

**IE531: Algorithms for Data Analytics**  
**Spring, 2018**  
**Programming Assignment 4: Multivariate Gaussian RV**  
**Generator via Metropolis-Hasting and Gibbs Sampling**  
**Due Date: April 13, 2018**  
 ©Prof. R.S. Sreenivas

A **Multivariate Gaussian** RV generator has to generate i.i.d.  $(d \times 1)$  vectors  $\mathbf{x}$ , whose  $(d \times 1)$  mean is  $\boldsymbol{\mu}$  and whose  $(d \times d)$  covariance-matrix is  $\boldsymbol{\Sigma}$ . That is,

$$\mathbf{x} \sim \underbrace{\frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}}_{=f(\mathbf{x})} \quad (1)$$

We write the above expression in short-hand as  $\mathbf{x} \sim N(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ .

In this programming assignment we will write two C++ programs that can produce a set of vectors  $\{\mathbf{x}_i\}_{i=1}^{\infty}$ , where  $\mathbf{x}_i \sim N(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ . The approach that uses the *Cholesky decomposition* of  $\boldsymbol{\Sigma}$  (see section 1.2.1 of *Probability and Statistics Review*, for example) would not work if  $d$  is very large. We will cover two (of many) approaches to overcome this issue that uses concepts from the theory of *Markov-Chain Monte Carlo* (MCMC) methods. The first approach uses the *Metropolis-Hasting Algorithm*; the second approach uses *Gibbs Sampling*.

## 1 Part 1: Metropolis-Hasting Algorithm

The *Proposal Distribution* is the  $d$ -dimensional Multivariate Gaussian with zero-mean (i.e.  $\boldsymbol{\mu} = \mathbf{0}$ ) and Unit-Covariance (i.e.  $\boldsymbol{\Sigma} = \mathbf{I}$ ). Assume that for right now we have generated  $\{\mathbf{x}_i\}_{i=1}^j$ , where  $\mathbf{x}_i \sim N(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ . We can generate RVs  $\hat{\mathbf{x}} \sim N(\mathbf{x}, \mathbf{x}_j, \mathbf{I})$  as  $\hat{\mathbf{x}} = \mathbf{x}_j + \mathbf{y}$ , where  $\mathbf{y} \sim N(\mathbf{y}, \mathbf{0}, \mathbf{I})$  (which can be generated by  $d$ -many calls to `get_gaussian()`).

Since the Proposal Distribution is symmetric, following equation 2 of section 4 of my notes on the material in Chapter 4 of the text, we accept  $\hat{\mathbf{x}}$  with probability

$$p(\hat{\mathbf{x}}, \mathbf{x}_j) = \min \left\{ 1, \frac{f(\hat{\mathbf{x}})}{f(\mathbf{x}_j)} \right\} = \min \left\{ 1, \frac{e^{-\frac{1}{2}(\hat{\mathbf{x}}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\hat{\mathbf{x}}-\boldsymbol{\mu})}}{e^{-\frac{1}{2}(\mathbf{x}_j-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_j-\boldsymbol{\mu})}} \right\}. \quad (2)$$

If  $\hat{\mathbf{x}}$  is rejected, the process is repeated with no change. If  $\hat{\mathbf{x}}$  is accepted, then  $\{\mathbf{x}_i\}_{i=1}^{j+1} \leftarrow \{\mathbf{x}_i\}_{i=1}^j \cup \{\hat{\mathbf{x}}\}$ , and the process repeats with  $j \leftarrow (j+1)$ . That is, we present  $\hat{\mathbf{x}}$  as the  $(j+1)$ -th RV  $\mathbf{x}_{j+1} \sim N(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ .

## 2 Part 2: Gibbs Sampling

For this part, you are going to use **Gibbs Sampling** to generate Multivariate Gaussian RVs. The C++ STL has a **Univariate Gaussian Generator**. That is,

it can generate i.i.d. scalars  $x \sim N(\mu, \sigma)$ , with mean  $\mu$  and standard-deviation  $\sigma$ .

The following [web page](#)<sup>1</sup> derives the marginal- and conditional distributions of a Multivariate Gaussian Distribution. Suppose the  $(d \times 1)$  vector  $\mathbf{x}$  is partitioned as

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}$$

where  $\mathbf{x}_1$  is  $p \times 1$   $\mathbf{x}_2$  is  $q \times 1$  where  $p + q = d$ , where

$$\mathbf{x} \sim N(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}); \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}; \text{ and } \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}.$$

