

STA 360/602L: MODULE 3.8

MCMC AND GIBBS SAMPLING II

DR. OLANREWAJU MICHAEL AKANDE

EXAMPLE: BIVARIATE NORMAL

- Consider

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right]$$

where ρ is known (and is the correlation between θ_1 and θ_2).

- We will review details of the multivariate normal distribution very soon but for now, let's use this example to explore Gibbs sampling.
- For this density, turns out that we have

$$\theta_1 | \theta_2 \sim \mathcal{N}(\rho\theta_2, 1 - \rho^2)$$

and

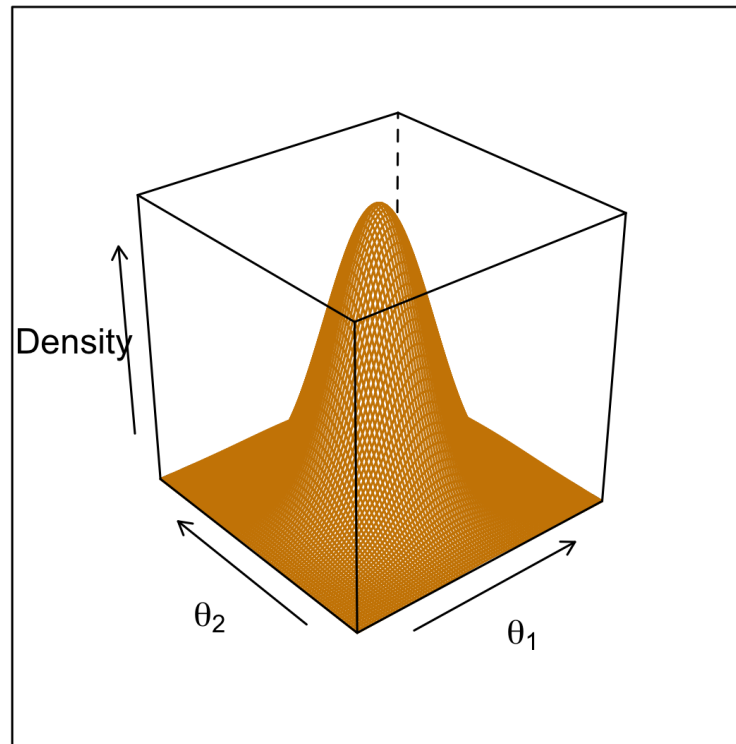
$$\theta_2 | \theta_1 \sim \mathcal{N}(\rho\theta_1, 1 - \rho^2)$$

- While we can easily sample directly from this distribution (using the `mvtnorm` or `MASS` packages in R), let's instead use the Gibbs sampler to draw samples from it.

