3 Rec 3: Common Probability Distributions

Directions: Your instructor will spend the the first 40 minutes of the recitation period working some review problems and going over one or more Matlab experiments in the following. During the last 10 minutes of recitation, your proctor will give you a "Lab Form" that your recitation team completes, signs, and turns in. See the last page for an indication of what you will be asked to do on the Lab Form.

Due to time limitations, only a part of the following can be covered during the recitation period. However, you might want in the future to try some of the uncovered experiments on your own. They could give skills useful on some future homework problems and could lend insight into your understanding of the course from an experimental point of view.

This Week's Topics.

- Estimating PDF/CDF from Data
- Simulating a Nonstandard Uniform Random Variable
- Simulating a Nonstandard Gaussian Random Variable
- Simulating a Binomial Random Variable
- Simulating a Poisson Random Variable
- Simulating an Exponential Random Variable

3.1 Exp 1: Estimating PDF/CDF from Data

Suppose you have a vector of many observations from a discrete distribution. You have seen in Recitations 1 and 2 how you can directly use the histogram function to get an estimate of the PMF (probability mass function) of the discrete distribution. If the vector instead contains observations from a continuous probability distribution, it is a bit more tricky to use the histogram function to get an estimate of the PDF (probability density function). This experiment tells you how to go about this.

Suppose we have a vector `x` of observations from a continuous probability distribution with density function (PDF) $f(x)$. This experiment teaches you how we can get an estimated plot of $f(x)$. Let's suppose that `x` consists of n samples, n large. (We will typically take $n = 100000$ in our experiments.) We then subdivide the range of `x` values into N bins of equal width. (In our experiments, we typically take $N = n/100$, so that if the data were uniformly distributed, we would expect about 1000 samples in each bin.) Let Δ be the bin width. Let x_i be the midpoint of the i-th bin and let n_i be the number of samples in the i-th bin (which can be found with the Matlab function "`hist`"). Then

$$n_i/n \approx f(x_i)\Delta.$$

(The left side is an empirical estimate, based on the n observations, of the probability that a data value will fall in the given bin; the right side is the approximate area captured by this bin under the density function.) From this, we obtain

$$f(x_i) \approx n_i/(n\Delta)$$

as an estimate for $f(x_i)$. We can implement this idea with the following Matlab code:

```
n=length(x); %n is the number of samples in x
N=floor(n/100); %N is the number of bins
A=min(x); B=max(x); %[A,B] is the range of x values
Delta=(B-A)/N; %Delta is the bin width
t=A-Delta/2+[1:N]*Delta; %horizontal axis of bin midpoints
f=hist(x,t)/(Delta*n); %vertical axis of density estimates
bar(t,f); %estimated density plot
```

Example 1: In this example, you obtain estimates of the plots of the PDF and CDF of a random variable uniformly distributed in the interval $[0, 1]$. (Recall that data points according to this distribution are generated using the Matlab function "`rand`.")

(a) Obtain a plot on your screen of estimated uniform$(0, 1)$ density (PDF) from 100000 "`rand`" pseudorandomly generated samples by running the following code:

```
n=100000;
x=rand(1,n);
N=floor(n/100);
A=min(x); B=max(x);
Delta=(B-A)/N;
t=A-Delta/2+[1:N]*Delta;
f=hist(x,t)/(Delta*n);
bar(t,f)
title('Estimated Uniform(0,1) PDF')
```

Does your plot look like a "jagged" rectangular pulse of amplitude one? One way to reduce the jaggedness in the density estimate is to use a smoothing filter. There is an entire theory devoted to such smoothing filters, which we do not touch upon in 3025.

(b) Run the following lines of code (as an add-on to the code already run in Example 1(a)) in order to obtain an estimated CDF for uniform(0,1) data:

```
p=hist(x,t)/n;
CDF=cumsum(p);
plot(t,CDF)
title('Estimated Uniform(0,1) CDF')
```

Does the plot look like the CDF of the uniform(0,1) density? The estimated CDF plot looks smoother than the estimated density plot—Why?

