Markov Chain is **reversible** with respect to f then f is a stationary distribution for the chain.
- A Markov Chain with stationary distribution f is called **aperiodic** if there do not exist $d \geq 2$ and disjoint subsets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_d \subseteq \mathcal{X}$, with $K(x, \mathcal{X}_{i+1}) = 1$, for all $x \in \mathcal{X}_i$ $(1 \leq i \leq d-1)$, and $K(x, \mathcal{X}_1) = 1$ for all $x \in \mathcal{X}_d$, such that $f(\mathcal{X}_1) > 0$ (and hence $f(\mathcal{X}_i) > 0$ for all i). (Otherwise the chain is **periodic** with period d and periodic composition $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_d$).

Markov Chains - Properties (cont'd)

- A Markov Chain with stationary distribution f is called **Harris recurrent** if for all $\mathcal{B} \subseteq \mathcal{X}$, with $f(\mathcal{B}) > 0$, and all $x \in \mathcal{X}$, the chain will eventually reach $\mathcal{B}$ from x with probability 1.
- Let $\mathcal{A} \subseteq \mathcal{X}$ ($\mathcal{A}$ measurable set). Then for $n \in \mathbb{N}$, we define $K^n(x, \mathcal{A})$, $x \in \mathcal{X}$, for the n-step transition law, as

$$K^n(x, \mathcal{A}) = \mathbb{P}[X^{(n)} \in \mathcal{A} | X^{(0)} = x].$$

  We call $lim_{n \to \infty} K^n(x, \cdot)$ the **limiting distribution** of the chain.
- Let $\{X^{(t)}\}$ be Markov Chain, with stationary distribution f. If the Markov chain is f - **irreducible**, then f is the **unique stationary** distribution of the Markov Chain. If additionally the chain is **aperiodic**, the **limiting distribution** of the chain **converges** to f, for almost every $x \in \mathcal{X}$. If additionally the chain is **Harris recurrent**, the limiting distribution of the chain **converges** to f, for all $x \in \mathcal{X}$.

Markov Chains - Properties (cont'd)

- As a consequence of the last result we have the **Ergodic Theorem**. Ergodicity has major consequences from a simulation point of view. In particular, the Ergodic Theorem says that for integrable functions h, we have

$$\frac{1}{T}\sum_{t=1}^{T} h\left(X^{(t)}\right) \rightarrow \mathbb{E}_f[h(X)],$$

  which means the **Strong Law of Large Numbers** that lies at the basis of Monte Carlo methods can also be applied in Markov Chains.

Markov Chain Monte Carlo

- The working principle of **Markov Chain Monte Carlo** (MCMC) methods is the following. Given a target density f, we build a Markov Kernel K, with a unique stationary distribution f. We generate a Markov Chain $\{X^{(t)}\}$ then, using this kernel, so that the limiting distribution of $\{X^{(t)}\}$ is f and integrals can be approximated according to the Ergodic Theorem.
- Will see two methods for constructing such a kernel K, that is associated with an arbitrary (target) density f.
  1. Metropolis-Hastings Algorithm
  2. Gibbs Sampling

Metropolis-Hastings Algorithm

The **Metropolis-Hastings (M-H) Algorithm** associated with the objective (**target**) density f and the conditional density q (**proposal**) produces a Markov chain $\{X^{(t)}\}$ through the following transition kernel:

Algorithm · **Metropolis–Hastings**

Given $x^{(t)}$,

1. Generate $Y_t \sim q(y|x^{(t)})$.
2. Take

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability } \rho(x^{(t)}, Y_t), \\ x^{(t)} & \text{with probability } 1 - \rho(x^{(t)}, Y_t), \end{cases}$$

where

$$\rho(x, y) = \min\left\{ \frac{f(y)}{f(x)} \frac{q(x|y)}{q(y|x)}, 1 \right\}.$$

- To avoid numeric overflow problems, it is highly recommended to use the log-scale.
- We start the algorithm with an initial value $x^{(0)} \in \mathcal{X}$.
- To check the condition in step 2, we draw $U \sim U(0, 1)$. If $U \leq \rho(x^{(t)}, Y_t)$ we accept the move, otherwise we stay where we are.