BIVARIATE NORMAL

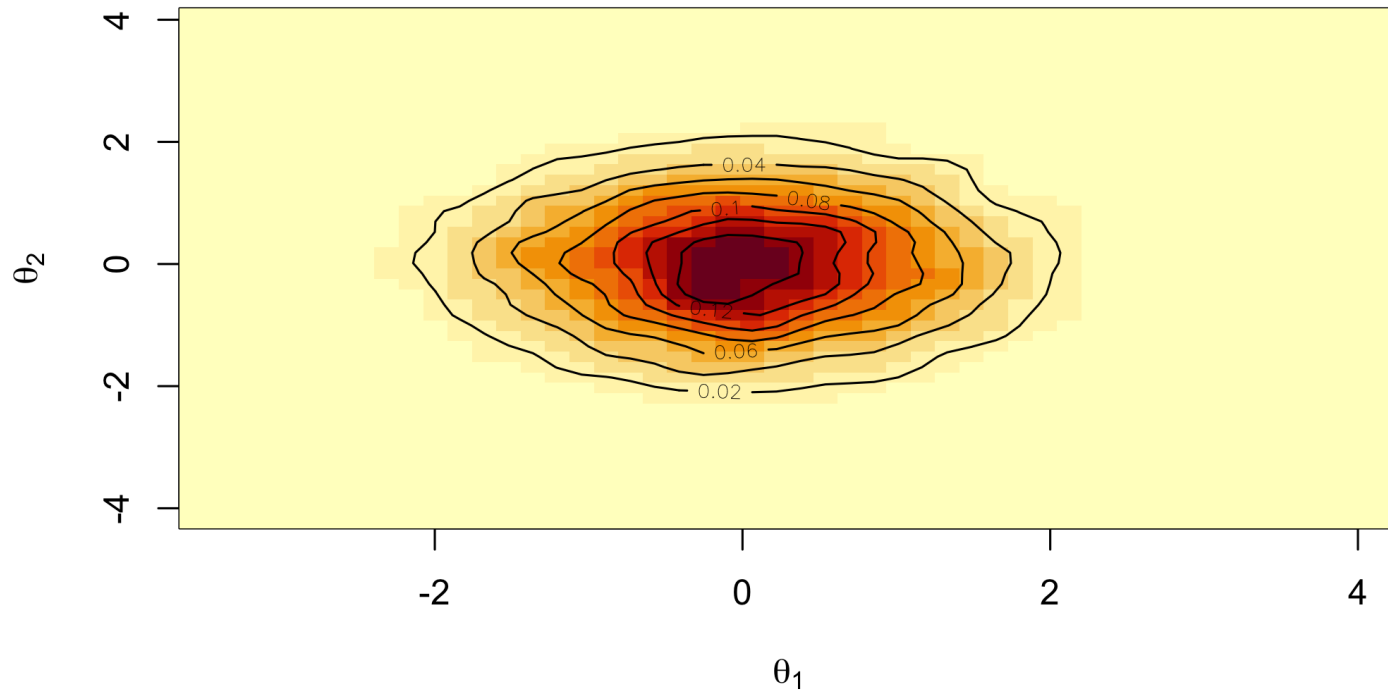
First, a few examples of the bivariate normal distribution.

