Gibbs Samplers

Jonathan Navarrete

July 2, 2017

Introduction

Now that we've familiarized ourselves with MCMC and the Metropolis-Hastings algorithms, we begin to analyze a now-common MCMC algorithm called Gibbs sampler. The Gibbs sampler is in fact a special case of the Metropolis-Hastings algorithm for high dimensional target distributions.

We introduce the Gibbs sampler with a two-stage example. The **two-state Gibbs sampler algorithm** as described Robert & Casella goes as follows

`Take` $X_0 = x_0$

For $t = 1, 2, ...,$ `generate`

1. $Y_t \sim f_{Y|X}(\cdot|x_{t-1})$
2. $X_t \sim f_{X|Y}(\cdot|y_t)$

The *two-stage* Gibbs sampler creates a Markov chain from a joint distribution in the following way. If two random variables X and Y have joint density $f(x, y)$, with corresponding conditional densities $f_{Y|X}$ and $f_{X|Y}$, the two stage Gibbs sampler generates a Markov chain (X_t, Y_t) by generating Y_t from conditional density $f_{Y|X}$ and then generating X_t from conditional density $f_{X|Y}$.

We illustrate the implementation of the Gibbs sampler with a simple example. Consider a bivariate Normal distribution where

$$X, Y \sim N_2(\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} \sigma_X^2 & \sigma_{XY}^2 \\ \sigma_{YX}^2 & \sigma_Y^2 \end{pmatrix})$$

The marginal distributions of X and Y are $N(\mu_X, \sigma_X)$ and $N(\mu_X, \sigma_X)$. The conditional distributions of Y and X are

$$Y|X = x \sim N(\mu_Y + \frac{\rho\sigma_Y}{\sigma_X}(x - \mu_X), (1 - \rho^2)\sigma_Y^2)$$

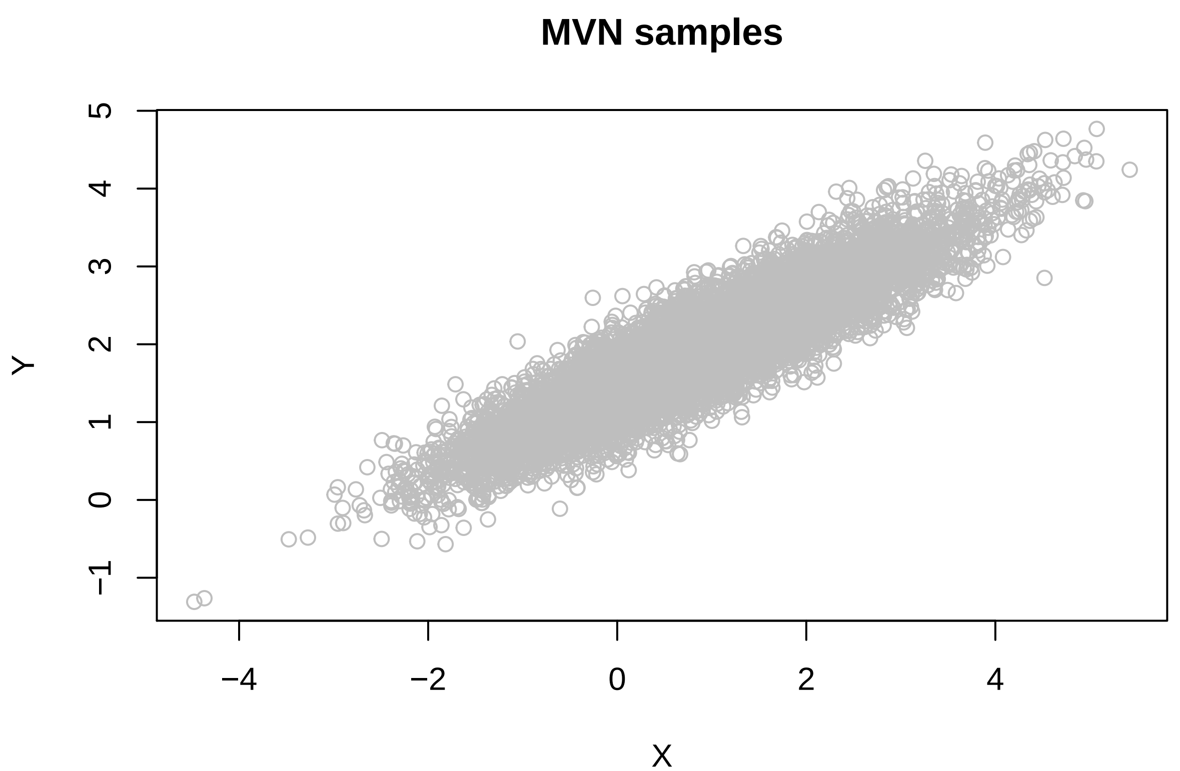
and

$$X|Y = y \sim N(\mu_X + \frac{\rho\sigma_X}{\sigma_Y}(y - \mu_Y), (1 - \rho^2)\sigma_X^2)$$

ρ is the correlation between X and Y, and $(1 - \rho^2)\sigma_X^2$ is the variance.

```
N = 10000

rho = 0.9
mu_x = 1
mu_y = 2
sd_x = 1.2
sd_y = 0.75

s1 = sqrt(1 - rho^2) * sd_x
s2 = sqrt(1 - rho^2) * sd_y

MVN = matrix(data = NA, nrow = N, ncol = 2,
           dimnames = list(NULL, c("X", "Y")))

MVN[1, ] = c(mu_x, mu_y)
```

```r
Y = MVN[1, 2] ## get Y vals
for(i in 1:(N-1)){
  mx = mu_x + rho * (Y - mu_y) * sd_x/sd_y
  X = rnorm(n = 1, mx, s1)
  MVN[i+1, 1] = X
  my = mu_y + rho * (X - mu_x) * sd_y/sd_x
  Y = rnorm(n = 1, mean = my, sd = s2)
  MVN[i+1, 2] = Y
}


plot(MVN, type = "p", col = 8,
     main = "MVN samples")
```




```r
## means
colMeans(MVN)
```

```
##         X         Y
## 0.9055049 1.9437401
```

```r
## correlation
cor(MVN)
```

```
##           X         Y
## X 1.0000000 0.9017219
## Y 0.9017219 1.0000000
```