Metropolis-Hastings Algorithm (cont'd)

- The distribution q is called **proposal**. The probability $\rho(x, y)$ is called **acceptance probability**. It is to be distinguished from the **acceptance rate**, which is the average of the acceptance probability over iterations:

$$\bar{\rho} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} \rho(X^{(t)}, Y_t) = \int_{\mathcal{X}} \rho(x, y) f(x) q(y|x) dy dx,$$

  where T denotes the total number of iterations.
- The produced chain from M-H is **reversible** with respect to f (satisfies the detailed balance equation), *for any choice* of q! Therefore f is a stationary distribution.
- If the proposed move is rejected, then the algorithm stays at the current state for another iteration. This produces an **aperiodic chain**.
- By using the Ergodic Theorem, we can estimate any expected value, w.r.t. f, by producing a Markov Chain $\{X^{(t)}\}$ with the M-H algorithm. In words, we can start at essentially any x, run the chain for a long time, and the final draw has a distribution that is approximately f. The "long time" is called the **burn-in**.

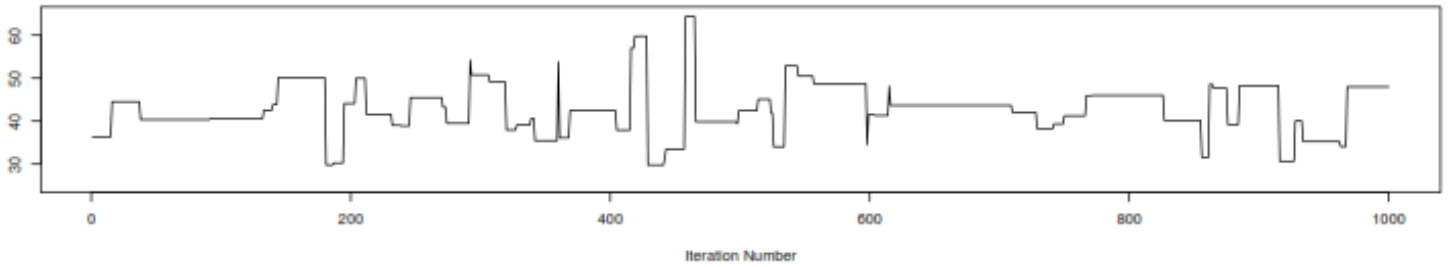
Metropolis-Hastings Algorithm (cont'd)

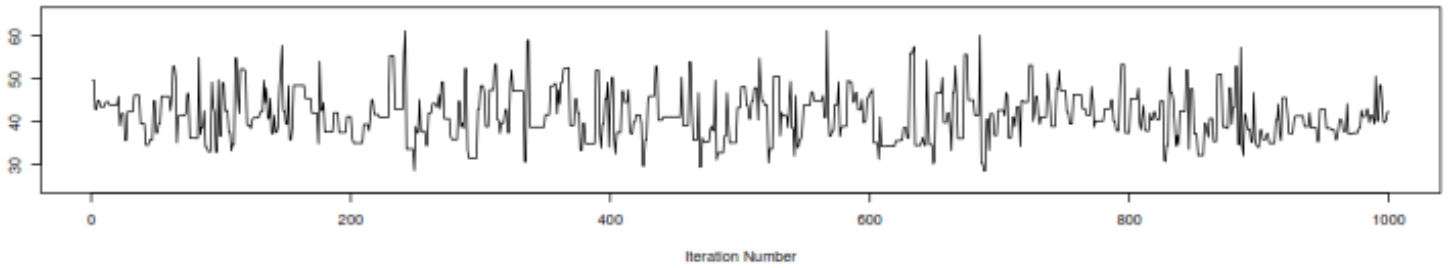
- The performance (**convergence**) of the algorithm will strongly depend on the choice of q. Furthermore in some cases (e.g. independent M-H; see later) specific choices of q may result to chains that are not irreducible.
- Quite often q is chosen in such a way that: (a): looks like a somewhat overdispersed version of f; (b): $\mathbb{E}_q[Y_t|x^{(t)}] = x^{(t)}$ and therefore when you do make a move, there will be an approximate left-right balance, so to speak, in the direction you move away from current $x^{(t)}$, which will encourage a faster exploration of the state space.
- An interesting property of the M-H algorithm is that it depends to f only via *ratios*. Therefore we do not need to know **normalizing constants** and equivalently we can work with $f' \propto f$. This makes the algorithm ideal for simulating values from the posterior distribution in Bayesian Statistics.
- Usually, we represent the produced sequence $\{x^{(t)}\}$ by a **trace plot**. In this plot we can see if the sequence has converged. If all values are within a zone without strong periodicities and (especially) tendencies, then we can assume convergence.

