STAT 37810 Final Project: Gibbs sampling in Python

Eileen Li

October 30, 2016

Introduction

Let $\mathbf{X}$ be a 2-dimensional random vector with exponential conditional densities

$$p_{X|Y}(x \mid y) \propto ye^{-yx}, \quad x \in (0, B) \tag{1}$$
$$p_{Y|X}(y \mid x) \propto xe^{-yx}, \quad y \in (0, B) \tag{2}$$

where B is a finite, positive constant. The normalizing constants are respectively $(1 - e^{-yB})^{-1}$ and $(1 - e^{-Bx})^{-1}$. We'll use Gibbs sampling to generate T samples from the marginal distribution $p_X(x)$. The scheme proceeds from the i^{th} sample to the $(i+1)^{\text{st}}$ sample as follows.

1. Given $\mathbf{X}^{(i)} = (X^{(i)}, Y^{(i)})$, condition on Y and generate $X^{(i+1)}$ from $p_{X|Y}(x \mid Y^{(i)})$.
2. Condition on X and generate $Y^{(i+1)}$ from $p_{Y|X}(y \mid X^{(i+1)})$.
3. Set $\mathbf{X}^{(i+1)} = (X^{(i+1)}, Y^{(i+1)})$.

To sample from (1) and (2), we'll use inverse transform sampling. Namely, the conditional CDF of (1) is

$$F_{X|Y}(x \mid y) \propto \int_0^x ye^{-yt}\, \mathrm{d}t = 1 - e^{-yx}, \tag{3}$$

whose inverse can easily computed as

$$F^{-1}_{X|Y}(u \mid y) = \frac{1}{y} \log\left(\frac{1}{1 - u(1 - e^{-yB})}\right). \tag{4}$$

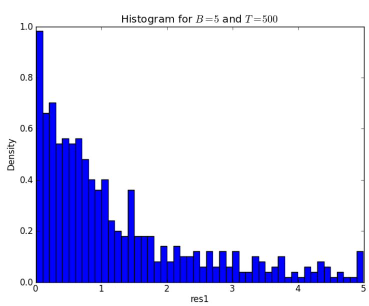
As shown in class, if $U \sim \text{Unif}(0, 1)$ then $F^{-1}(U) \sim F_{X|Y}(x \mid y)$. Furthermore, since (1) and (2) are symmetric, we only need to switch the roles of x and y in (3) and (4) when sampling from (2).

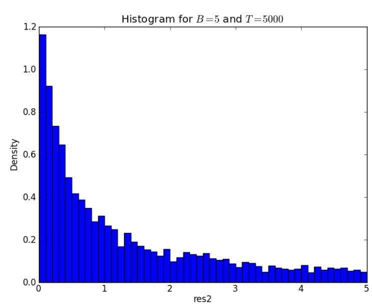
Code

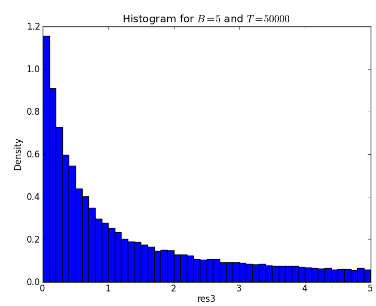
```
import random, math, numpy, matplotlib.pyplot as plt, matplotlib.mlab as mlab

def inv(u, z, B): # Inverse CDF (equation 4)
    return -math.log(1-u*(1-math.exp(-z*B))) / z

def gibbs(T, B):
    res = numpy.empty(shape = (T, 2))
    y = numpy.random.uniform(0, B)
    for i in range(T):
        x = inv(numpy.random.uniform(0, 1), y, B)
        y = inv(numpy.random.uniform(0, 1), x, B)
        res[i, ] = [x, y]
    return res
```

```
# T=500 histogram
res1 = gibbs(500, 5)
plt.hist(res1[:, 0], normed=True, bins=50)
plt.xlabel('res1')
plt.ylabel('Density')
plt.title(r'Histogram for $B=5$ and $T=500$')
plt.show()

# T=5000 histogram
res2 = gibbs(5000, 5)
plt.hist(res2[:, 0], normed=True, bins=50)
plt.xlabel('res2')
plt.ylabel('Density')
plt.title(r'Histogram for $B=5$ and $T=5000$')
plt.show()

# T=50000 histogram
res3 = gibbs(50000, 5)
plt.hist(res3[:, 0], normed=True, bins=50)
plt.xlabel('res3')
plt.ylabel('Density')
plt.title(r'Histogram for $B=5$ and $T=50000$')
plt.show()

# Estimation of the first moment of X
print numpy.mean(res1[:, 0])
print numpy.mean(res2[:, 0])
print numpy.mean(res3[:, 0])
```

Results







The mean of X is 1.155 for $T = 500$, 1.260 for $T = 5000$, and 1.261 for $T = 50000$.