Example 2: In this example, you obtain estimates of the plots of the PDF and CDF of a random variable having the standard Gaussian density (PDF). This is the PDF

$$\frac{1}{\sqrt{2\pi}}\exp(-x^2/2), \quad -\infty < x < \infty.$$

(Data points according to this distribution are generated using the Matlab function "`randn`.")

(a) Obtain a plot on your screen of estimated standard Gaussian density from 100000 "`randn`" pseudorandomly generated samples by running the following code:

```
n=100000;
x=randn(1,n);
N=floor(n/100);
A=min(x); B=max(x);
Delta=(B-A)/N;
t=A-Delta/2+[1:N]*Delta;
f=hist(x,t)/(Delta*n);
bar(t,f)
title('Estimated Standard Gaussian PDF')
```

Compute $1/\sqrt{2\pi}$. Is this about equal to the peak value of the estimated density curve that you see on your computer screen?

(b) Run the following lines of code (as an add-on to the code already run in Example 2(a)) in order to obtain an estimate of the plot of the CDF of a standard Gaussian distribution:

```
p=hist(x,t)/n;
CDF=cumsum(p);
plot(t,CDF)
title('Estimated Gaussian(0,1) CDF')
```

Use your standard Gaussian CDF table on page 123 of Yates-Goodman to see if the actual CDF values at $z = 0, 0.5, 1.0, 1.5, 2.0$ conform to the estimated CDF values you get from the curve on your screen.

3.2 Exp 2: Simulating a Nonstandard Uniform Random Variable

We already know that a random variable uniformly distributed in interval $[0, 1]$ ("standard uniform distribution") is simulated with Matlab command "rand(1,1)". In this experiment, you will see how to simulate values of a random variable uniformly distributed over some other interval ("nonstandard uniform distribution").

Example 3: Execute the following lines.

```
a=3; b=7;
x=(b-a)*rand(1,50000)+a;
```

We will try to convince you that you have stored in Matlab memory 50000 simulations of the value of a uniform RV distributed over the interval [3,7]. Execute the lines:

```
max(x)
min(x)
```

Do the max and min values on your screen convince you that your 50000 data points all lie between 3 and 7? Run the following line:

```
mean(x>2 & x<5)
```

If RV X were uniformly distributed between 3 and 7, what would the exact value of the probability $P(2 < X < 5)$ be? (Use pen or pencil here.) Does this correspond with what the Matlab estimate of this probability just gave you? Run the following line:

```
mean(x > 6)
```

and interpret the result.

This example has hopefully convinced you that if you make the change of variable

$$X = (b - a)Z + a,$$

where Z has the standard uniform distribution, then you obtain a random variable X uniformly distributed in the interval $[a, b]$.

3.3 Exp 3: Simulating a Nonstandard Gaussian Random Variable

A Gaussian density function $f_X(x)$ is specified by two parameters, the mean μ and the variance σ^2. (Equivalently, it is determined by the mean μ and the standard deviation σ; the standard deviation is the square root of the variance.) The formula for the Gaussian density $f_X(x)$ is:

$$f_X(x) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right) \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

If you haven't done so already, then before starting this experiment you should look at figure 3.5 on page 119 of your textbook in order to see how changing μ and σ^2 affects the position and shape of the density curve.

The Matlab function "`randn`" simulates values of a standard Gaussian random variable (meaning that $\mu = 0$ and $\sigma^2 = 1$). By the time you finish this experiment, you will see how to simulate a nonstandard Gaussian RV. You will also learn ways to compute Gaussian probabilities.