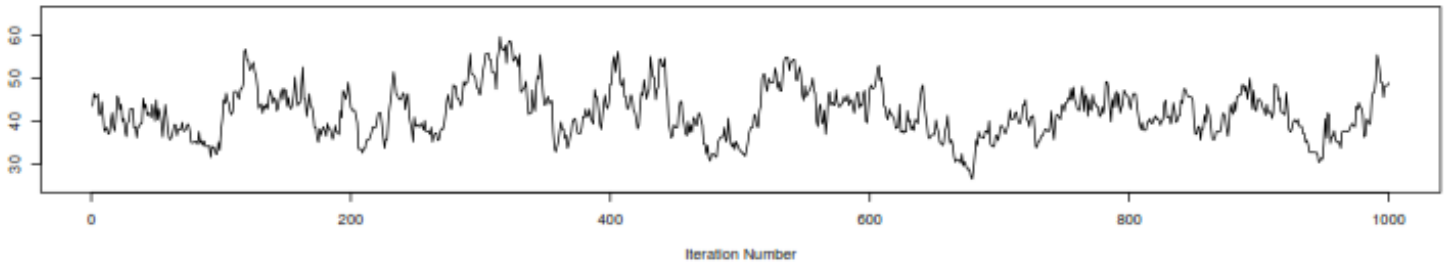
Metropolis-Hastings Algorithm (cont'd)

- If the proposal has been chosen in such way that most moves are rejected (i.e. small $\bar{\rho}$), then the algorithm will converge slow (1st plot next slide).
- If the proposal has been chosen in such way that most moves are accepted (i.e. large $\bar{\rho}$), then the algorithm usually moves in a very small area (around the mode of f) and therefore again fails to explore the whole state space fast (3rd plot next slide).
- Choose q in such a way so $\bar{\rho} \approx 0.45$ (2nd plot next slide).
- As we said before usually we select q to be "close" to f, with mean value equal to the current value of the chain $x^{(t)}$. Then we select (tune) the variance of q, so that $\bar{\rho} \approx 0.45$. For special cases of M-H algorithm (see next slides) this value is different.

Metropolis-Hastings Algorithm (cont'd)

