Data Driven Sampling (Lecture 3): Sample random variable on computer

Hongwei YUAN

University of Macau

Definition (Uniform distribution)

If $\theta_1 < \theta_2$, a random variable X is said to have a continuous uniform probability distribution on the interval (θ_1, θ_2) if and only if the density function of Y is

$$f(x) = \begin{cases} \frac{1}{\theta_2 - \theta_1}, & \theta_1 \leq x \leq \theta_2, \\ 0, & elsewhere. \end{cases}$$

We denote this distribution by $\mathcal{U}(\theta_1, \theta_2)$.

- In this lecture, we will often use the uniform distribution $\mathcal{U}(0, 1)$.

Let X be a random variable, its cumulative distribution function (abbreviated as cdf) is

$$F(x) = \mathbb{P}(X \leq x), \qquad x \in \mathbb{R}.$$

Theorem

If cdf F is a strictly increasing and continuous function, then the random variable $F(X)$ is a $\mathcal{U}(0,1)$ distributed random variable.

- By the definition $F(x) = \mathbb{P}(X \leq x)$, we know that F is a nondecreasing function, i.e. for any $x_1 < x_2$, we have $F(x_1) \leq F(x_2)$.
- In the above theorem, we assume that F is strictly increasing, i.e. for any $x_1 < x_2$, we have $F(x_1) < F(x_2)$. Hence, we know F has its inverse function F^{-1}.

Proof.

Since F is strictly increasing and continuous, it has its inverse function F^{-1}, for any $u \in (0,1)$, there exists a unique $x \in \mathbb{R}$ so that $u = F(x)$. Thus,

$$\mathbb{P}(F(X) \leq u) = \mathbb{P}(F(X) \leq F(x)) = \mathbb{P}(F^{-1}(F(X)) \leq F^{-1}(F(x)))$$

$$= \mathbb{P}(X \leq x) = F(x) = u.$$

$\square$

Theorem

If cdf F is strictly increasing and continuous, and U is a $\mathcal{U}(0,1)$-distributed random variable, then the random variable $F^{-1}(U)$ has a cdf F.

Proof.

For any $x \in \mathbb{R}$, we have

$$\mathbb{P}(X \leq x) = \mathbb{P}(F^{-1}(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x).$$

□

- This theorem is actually true for any cdf.

For an arbitrary cdf F, define its general inverse:

$$F^{-}(u) = \inf\{x : F(x) \geq u\}, \qquad u \in [0,1].$$

Theorem

Given a cdf F, then the random variable X defined by $X = F^{-}(U)$ has a cdf F.

The proof is not required!

Assignment: Verify that the theorem is true for the Binomial distribution $Bin(4, 1/2)$.

Example

Given $Exp(1)$ distribution's cdf $F(x) = 1 - e^{-x}$. The inverse function of F is $F^{-1}(u) = -\log(1-u)$ for all $u \in [0,1]$. By Theorem 4, if $U \sim \mathcal{U}(0,1)$, then we know

$$F^{-1}(U) \sim Exp(1).$$

Since $1 - U \sim \mathcal{U}(0,1)$, then

$$F^{-1}(1-U) = -\log(U) \sim Exp(1).$$

Sampling a random variable on computer

- From the python program, we see that **a random variable on computer** is actually **a random number**.
- In the practice, we often need to generate many random numbers (e.g. 1000 random numbers).
- Create a histogram of these random numbers to visualize these random variables.

Sampling a random variable on computer

The following is a Python program (red sentences) for generating one random variable $U \sim \mathcal{U}(0,1)$

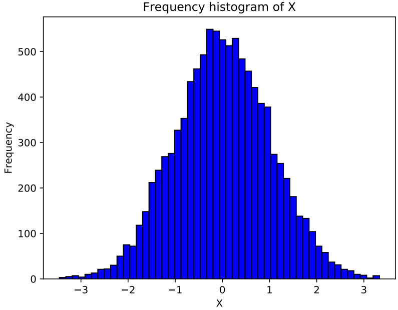
- import numpy as np   #import the python library 'numpy'
- left,right,size=0,1,1   # three parameters of uniform distribution
- X=np.random.uniform(left,right,size)   #generate a $\mathcal{U}(0,1)$ random variables
- print(X)   #print the random variable X

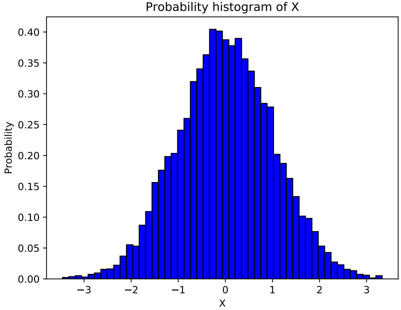
Assignment: Try this code by yourself and choose different parameters:left, right, size

Frequency histogram and probability histogram of 10000 standard normal distribution $N(0,1)$ random variables