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right]$$



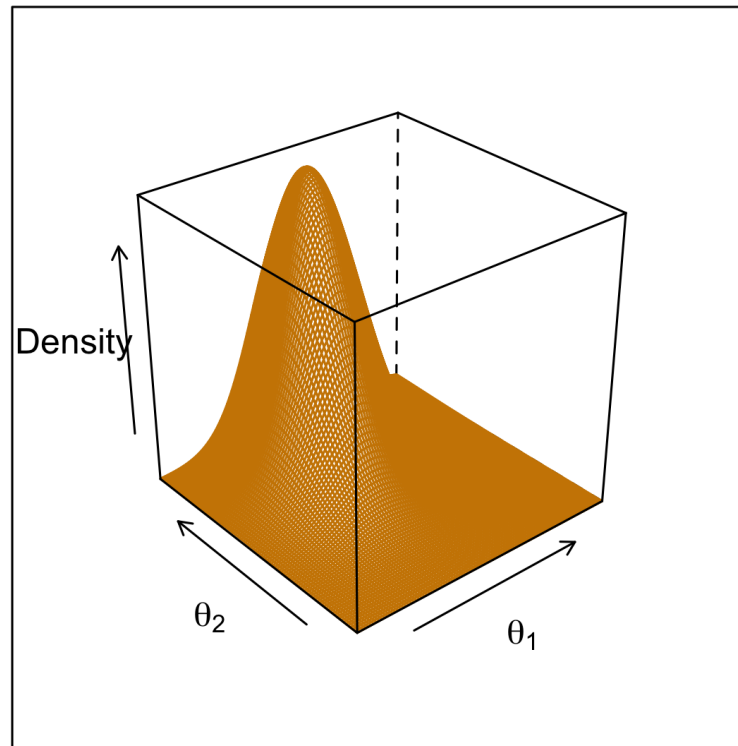
BIVARIATE NORMAL

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right]$$



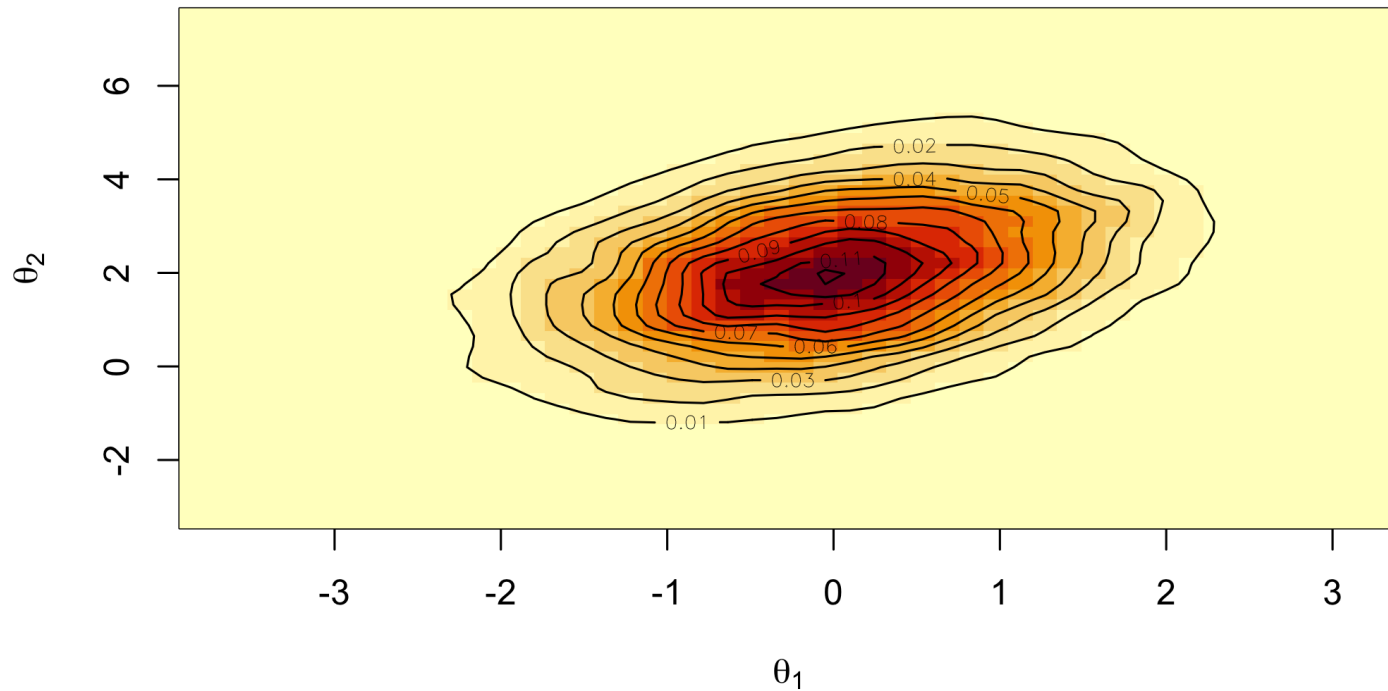
BIVARIATE NORMAL

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix} \right]$$



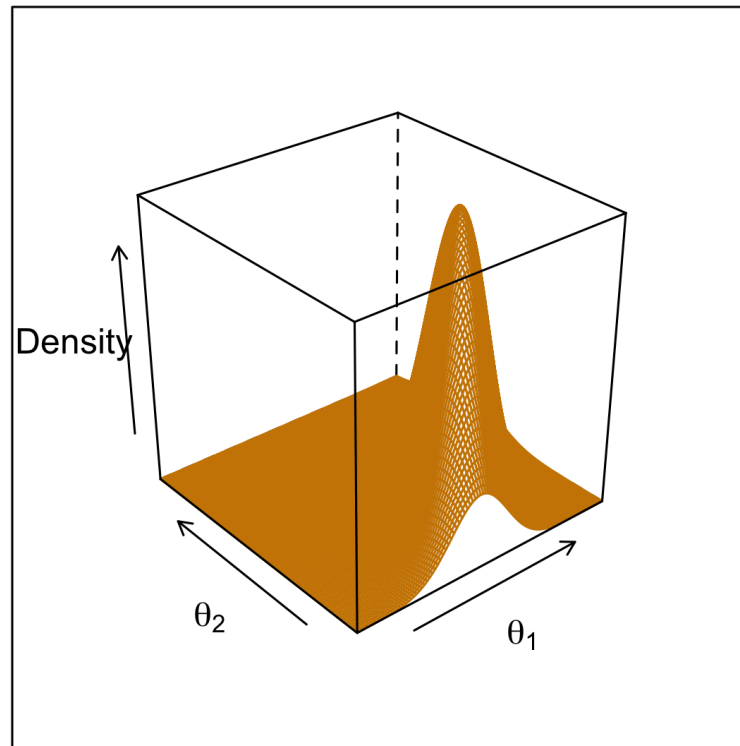
BIVARIATE NORMAL

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix} \right]$$



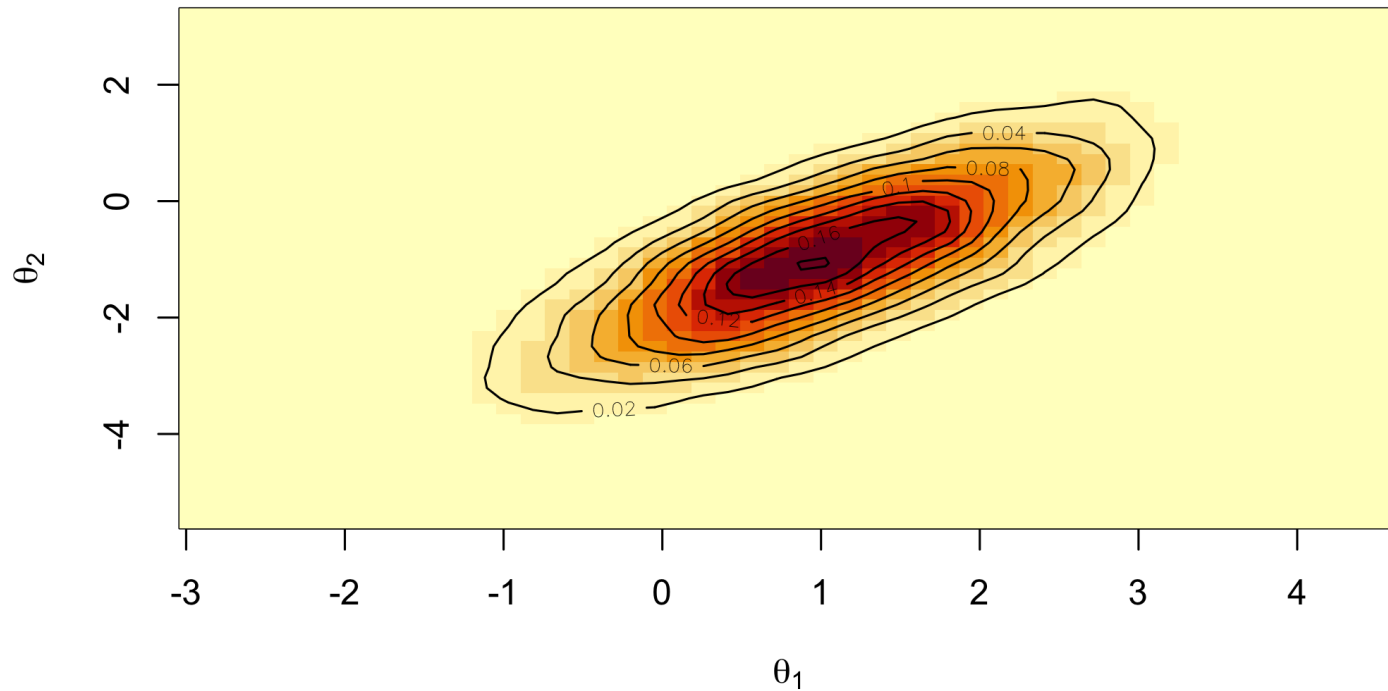
BIVARIATE NORMAL

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1.5 \end{pmatrix} \right]$$



BIVARIATE NORMAL

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1.5 \end{pmatrix} \right]$$



BACK TO THE EXAMPLE

- Again, we have

$$\theta_1|\theta_2 \sim \mathcal{N}(\rho\theta_2, 1 - \rho^2); \quad \theta_2|\theta_1 \sim \mathcal{N}(\rho\theta_1, 1 - \rho^2)$$

- Here's a code to do Gibbs sampling using those full conditionals:

```
rho <- #set correlation
S <- #set number of MCMC samples
thetamat <- matrix(0,nrow=S,ncol=2)
theta <- c(10,10) #initialize values of theta
for (s in 1:S) {
  theta[1] <- rnorm(1,rho*theta[2],sqrt(1-rho^2)) #sample theta1
  theta[2] <- rnorm(1,rho*theta[1],sqrt(1-rho^2)) #sample theta2
  thetamat[s,] <- theta
}
```