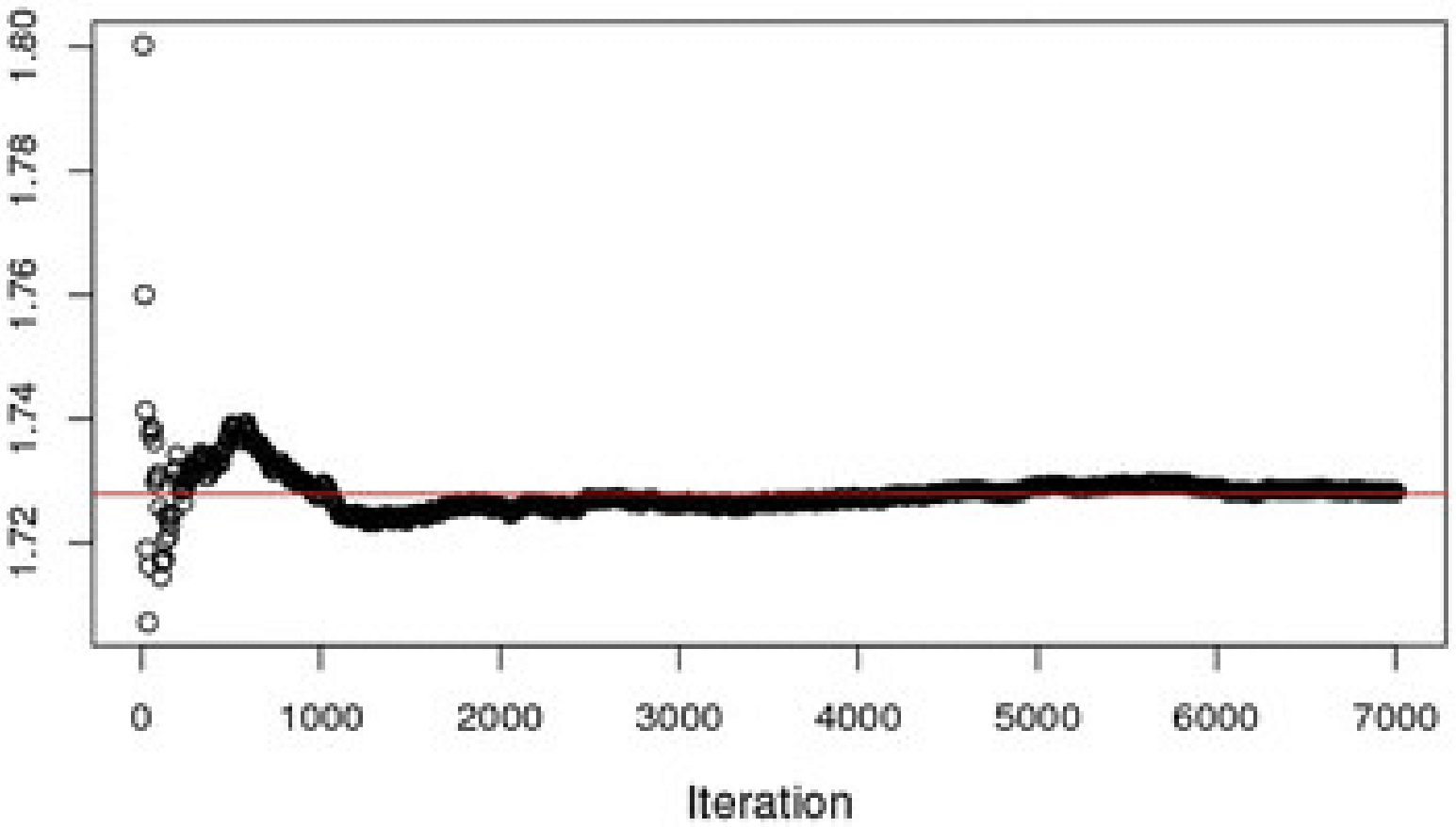






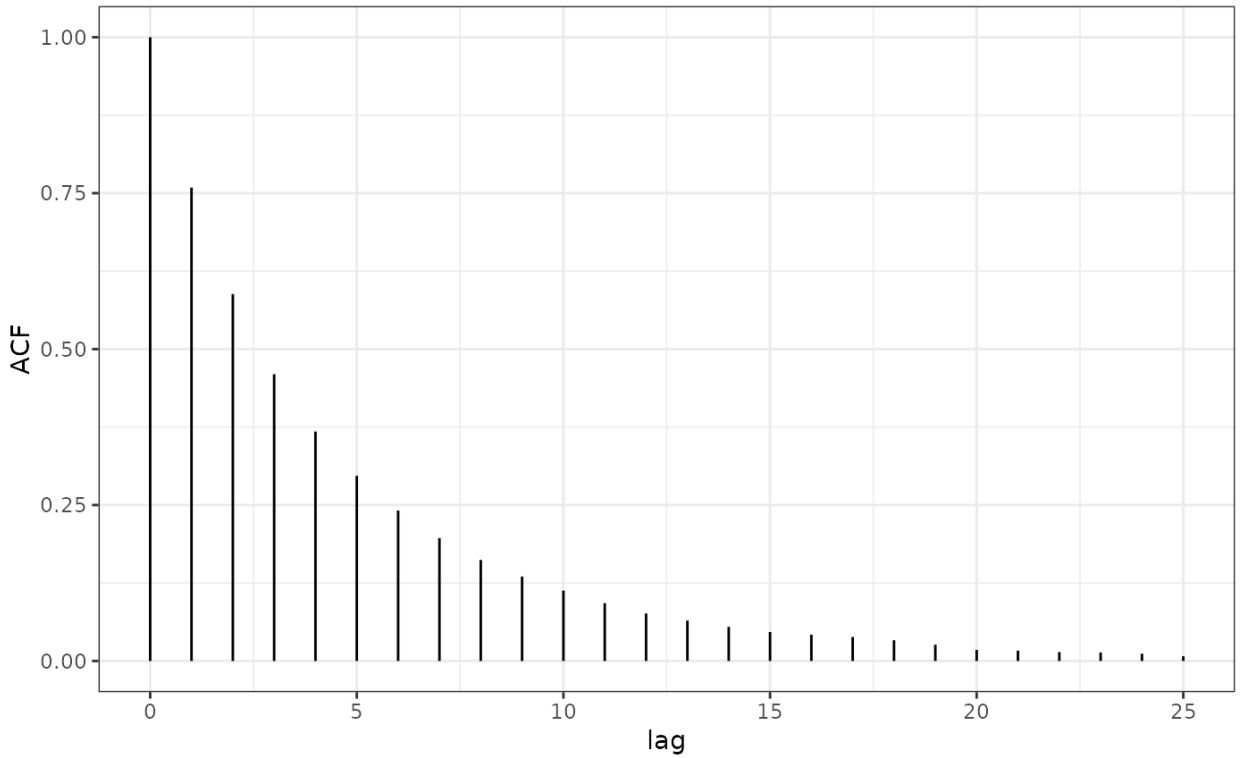
Metropolis-Hastings Algorithm (cont'd)

