2 Rec 2: Conditional Probability and Independence

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.

- Estimation of Conditional Probabilities
- Simulation of Relay Circuits
- Bayes Method in Matlab

2.1 Exp 1: Forward Conditional Probabilities of a Channel

Think of a discrete communication channel modeled as a box:

$$\text{input } X \rightarrow \boxed{\text{channel}} \rightarrow \text{output } Y$$

For each possible value y of the output Y, and each possible value x of the input X, we can attempt to estimate the so-called “forward conditional probability”

$$P(Y = y|X = x)$$

that the output will be y given that the input is x. To do this, we can let the channel operate on a long sequence of inputs

$$(x_i : i = 1, 2, \cdots, n)$$

and observe the resulting sequence of outputs

$$(y_i : i = 1, 2, \cdots, n).$$

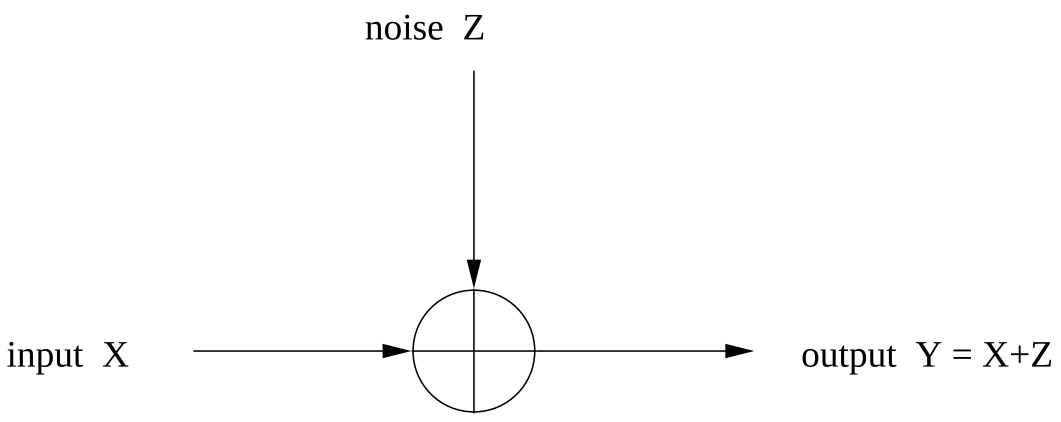
Then

$$P(Y = y|X = x) \approx \frac{N(x, y)}{N(x)},$$

where

- $N(x)$ is the number of entries x_i of the input sequence (x_i) that are equal to x.
- $N(x, y)$ is the number of entries (x_i, y_i) of the input/output sequence $\{(x_i, y_i)\}$ that are equal to (x, y).

This experiment illustrates this estimation technique for an "additive random noise channel," modeled as



where the "channel noise" Z is random and is independent of the input X.

2.1.1 Generating Channel Inputs

- Execute the commands:

```
x = ceil(2*rand(1,10000));
bar(1:2,hist(x,1:2)/10000)
```

The first command generates 10000 pseudorandom inputs to the channel. The second command plots the histogram of these inputs. Look at the histogram and answer the following questions:

- There are two possible channel inputs: what are they?
- Based upon the histogram estimates, what do you think are the theoretical values of the two channel input probabilities?

2.1.2 Generating Channel Noise

- Execute the commands:

```
z=floor(3*rand(1,10000));
bar(0:2,hist(z,0:2)/10000)
```

The first command generates 10000 pseudorandom channel noise samples. The second command plots the histogram of these samples. Look at the histogram and answer the following questions:

- There are three possible values for each channel noise sample: what are they?
- Based upon the histogram estimates, what do you think are the theoretical values of the three channel noise probabilities?

2.1.3 Generating Channel Outputs

- Execute the commands:

```
y=x+z;
bar(1:4,hist(y,1:4)/10000)
```

The first command generates 10000 pseudorandom channel outputs. The second command plots the histogram of these outputs. Look at the histogram and answer the following questions:

- There are four possible values for each channel output: what are they?
- Based upon the histogram estimates, what do you think are the theoretical values of the four channel output probabilities?

2.1.4 Estimating Forward Conditional Probabilities

- Execute the commands:

```
N1=sum(x==1)
N12= sum(x==1 & y==2)
N12/N1
```

You will see three numbers on your screen. The first number is how many of the 10000 channel inputs are equal to 1. The second number is how many of the 10000 channel input/output pairs are equal to $(1, 2)$. The third number is the estimate of

$$P(Y = 2|X = 1),$$

the conditional probability that the output is 2 given that the input is 1.

2.2 Exp 2: Estimating Channel Matrix

This experiment uses the same channel used in Experiment 1 and continues beyond Experiment 1.