Example 4: In this example, you will learn how to compute Gaussian probabilities using the Matlab function "`erf`". We have

$$P\{a \le X \le b\} = (1/2)\text{erf}((b-\mu)/\sigma\sqrt{2}) - (1/2)\text{erf}((a-\mu)/\sigma\sqrt{2})$$

for a RV X which is Gaussian(μ, σ^2). The line of Matlab code for this probability is

```
(1/2)*erf((b-M)/(sqrt(2*V)))-(1/2)*erf((a-M)/(sqrt(2*V)))
```

where `M` is the mean and `V` is the variance.

- Let X be standard Gaussian. Compute $P[-0.5 \leq X \leq 1]$ using the above line of Matlab code. Then, run the script:

  ```
  x=randn(1,50000);
  mean(x >= -0.5 & x<=1)
  ```

  Do you obtain roughly the same answer?

- Compute $P[3.5 \leq Y \leq 5]$ for Y Gaussian with mean 4 and variance 4.

- Compute $P[Y \geq 3.75]$ for this same Y. (Hint: In the format $P[a \leq Y \leq b]$, we want b to be infinite. Enter "`b=inf`" before executing the above line of Matlab code; Matlab recognizes "inf" as ∞.)

Example 5: You can also use the Matlab symbolic integrator to compute Gaussian probabilities, if the symbolic integrator comes with your version of Matlab. To illustrate this, see if your machine allows you to execute the following commands:

```
syms z
P=int((1/sqrt(2*pi))*exp(-z^2/2),0,1);
eval(P)
```

Do you see the value 0.3413 on your screen? This is the probability $P(0 \leq Z \leq 1)$ for a standard Gaussian random variable Z.

Example 6: In this example, you will see how to simulate the values of a Gaussian(μ,σ^2) random variable.

- Run the code:

  ```
  x=randn(1,50000);
  mean(x)
  var(x)
  ```

  Are you getting good estimates of the mean and variance of a standard Gaussian distribution?

- Run the code:

  ```
  M=-9; V=16;
  x=sqrt(V)*randn(1,50000)+M;
  mean(x)
  var(x)
  ```

  Are you getting good estimates of the mean and variance of a Gaussian distribution with mean -9 and variance 16?

As a result of Example 6, you have seen that the change of variable

$$X = \sigma Z + \mu$$

converts a standard Gaussian RV Z into a nonstandard Gaussian RV X with mean μ and variance σ^2.

3.4 Exp 4: Simulating a Binomial Random Variable

In this experiment, you will learn how to simulate a binomial random variable and how to compute binomial probabilities using Matlab.

Suppose you have a random variable X which has a binomial distribution with parameters n and p. Then, the PMF is given by

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, 1, 2, \cdots, n, \tag{1}$$

where the binomial coefficient $\binom{n}{x}$ is computed by:

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}.$$

For example, if you flip a coin with $P[H] = p$ n times and add up the number of heads, you get a Binomial(n,p) random variable.

Example 7: The following Matlab script gives a nice way to simulate values of a Binomial(n,p) random variable.

```
n=   ; %enter the value of n
p=   ; %enter the value of p
x=sum(rand(n,50000)>1-p);
```

The entries of vector `x` simulates values of a Binomial(n,p) random variable on 50000 independent trials. (In the preceding script, you can change the 50000 to any other number of trials you want.) Run the preceding script with $n = 3$ and $p = 1/2$. Then do the following histogram plot:

```
bar(0:3,hist(x,0:3)/50000)
```

Are the histogram heights roughly $1/8, 3/8, 3/8, 1/8$? (This is the binomial PMF you obtain, for example, in tossing a fair coin 3 times and counting the number of heads.)

Example 8: This example teaches you about *Pascal's triangle.* The following Matlab code generates the first n rows of Pascal's triangle (you have to enter the value of n before running the code):

```
S=1;
for i=1:n
v1=[0 S];
v2=[S 0];
S=v1+v2
end
```

- Run the above code for $n = 8$. Take the bottom row you see on the screen and verify by hand that these nine numbers are $\binom{8}{i}$, $i = 0, 1, 2, 3, 4, 5, 6, 7, 8$.