- Here's a code to do sample directly instead:

```
library(mvtnorm)
rho <- #set correlation; no need to set again once you've used previous code
S <- #set number of MCMC samples; no need to set again once you've used previous code
Mu <- c(0,0)
Sigma <- matrix(c(1,rho,rho,1),ncol=2)
thetamat_direct <- rmvnorm(S, mean = Mu,sigma = Sigma)
```

MORE CODE

See how the chain actually evolves with an overlay on the true density:

```
rho <- #set correlation
Sigma <- matrix(c(1,rho,rho,1),ncol=2); Mu <- c(0,0)
x.points <- seq(-3,3,length.out=100)
y.points <- x.points
z <- matrix(0,nrow=100,ncol=100)
for (i in 1:100) {
  for (j in 1:100) {
    z[i,j] <- dmvnorm(c(x.points[i],y.points[j]),mean=Mu,sigma=Sigma)
  }
}
contour(x.points,y.points,z,xlim=c(-3,10),ylim=c(-3,10),"orange2",
        xlab=expression(theta[1]),ylab=expression(theta[2]))

S <- #set number of MCMC samples;
thetamat <- matrix(0,nrow=S,ncol=2)
theta <- c(10,10)
points(x=theta[1],y=theta[2],col="black",pch=2)
for (s in 1:S) {
  theta[1] <- rnorm(1,rho*theta[2],sqrt(1-rho^2))
  theta[2] <- rnorm(1,rho*theta[1],sqrt(1-rho^2))
  thetamat[s,] <- theta
  if(s < 20){
    points(x=theta[1],y=theta[2],col="red4",pch=16); Sys.sleep(1)
  } else {
    points(x=theta[1],y=theta[2],col="green4",pch=16); Sys.sleep(0.1)
  }
}
```

MCMC

- Gibbs sampling is one of several flavors of **Markov chain Monte Carlo (MCMC)**.
 - **Markov chain**: a stochastic process in which future states are independent of past states conditional on the present state.
 - **Monte Carlo**: simulation.
- MCMC provides an approach for generating samples from posterior distributions.
- From these samples, we can obtain summaries (including summaries of functions) of the posterior distribution for θ , our parameter of interest.

HOW DOES MCMC WORK?

- Let $\theta^{(s)} = (\theta_1^{(s)}, \dots, \theta_p^{(s)})$ denote the value of the $p \times 1$ vector of parameters at iteration s .
- Let $\theta^{(0)}$ be an initial value used to start the chain (*should not be sensitive*).
- MCMC generates $\theta^{(s)}$ from a distribution that depends on the data and potentially on $\theta^{(s-1)}$, but not on $\theta^{(1)}, \dots, \theta^{(s-2)}$.
- This results in a Markov chain with **stationary distribution** $\pi(\theta|Y)$ under some conditions on the sampling distribution.
- The theory of Markov Chains (structure, convergence, reversibility, detailed balance, stationarity, etc) is well beyond the scope of this course so we will not dive into it.
- If you are interested, consider taking courses on stochastic process.

PROPERTIES

- **Note:** Our Markov chain is a collection of draws of θ that are (slightly we hope!) dependent on the previous draw.
- The chain will wander around our parameter space, only remembering where it had been in the last draw.
- We want to have our MCMC sample size, S , big enough so that we can
 - Move out of areas of low probability into regions of high probability (convergence)
 - Move between high probability regions (good mixing)
 - Know our Markov chain is stationary in time (the distribution of samples is the same for all samples, regardless of location in the chain)
- At the start of the sampling, the samples are **not** from the posterior distribution. It is necessary to discard the initial samples as a **burn-in** to allow convergence. We'll talk more about that in the next class.

DIFFERENT FLAVORS OF MCMC

- The most commonly used MCMC algorithms are:
 - Metropolis sampling (Metropolis et al., 1953).
 - Metropolis-Hastings (MH) (Hastings, 1970).
 - Gibbs sampling (Geman & Geman, 1984; Gelfand & Smith, 1990).
- Overview of Gibbs - Casella & George (1992, The American Statistician, 46, 167-174). the first two
- Overview of MH - Chib & Greenberg (1995, The American Statistician).
- We will get to Metropolis and Metropolis-Hastings later in the course.

WHAT'S NEXT?

MOVE ON TO THE READINGS FOR THE NEXT MODULE!