It follows that  $\boldsymbol{\Sigma}^T = \boldsymbol{\Sigma}$ . Then, it is known that the conditional distribution of  $\mathbf{x}_i$  given the values of  $\mathbf{x}_j$  is also Normally distributed

$$(\mathbf{x}_i | \mathbf{x}_j) \sim N(\mathbf{x}_i, (\boldsymbol{\mu}_i | \boldsymbol{\mu}_j), (\boldsymbol{\Sigma}_i | \boldsymbol{\Sigma}_j))$$

where

$$(\boldsymbol{\mu}_i | \boldsymbol{\mu}_j) = \boldsymbol{\mu}_i + \boldsymbol{\Sigma}_{ij} \boldsymbol{\Sigma}_{jj}^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_j)$$

and

$$(\boldsymbol{\Sigma}_i | \boldsymbol{\Sigma}_j) = \boldsymbol{\Sigma}_{ii} - \boldsymbol{\Sigma}_{ji}^T \boldsymbol{\Sigma}_{jj}^{-1} \boldsymbol{\Sigma}_{ji}$$

In your implementation of a Multivariate Gaussian RV generator that uses Gibbs Sampling for this programming assignment, I ask that you assume  $p = d - 1$  and  $q = 1$  in the above interpretation. We know that  $(\mathbf{x}_2 | \mathbf{x}_1)$  is essentially a Univariate Gaussian (with known mean and variance; that can be computed using the above formula). By making a call to the C++ STL Univariate Gaussian RV generator, you can find a (random) value for  $\mathbf{x}_2$ . The following [YouTube video](#) describes the MCMC-part of the algorithm. Watch it before you start this programming assignment.

## The Programming Assignment

For the first part of the programming assignment you will write a Multivariate Gaussian Generator using the MCMC-MH algorithm. I have provided a hint for the first part on Compass. Although the code is meant to run for high-dimensions, I am verifying it for the case when  $d = 2$ . You will do the same. The code runs on command-line, the first variable is the number of trials, the second is the name of the CSV file that contains the experimental 2D-PDF, the third contains the theoretical 2D-PDF. You can plot the two figures in MATLAB, which is used in lieu of a formal verification of the correctness of the code. Figures 1 and 2 show a sample run on my Mac.

For the second part of the programming assignment you will write a Multivariate Gaussian Generator using Gibbs Sampling. This routine takes as input

---

<sup>1</sup>Save a small typo.

```
wirelessprv-10-194-149-149:Debug sreenivas$ time ./Multivariate\ Gaussian\ via\ Metropolis-Hastings 10000000 x y
Multivariate Gaussian Generator using MCMC-MH
Dimension = 2

Mean Vector =
1.000000
2.000000

Covariance Matrix =
1.000000 0.500000
0.500000 1.000000

real    14m2.742s
user    13m26.041s
sys      0m5.027s
wirelessprv-10-194-149-149:Debug sreenivas$
```

Figure 1: Sample run of the MCMC-MH based Multivariate Gaussian RV Generator;  $d = 2$  for this illustration;  $\text{no\_of\_trials} = 10,000,000$ .

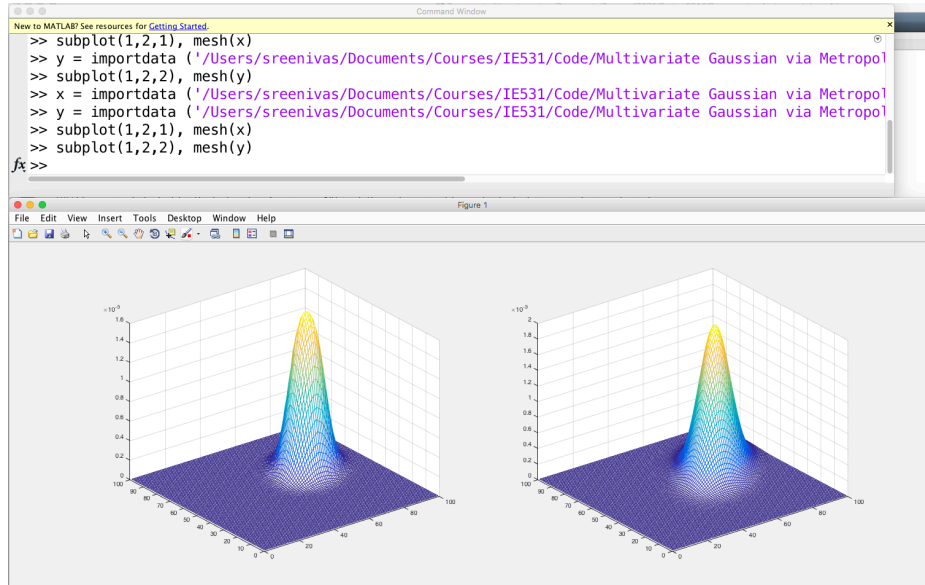


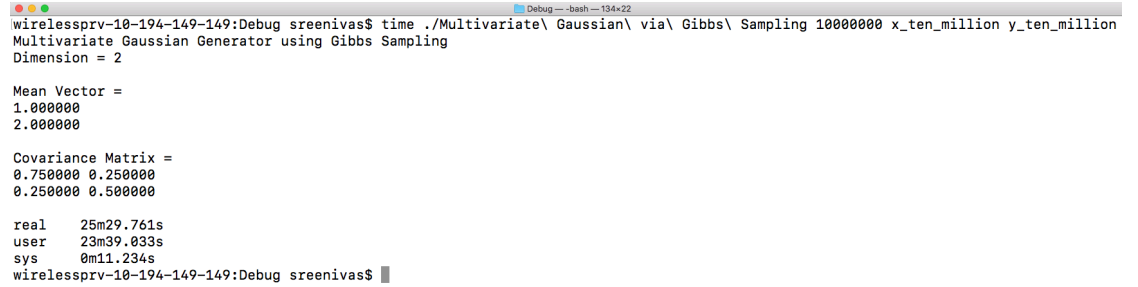
Figure 2: A comparison of the experimentally observed PDF/histogram plot (on the left) vs. the theoretical PDF (on the right) for the trial shown in figure 1 ( $\text{no\_of\_trials} = 10,000,000$ ).