Example 9: This example teaches you a Matlab script from which binomial probabilities can be computed very fast. The following Matlab code generates the PMF of a binomial random variable with parameters n and p (the values of n and p have to be entered in first before running the code):

```
S=1;
for i=1:n
v1=[0 S];
v2=[S 0];
S=p*v1+(1-p)*v2;
end
PMF=S
```

- Run the above code for $n = 3$ and $p = 1/2$ and see if you do get the PMF $1/8, 3/8, 3/8, 1/8$ that was considered in Example 7.

3.5 Exp 5: Simulating a Poisson Random Variable

This experiment teaches you the following things concerning the Poisson(α) distribution:

- A nice way to compute Poisson probabilities.
- A nice way to simulate Poisson data.

PMF values for a Poisson(α) RV X are given by the formula:

$$p_X(x) = \exp(-\alpha)\frac{\alpha^k}{k!}, \quad k = 0, 1, 2, 3, \cdots. \tag{2}$$

The following Matlab code generates the first $n + 1$ of these PMF values $p_X(x)$ ($x = 0, 1, \cdots, n$):

```
n=   ; %enter the value of n
alpha =    ; %enter value of Poisson parameter alpha
S(1)=1;
for k=1:n
S(k+1)=alpha*S(k)/k;
end
PMF=exp(-alpha)*S
```

Example 10: Run the preceding Matlab code with alpha=0.5 and $n = 3$. Examine the four Poisson PMF probabilities that you see on your screen. Now try to use formula (2) to directly compute some of these four probabilities to see if they agree with what is on your Matlab screen.

The following Matlab code generates simulated values of a Poisson(α) random variable for n independent trials:

```
clear;
n=   ; %enter the number of trials n
alpha=   ; %enter in the value of Poisson parameter alpha
for i=1:n
t=0;
count=-1;
while t<1
t=t-log(rand(1,1))/alpha;
count=count+1;
end
x(i)=count;
end
```

(Do not try to understand why this code works now. In a future EE 3025 lecture, I will try to explain why it works.)

Example 11: Run the preceding script with alpha=0.5 and $n = 10000$. Then execute the line

```
mean(x==1)
```

This should be an estimate of the following Poisson probability:

$$P(X = 1) = (0.5)\exp(-0.5) = 0.3033.$$

Is the estimate pretty good? Now compare the estimates

```
mean(x==0), mean(x==2)
```

to

$$P(X = 0) = \exp(-0.5), \quad P(X = 2) = (0.5)^2 \exp(-0.5)/2,$$

respectively.

3.6 Exp 6: Simulating an Exponential Random Variable

The density function for a random variable X having the Exponential(a) distribution is

$$f_X(x) = a\exp(-ax)u(x).$$

The following very simple script simulates values of such a random variable X over 100000 independent trials:

```
clear;
a= ; %enter in value of parameter a
x=-log(rand(1,100000))/a;
```

(We will eventually prove in class why this works.)

Example 12: Suppose a randomly chosen citizen is expected to live 70 years. Then we would model the lifetime of this citizen as an Exponential(a) random variable X with $a = 1/70$. Run the preceding script with $a = 1/70$. Execute the command

```
mean(x)
```

Is this the result you expected? Also, execute

```
mean(x>70)
```

and then compare this with the exact probability

$$P(X > 70) = \int_{70}^{\infty} a \exp(-ax) dx = \exp(-1)$$

that you get for $a = 1/70$. If you have time, re-do Example 1 in which you take the data vector `x` to be the data vector you just generated. You will then obtain an estimated plot of the exponential function $f_X(x)$.

EE 3025 S2007 Recitation 3 Lab Form

Name and Student Number of Team Member 1:

Name and Student Number of Team Member 2:

Name and Student Number of Team Member 3:

****

Study Experiments 2-3 on simulating nonstandard uniform and Gaussian data. In this week's lab report, I will have you simulate some data of this type and have you estimate various probabilities, means, or variances using the data.