- First, run the following command in order that Matlab will simulate 10000 channel inputs and outputs:

```
x=ceil(2*rand(1,10000));
z=floor(3*rand(1,10000));
y=x+z;
```

You have stored in Matlab memory a vector x of 10000 simulated channel inputs, a vector z of 10000 simulated channel noise samples, and a vector y of 10000 simulated channel outputs.

- Execute the following Matlab script:

```
t=1:10000;
M=zeros(2,4);
for i=1:2
for j=1:4
if sum(x(t)==i)>0
M(i,j)=sum(x(t)==i & y(t)==j)/sum(x(t)==i);
else
end
end
end
M
```

Examine the 2×4 matrix M that appears on your computer screen. It should be approximately equal to the matrix

$$\begin{array}{cccc} 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \end{array}$$

The matrix just given is called *the channel matrix.* So, you have just estimated the channel matrix.

Here is what the channel matrix means. For our channel, we have two possible inputs $1, 2$ and four possible outputs $1, 2, 3, 4$. The channel matrix has two rows and four columns, reflecting the fact that there are two inputs and four outputs. The element in row i and column j of the channel matrix is

$$P(Y = j|X = i),$$

the conditional probability that the channel output is equal to j given that the channel input is equal to i.

- Look at the estimated channel matrix that you generated on your computer screen. Check that the sum of each of the two rows is equal to 1. What is the significance of this fact?

The channel matrix can be used to define how the channel operates and does not depend upon which sequence of channel inputs is used. Therefore, if you use some other sequence of channel inputs to estimate the channel matrix, we will get roughly the same estimate for the channel matrix that you got before. Let's check this property by biasing the channel inputs $1, 2$ so that input 1 is used with prob $2/3$ and input 2 is used with prob $1/3$.

$\rightarrow$ $\boxed{\text{Execute the following Matlab one-liner:}}$

```
x=1+(rand(1,10000)>2/3);
```

$\rightarrow$ $\boxed{\text{Generate a histogram of the values of } x \text{ by executing the Matlab one-liner:}}$

```
bar(1:2,hist(x,1:2)/10000)
```

- Does the histogram indicate that the entries $1, 2$ of x are appearing with probabilities $2/3, 1/3$?

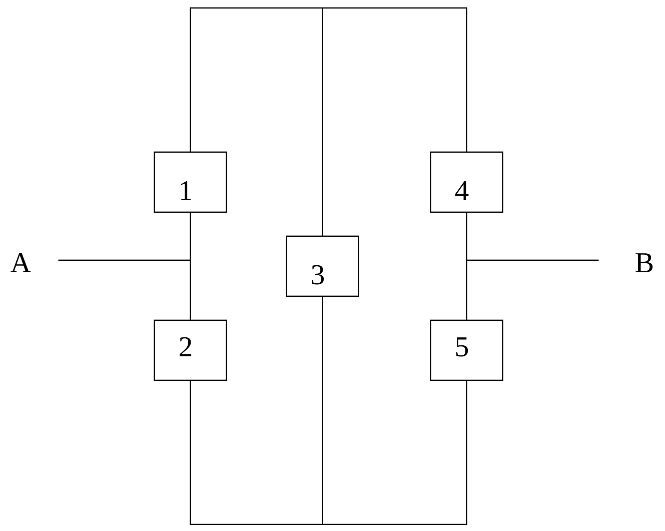
$\rightarrow$ $\boxed{\text{Execute the Matlab script:}}$

```
x=1+(rand(1,10000)>2/3);
z=floor(3*rand(1,10000));
y=x+z;
t=1:10000;
M=zeros(2,4);
for i=1:2
for j=1:4
if sum(x(t)==i)>0
M(i,j)=sum(x(t)==i & y(t)==j)/sum(x(t)==i);
end
end
end
M
```

- Does the estimated channel matrix M on your screen approximately agree with what you got before?

2.3 Exp 3: Relay Circuits

In this experiment, you will test the reliability of a relay circuit by doing a simulation of the circuit below. Let us suppose that the switches 1-5 are each 90% reliable, and that



they operate independently. Letting p_i denote the probability that switch i will work ($i = 1, 2, 3, 4, 5$), we are assuming that $p_i = 0.90$ for each i. As discussed in class, let X_i be a RV modeling the action of switch i; X_i takes the value 1 when switch i works, and takes the value 0 when switch i doesn't work. Let us call X_i the "state variable for switch i", $i = 1, 2, 3, 4, 5$. We want to let the circuit run 10000 times, by repeatedly re-setting the values of the state variables. Let the vectors $x1, x2, x3, x4, x5$ denote the 10000 simulated values of state variables X_1, X_2, X_3, X_4, X_5 that we shall be using, respectively. We want to let Matlab select these values for us.