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# Generate 10000 standard normal random variables and store them in X
mu,sigma,size=0,1,10000
X = pd.Series(np.random.normal(mu,sigma,size))
# Create frequency histogram of these 10000 random variables.
bins=50
n,bins,patches  = plt.hist(X,bins, facecolor='blue', edgecolor='black')
plt.title('Frequency histogram of X')
plt.xlabel('X')
plt.ylabel('Frequency')
plt.show()
# Create probability histogram of these 10000 random variables.
bins=50
n,bins,patches  = plt.hist(X,bins, facecolor='blue', edgecolor='black',density='true')
plt.title('Probability histogram of X')
plt.xlabel('X')
plt.ylabel('Probability')
plt.show()
```

Frequency histogram and probability histogram of 10000 standard normal distribution $N(0,1)$ random variables

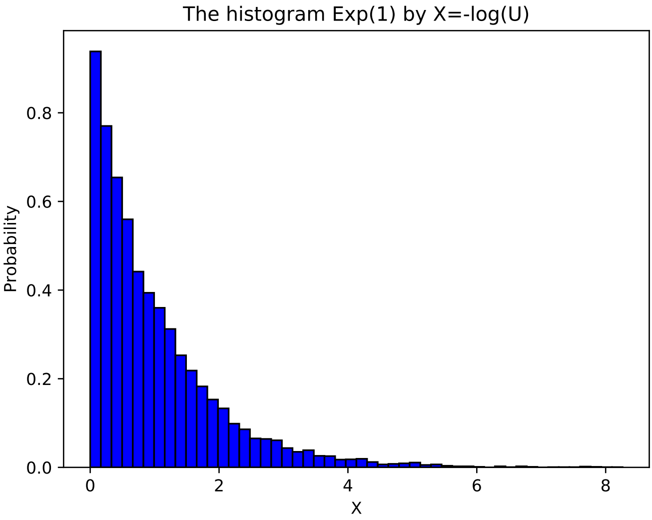




Python Program for Example 5

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# Generate 10000 standard normal random variables and store them in X
left,right,size=0,1,10000
U = pd.Series(np.random.uniform(left,right,size))
X=-np.log(U)
# Create probability histogram of Exp(1)
bins=50
n,bins,patches  = plt.hist(X,bins, facecolor='blue', edgecolor='black',density='true')
plt.title('The histogram Exp(1) by X=-log(U)')
plt.xlabel('X')
plt.ylabel('Probability')
plt.savefig("EPHist.pdf")
plt.show()
```

Probability histogram of Exp(1) in Example 5



Assignment: Run two codes on your own computer and change the parameters therein.

We have learned how to sample a **one** dimensional random variable for a given cdf F, in the following way:

$$X = F^{-}(U)$$

where $U \sim \mathcal{U}(0,1)$ and

$$F^{-}(u) = \inf\{x : F(x) \geq u\}, \qquad u \in [0,1].$$

Problem: This method only works for **one** dimensional random variable!

Question: How to sample a **multi-dimensional** random variable?

Accept-reject method

- We know a (possibly multi-dimensional) distribution whose probability density function (abbreviated as pdf) is known as f. We shall use accept-reject method to generate a random variable X (by computer) with this distribution.
- The basic idea of accept-reject method is as the following:
  - first generate a random variable Y which has a pdf g and can be easily sampled;
  - then compare U and $f(Y)/g(Y)$.

Accept-reject method

Assumption: Let the pdfs f and g satisfy the following property: there exists a constant $M > 0$ such that

$$\frac{f(x)}{g(x)} \leq M, \quad \forall\, x.$$

Algorithm: Accept-reject method

1. Sample a random variable $Y \sim g$, $U \sim \mathcal{U}(0,1)$;
2. If $U \leq \frac{f(Y)}{Mg(Y)}$, **accept**, i.e., take $X = Y$ and stop;
3. If $U > \frac{f(Y)}{Mg(Y)}$, **reject**, i.e., do nothing but return to step 1.

Accept-reject method

Theorem

Let f and g satisfy the assumption in the previous slide, then the random variable produced by the algorithm in the previous slide has a distribution with the density f.

Proof.

We only show the theorem for one dimensional case. It suffices to show that

$$\mathbb{P}\left(Y \leq x \middle| U \leq \frac{f(Y)}{Mg(Y)}\right) = \mathbb{P}(X \leq x), \quad \forall \ \ x \in \mathbb{R}.$$

Let us compute the conditional probability on the left hand. □

Accept-reject method

$$\mathbb{P}\left(Y \le x \middle| U \le \frac{f(Y)}{Mg(Y)}\right) = \frac{\mathbb{P}(Y \le x, U \le \frac{f(Y)}{Mg(Y)})}{\mathbb{P}(U \le \frac{f(Y)}{Mg(Y)})}$$

$$= \frac{\int_{-\infty}^{x}\left(\int_0^{\frac{f(y)}{Mg(y)}} \mathrm{d}u\right) g(y)\mathrm{d}y}{\int_{-\infty}^{\infty}\left(\int_0^{\frac{f(y)}{Mg(y)}} \mathrm{d}u\right) g(y)\mathrm{d}y} = \frac{\int_{-\infty}^{x} \frac{f(y)}{Mg(y)} g(y)\mathrm{d}y}{\int_{-\infty}^{\infty} \frac{f(y)}{Mg(y)} g(y)\mathrm{d}y}$$

$$= \frac{\int_{-\infty}^{x} f(y)\mathrm{d}y}{\int_{-\infty}^{\infty} f(y)\mathrm{d}y} = \int_{-\infty}^{x} f(y)\mathrm{d}y = \mathbb{P}(X \le x).$$