- In addition, we can create an **ergodic mean plot**. With this plot, we see how the average value in our chain changes as we move on. For the Ergodic Theorem to hold this average should converge eventually. The plot can help us also to chose the burn-in period.



Metropolis-Hastings Algorithm (cont'd)

- Finally, an **autocorrelation plot** can reveal the order (**lag**) of high autocorrelations in our chain. In cases where we have high order autocorrelations the chain "learns" with a slow rate, and therefore we should run the algorithm for a larger number of iterations (using possibly **thinning** - for example keeping every 5th value).



Metropolis-Hastings Algorithm (cont'd)

- Suppose that we run our chain for T iterations and the burn-in period (including the initial value) is $B+1$ ($B << T$). Then, if our aim is to estimate $\theta = \mathbb{E}_f[h(X)]$, we can use the **MCMC esimator**, according to the Ergodic Theorem:

$$\hat{\theta} = \frac{1}{T-B-1} \sum_{t=B+2}^{T} h(x^{(t)})$$

- We can use the **batch mean method** to estimate the **Monte Carlo Error** of our estimator. That is, we divide the chain (after the burn-in period) into k groups of approximately equal size (ν) and for each group we find $\hat{\theta}$. The s.d. of the estimated values, produced in each batch, can be used as an estimate of the Monte Carlo Error (k and ν should be large enough). This error should be small (e.g. 0.001); if it is not we need to run the chain for a larger number of iterations.
- To check the convergence of our algorithm we can also produce **multiple chains** (starting from different initial values) and see if the estimated values of the quantity of interest agree.

The Independent Metropolis-Hastings Algorithm

- Suppose that $q(y|x) = g(y)$, i.e. q is independent of the present state of the chain. Then we have the **Independent Metropolis-Hastings Algorithm**.
- This method then appears as a straightforward generalization of the *Rejection sampling*. In such cases g should have the same (or larger) support with f (if not the chain is not $f-$ irreducible).
- Choose g in such a way that $\bar{\rho}$ is large (e.g. 0.9).
- Even though the proposed values y are generated independently, the final accepted moves are not (since they depend on ρ, which depends on the current state of the chain).
- The M-H sample will involve repeated occurrences of the same value since rejection of the proposed (in contrast with the Rejection sampling). Finally you do not need to find the upper bound M (envelope) as in Rejection sampling.