$\rightarrow$ $\boxed{\text{Execute the Matlab script:}}$

```
p1=.9; p2=.9; p3=.9; p4=.9; p5=.9;
x1=(rand(1,10000)<p1);
x2=(rand(1,10000)<p2);
x3=(rand(1,10000)<p3);
x4=(rand(1,10000)<p4);
x5=(rand(1,10000)<p5);
```

We have stored in Matlab memory the information we need to see how well the circuit is operating. How do we make use of this information to see whether the circuit works (i.e., passes current from point A to point B) in each of the 10000 trials? Notice that

- The circuit works if switches 1,4 both work.
- The circuit works if switches 2,5 both work.
- The circuit works if switches 1,3,5 all work.
- The circuit works if switches 2,3,4 all work.
- The four cases above are the only cases in which the circuit works.

We can generate a binary vector "out" as follows. If the circuit works on trial i ($i = 1, \cdots, 10000$) then the i-th entry of "out" is 1; otherwise it is 0.

$\rightarrow$ $\boxed{\text{Execute the Matlab one-liner:}}$

```
out=max([x1.*x4;x2.*x5;(x1.*x3).*x5;(x2.*x3).*x4]);
```

The frequency with which "1" appears in "out" is an estimate of the probability that the circuit works.

$\rightarrow$ $\boxed{\text{Execute the Matlab one-liner:}}$

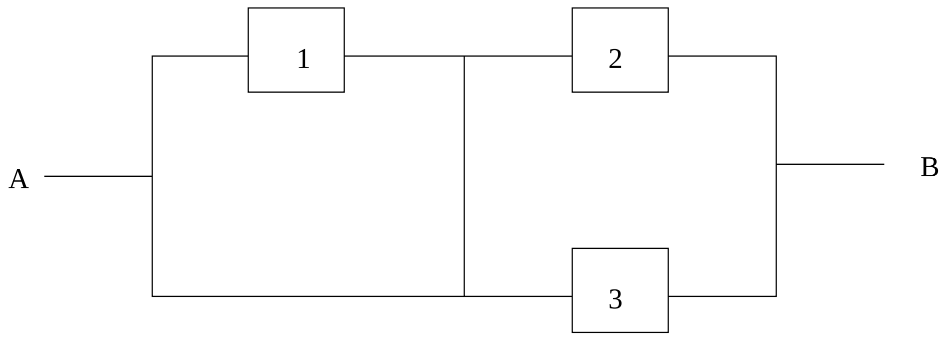
```
empprob_circuit_works=mean(out)
```

"empprob" refers to the fact that your estimate is an empirical probability.

- Do the experiment again. Does your estimate for the probability that the circuit works change by much? (It shouldn't.)
- Try the experiment with a different choice of $p1, p2, p3, p4, p5$. For example, take $p2 = p3 = p5 = 0$. Then, you just have switches 1,4 in series, and we know from class the probability the circuit works is p_1p_2— see if the empirical estimate of the probability that the circuit works is close to p_1p_2.
- The key part of our solution to the circuit-checking via simulation problem was the Boolean Matlab expression

$$\text{out} = \max([\text{x1.} * \text{x4}; \text{x2.} * \text{x5}; (\text{x1.} * \text{x3}). * \text{x5}; (\text{x2.} * \text{x3}). * \text{x4}])$$

  Give a Boolean Matlab expression that you could use to test the circuit at the top of the next page.



2.4 Exp 4: Bayes Method in Matlab

This experiment shows how Matlab can be used to give an easy implementation of Bayes Method. Bayes Method starts with an array of "forward conditional probabilities"

	F_1	F_2	F_3
E_1	$P(F_1\|E_1)$	$P(F_2\|E_1)$	$P(F_3\|E_1)$
E_2	$P(F_1\|E_2)$	$P(F_2\|E_2)$	$P(F_3\|E_2)$

(1)

and "prior probabilities" $P[E_1], P[E_2]$, and then builds the array of "backward conditional probabilities"

	F_1	F_2	F_3
E_1	$P(E_1\|F_1)$	$P(E_1\|F_2)$	$P(E_1\|F_3)$
E_2	$P(E_2\|F_1)$	$P(E_2\|F_2)$	$P(E_2\|F_3)$

(2)

in two steps:

Step 1: For each i, the i-th row of array (1) is multiplied by $P(E_i)$. This yields the array of "joint probabilities"

	F_1	F_2	F_3
E_1	$P(E_1 \cap F_1)$	$P(E_1 \cap F_2)$	$P(E_1 \cap F_3)$
E_2	$P(E_2 \cap F_1)$	$P(E_2 \cap F_2)$	$P(E_2 \cap F_3)$

(3)

Step 2: For each j, the j-th column in the array (3) is divided through by its column sum. These column sums are the $P(F_j)$'s.