Example: Generate Gamma distribution

- Gamma(α, β) distribution density: $f(x) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}$ for $x \geq 0$;
- When $\alpha = 1$, it is an Exp$(1/\beta)$ distribution, whose random variable can be generated by $X = -\beta \log U$;
- When $\alpha \neq 1$, we cannot use the general inverse method because the inverse function often does not have an explicit form.
- When $\alpha \neq 1$, we can use an accept-reject method.

Example: Generate Gamma distribution

Let $\alpha = 2.5$ and $\beta = 2$, then

$$f(x) = \frac{2^{2.5} x^{1.5} e^{-2x}}{\Gamma(2.5)}.$$

We use Exp(1) distribution as a reference, i.e.,

$$g(x) = e^{-x}.$$

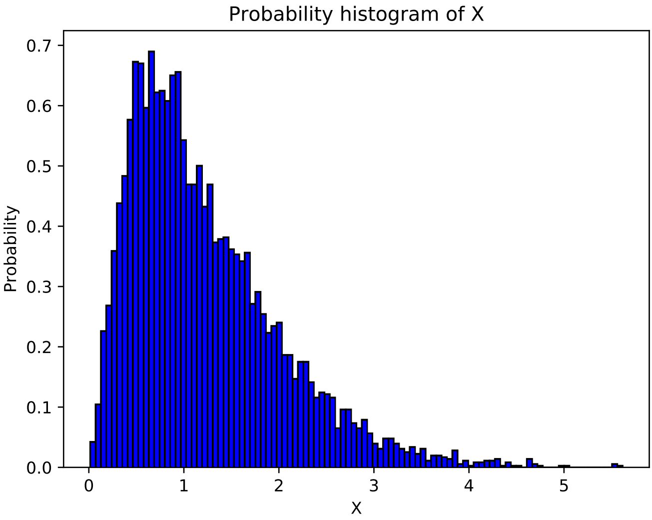
It is easily see that

$$\frac{f(x)}{g(x)} = \frac{2^{2.5} x^{1.5} e^{-x}}{\Gamma(2.5)} \leq 4$$

Example: Generate Gamma distribution

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
X = [ ]
size=10000
Y = np.random.exponential(1,size)
U = np.random.uniform(0,1,size)
a = 2**(2.5)/math.gamma(2.5)
f = a*(Y**1.5)*np.exp(-2*Y)
g = np.exp(-Y)
M = 4
for i in range(size):
    if U[i]<=f[i]/(M*g[i]):
       X.append(Y[i])
# Create histogram of these 10000 random variables.
# Create probability histogram of these 10000 random variables.
bins=100
n,bins,patches  = plt.hist(X,bins, facecolor='blue', edgecolor='black',density='true')
plt.title('Probability histogram of X')
plt.xlabel('X')
plt.ylabel('Probability')
plt.savefig("NPHist.pdf")
plt.show()
```

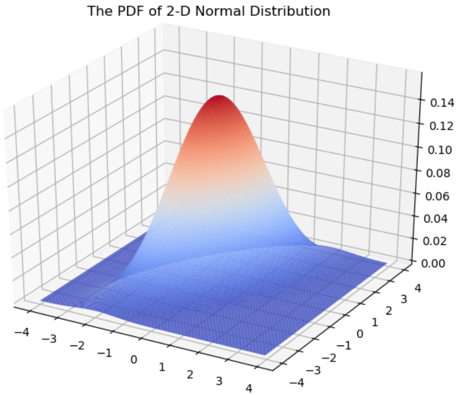
Example: Generate Gamma distribution



Example: 2D normal distribution

```
import matplotlib.pyplot as plt
import numpy
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm  #### get color map
import numpy as np
from scipy.stats import multivariate_normal as mvnorm

#### define the range of axises ####
x, y = np.mgrid[-4:4:.05, -4:4:.05]
pos = np.dstack((x, y))

#### define 2-D normal rv ####
mean = np.array([0,0])
cov  = np.array([[1,1],[1,2]])

rv = mvnorm(mean,cov)
Y = rv.pdf(pos)

#### define 3D figure ####
fig = plt.figure()
ax = Axes3D(fig)
ax.plot_surface(pos[:,:,0], pos[:,:,1], Y, rstride = 1, cstride = 1, cmap = cm.coolwarm)
fig.suptitle("The PDF of 2-D Normal Distribution")
plt.savefig('2D Norm')
plt.show()
```

Example: 2D normal distribution



Assignment: Use the accept-reject method to make a python program which will draw a probability histogram of this 2D normal distribution.