The Random Walk Metropolis-Hastings Algorithm

- Idea: *Local exploration* of the neighborhood of the current value of the Markov chain.
- Simulate Y_t according to

$$Y_t = X^{(t)} + \epsilon_t,$$

where ϵ_t is a r.v. with distribution g, symmetric around zero and independent of $X^{(t)}$. For example if $\epsilon_t \sim U(-\delta, \delta)$ then $Y_t|X^{(t)} \sim U(X^{(t)} - \delta, X^{(t)} + \delta)$, while if $\epsilon_t \sim N(0, \tau^2)$ then $Y_t|X^{(t)} \sim N(X^{(t)}, \tau^2)$. This is called **Random Walk Metropolis-Hastings Algorithm**.

- Because of symmetry $q(y|x) = q(x|y)$. Therefore in the M-H algorithm the acceptance ratio becomes

$$\rho(x, y) = min\{1, f(y)/f(x)\}$$

- The Markov chain associated with q is a random walk, but due to the additional M-H acceptance step, the produced Markov chain is not.
- The algorithm will always accept moves that lead to a higher density function f, but sometimes will move "downhills".

The Random Walk Metropolis-Hastings Algorithm (cont'd)

- Choose the variance of q such that $\bar{\rho} \approx 0.25$. In general small values of the variance of q, will result in high acceptance rates but slow convergence since the algorithm will need a large number of iterations to explore the state space. In this case, large autocorrelations will appear in the output analysis. On the other hand, high values of the variance of q will result in low acceptance rates. As a consequence, for a large number of iterations the algorithm will stick with the same values, again resulting in poor exploration of the state space and a highly autocorrelated sample.

Gibbs Sampling

- What if we want to sample from a joint pdf $f_{\mathbf{X}}$ of a random vector $\mathbf{X} = (X_1, X_2, \ldots, X_p)$? We can use the (Multistage) **Gibbs Sampling**.

Algorithm **The Multistage Gibbs Sampler**

At iteration $t = 1, 2, \ldots,$ given $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_p^{(t)})$, generate

1. $X_1^{(t+1)} \sim f_1(x_1 | x_2^{(t)}, \ldots, x_p^{(t)})$;
2. $X_2^{(t+1)} \sim f_2(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots, x_p^{(t)})$;

$\vdots$

p. $X_p^{(t+1)} \sim f_p(x_p | x_1^{(t+1)}, \ldots, x_{p-1}^{(t+1)})$.

- We start the algorithm with an initial value: $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_p^{(0)}) \in \mathcal{X}$.

Gibbs Sampling (cont'd)

- The densities $f_1, f_2, \ldots, f_p$ are called **full conditionals** and are the only densities used for simulation. Thus, even in high-dimensional problems, all of the simulations are *univariate*!
- To find a full conditional f_i we merely need to pick all of the terms in the joint pdf $f_{\mathbf{X}}$ that involve x_i. Then you need to find the normalizing constant to convert it to a pdf.
- If you cannot simulate directly from a full conditional f_i, *Adaptive Rejection Sampling* can be used, or *Metropolis-Hastings* algorithm (**Metropolis within Gibbs**).

Gibbs Sampling (cont'd)

- Gibbs Sampler can be seen as a special case of **Single Component Metropolis-Hastings**. In Single Component M-H the vector $\mathbf{x}$ is divided into the univariate subvectors (components) $(x_1, x_2, \ldots, x_p)$ that are updated sequentially using Metropolis-Hastings steps with target distribution the full conditionals.
- Therefore at iteration $(t+1)$ we first update X_1, using a M-H algorithm with target f_1, a proposal q_1 and the current values $(x_1^{(t)}, x_2^{(t)}, \ldots, x_p^{(t)})$. On the second stage we update X_2, using a M-H algorithm with target f_2, a proposal q_2 and the current values $(x_1^{(t+1)}, x_2^{(t)}, \ldots, x_p^{(t)})$. We continue like this, until, finally in the last stage we update X_p, using a M-H algorithm with target f_p, a proposal q_p and the current values $(x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_p^{(t)})$. If in the place of $q_1, q_2, \ldots, q_p$, in each stage, we put the full conditionals $f_1, f_2, \ldots, f_p$ (and therefore in the j stage the proposal for X_j depends only on the current values of all the other components but not on the current value of X_j) then the acceptance ratios are all equal to 1 (in each M-H algorithm) and we have the Gibbs Sampler.