the previous sample (i.e. `Previous_value`); and returns a  $d$ -dimensional Multivariate Gaussian RV  $\mathbf{x}$ , where

$$\mathbf{x}_i = \begin{cases} \text{Previous\_value}_i & \text{if } i \neq \text{index} \\ \sim N(\mu, \sigma) & \text{if } i = \text{index}; \text{ and } \mu \text{ and } \sigma \text{ are carefully selected.} \end{cases}$$

Keep in mind, for this to work properly you have to cycle-through all  $d$ -dimensions. That is,  $\text{index} \in \{1, 2, \dots, d\}$  in a cyclic-fashion.

In my `hint.cpp` I am verifying the correctness of the code by generating a 2D Multivariate Gaussian. The code should be self-explanatory. A sample run of the code is shown in figure 3. The experimental PDF is shown on the left of figure 4, while the theoretical PDF is shown on the right of figure 4. Your code should be verifiable for 2D (although it will be written for  $d$ -dimensions).



```
wirelessprv-10-194-149-149:Debug sreenivas$ time ./Multivariate Gaussian\ via\ Gibbs\ Sampling 10000000 x_ten_million y_ten_million
Multivariate Gaussian Generator using Gibbs Sampling
Dimension = 2

Mean Vector =
1.000000
2.000000

Covariance Matrix =
0.750000 0.250000
0.250000 0.500000

real    25m29.761s
user    23m39.033s
sys      0m11.234s
wirelessprv-10-194-149-149:Debug sreenivas$
```

Figure 3: Sample run of the Gibbs Sampling based Multivariate Gaussian RV Generator;  $d = 2$  for this illustration; `no_of_trials` = 10,000,000.

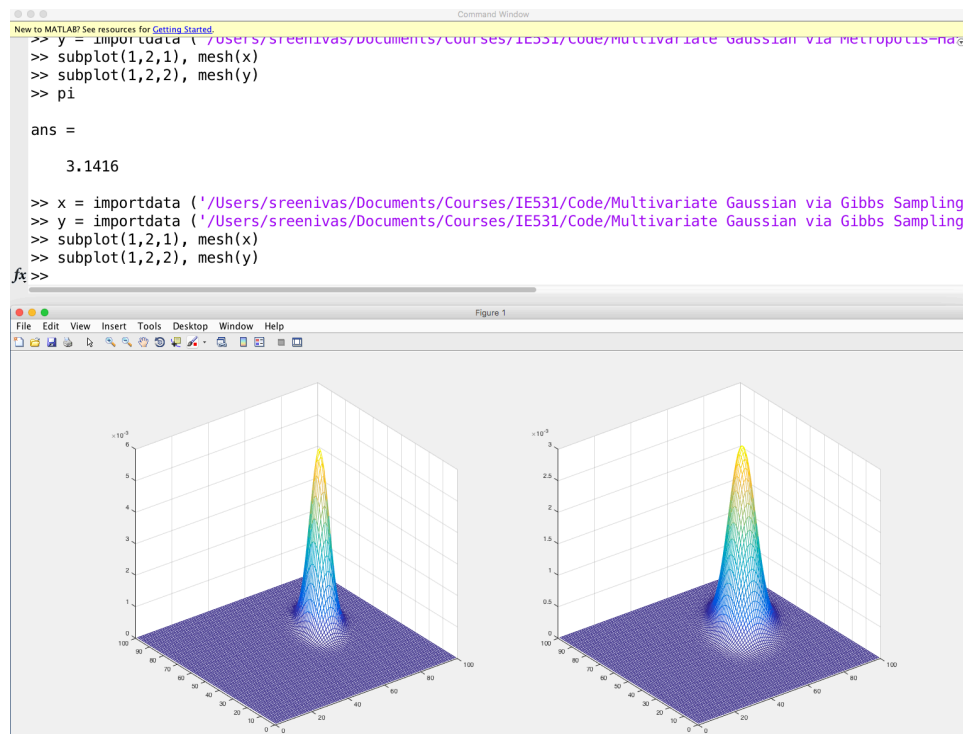


Figure 4: A comparison of the experimentally observed PDF/histogram plot (on the left) vs. the theoretical PDF (on the right) for the trial shown in figure 1 (no\_of\_trials = 10,000,000).