$\rightarrow$ $\boxed{\text{Execute the Matlab commands:}}$

```
M_forward = [ 0.23  0.47  0.30
              0.15  0.33  0.52];
pE = [0.40 0.60];
```

`M_forward` is the matrix of "forward conditional probabilities" (1) that we will operate on using Bayes rule, and pE is the vector $[P(E_1)\ \ P(E_2)]$ of "prior probabilities".

$\rightarrow$ $\boxed{\text{Execute the Matlab command:}}$

```
N=diag(pE)*M_forward
```

The matrix N that you see printed out on your computer screen is the matrix (3) of joint probabilities.

- Check that the sum of all of the elements of N is equal to one. (Type in a Matlab command that will do this.)

$\rightarrow$ $\boxed{\text{Execute the Matlab command:}}$

```
sum(N)
```

The vector that is on your computer screen consists of the column sums of the columns in N. The components are also $P(F_j)$'s, and you should check that they add up to 1.

$\rightarrow$ $\boxed{\text{Execute the Matlab command:}}$

```
M_backward = N*inv(diag(sum(N)))
```

The matrix you see on your screen now should be the matrix of "backward probabilities" (2). Check that each of its columns sums up to 1.

$\rightarrow$ $\boxed{\text{Execute the Matlab command:}}$

```
pF= pE*M_forward
```

The components of the vector pF you see on your screen are the probabilities $P(F_j)$. They are also the column sums of N, but we have just obtained them in a more direct way. (The vector pF should coincide with the vector `sum(N)` found earlier.)

- For the channel matrix

  ```
  1/4 1/2 1/4  0
   0  1/4 1/2 1/4
  ```

  and the channel input probabilities $P(1) = 2/3$, $P(2) = 1/3$, work out the matrix of "backward probabilities" according to the Matlab method just given. What are the four channel output probabilities? Is there a channel output for which the two channel inputs are equally likely?

2.5 Exp 5: Let's Make a Deal

Here is how the game "Let's Make a Deal" works. Before the contestant appears, Monte Hall hides a new car behind one of three doors; the door he selects is random. Then the contestant is led in and chooses a door at random as initial guess. The door selected by the contestant is not opened. Instead, Monte opens a door not selected by the contestant behind which there is no car. Then, Monte offers the contestant one of two choices: (1) the contestant can stay with his/her initial choice of door; or (2) the contestant switches his/her guess to the third remaining door. We are going to use Matlab to estimate the probability that the contestant will win the car under Strategy (1); then we will use Matlab to estimate the probability that the contestant will win the car under Strategy (2).

For convenience, we number the 3 doors 0,1,2 instead of 1,2,3. Monte Hall's choice of door to put the car behind will be modeled as a vector of length 50000:

```
cardoor = floor(3*rand(1,50000));
```

The contestant's choice of door under Strategy (1) will be modeled as a vector of length 50000:

```
strategy1door = floor(3*rand(1,50000));
```

Here is a line of code that will estimate the probability that the contestant will win the car under Strategy (1):

```
mean(cardoor==strategy1door)
```

Here are some lines of code that create a vector giving Monte's choice of door (the door not equal to cardoor or strategy1door):

```
x=cardoor; y=strategy1door;
montedoor = rem(x+1,3).*(x==y) + (3-x-y).*(x~=y);
```

To check that this code worked, run the three lines of code

```
sum(montedoor==cardoor)
sum(montedoor==strategy1door)
sum(montedoor==0) + sum(montedoor==1) + sum(montedoor==2)
```

If the 3 answers are 0,0,50000, then the montedoor vector is the right one. The Strategy (2) door choice can be computed as follows:

```
strategy2door =  3-montedoor-strategy1door;
```

Now you estimate how often you get the car:

```
mean(strategy2door==cardoor)
```

Is Strategy (1) or Strategy (2) the better strategy for playing "Let's Make a Deal"? Would you like to be a contestant on "Let's Make a Deal"? Let's see if you can use conditional probabilities to analyze the efficacy of Strategy (2):

- For simplicity, always assume Monte hides the car behind door 0.
- If the contestent's initial choice of door is door 0, what is the conditional probability he/she wins the car under Strategy (2)?
- If the contestent's initial choice of door is door 1, what is the conditional probability he/she wins the car under Strategy (2)?
- If the contestent's initial choice of door is door 2, what is the conditional probability he/she wins the car under Strategy (2)?

Use the law of total probability to take a weighted average of the three conditional probabilities you just computed in order to compute the probability that the car will be won under Strategy (2).

EE 3025 S2007 Recitation 2 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 Experiment 3 on relay systems. I will give you a relay system consisting of several components, along with the probability that each component is “on”. I will give you some Matlab code to simulate the state of each component over several thousand independent trials. Based on these simulations, I will ask you to estimate the probability that the relay system is operative.