Ajay Kc
013213328
Project 2 Report

1. Probability of erroneous transmission

Introduction

The experiment is replicating the real-life example of message transmission through a noisy communication channel. The objective is to check the probability of a bit being changed after the transmission. The probability of bit switching from 0 to 1 and vice-versa has been provided. Through the experiment, the probability of a transmitted signal being received incorrectly is calculated.

Methodology

A random number between 0 and 1 is generated which is defined as the transmitted signal. Then, based on the probabilities given, the received signal will also be calculated. After the received signal has been calculated, the program checks if transmitted signal is equal to received signal or not. If the two are equal, then the experiment is considered a "success". The experiment is repeated 100,000 times and the probability of getting incorrect signal is calculated.

Source code

```
"""
Ajay Kc
013213328
EE 381
Project 1 Part 1

The program calculates the probability of a transmitted
signal being received incorrectly
when a 0 or 1 is transmitted
"""

import numpy as np
p = 0.4 #Probability that generated bit is 0
e0 = 0.02 #Probability of transmission error when
transmitted signal is 1
e1 = 0.03 #Probability of transmission error when
transmitted signal is 0

def transmitSignal():

    m = np.random.rand()
    if ( m <=p):
        return 0
    else:
        return 1
```

```
def receiveSignal(bitTransmitted):

    m = np.random.rand()
    if(bitTransmitted==0):
      if(m<=e0):
          return 1
      else:
          return 0
    else:
       if(m<=e1):
           return 0
       else:
           return 1

count=0
for i in range(0,100000):
    bitTransmitted = transmitSignal()
    if(bitTransmitted!=receiveSignal(bitTransmitted)):
        count +=1

probabilityOfFailure = count/100000
print('The probability of transmitted bit being received
incorrectly is %s' %probabilityOfFailure)
```

Probability of transmission error	
Ans	**0.02592**

2. Conditional probability P(R=1|S=1)

Introduction

The experiment is similar to the first one where the objective is to check if the transmitted signal is equal to the received signal or not. However, the focus would on checking the probability of received signal being equal to one if the transmitted signal was equal to 1.

Methodology

Since the experiment is focused only on cases where transmitted signal is equal to 1, therefore the probability of getting a 1 as a transmitted signal is always set to 1. Then, the experiment is similar to the first experiment. If the received signal is equal to 1 given that transmitted signal was 1, the experiment is considered a "success". The experiment is repeated 100,000 times and the probability of getting "success" is calculated.

Source Code

```
"""
Ajay Kc
013213328
EE 381
Project 1 Part 2

The program calculates the probability of received signal
being 1 when 1 is transmitted as the signal
"""

import numpy as np
p = 0 #Probability that generated bit is 0
e0 = 0.02 #Probability of transmission error when
transmitted signal is 1
e1 = 0.03 #Probability of transmission error when
transmitted signal is 0

def transmitSignal():

    m = np.random.rand()
    if ( m <=p):
        return 0
    else:
        return 1

def receiveSignal(bitTransmitted):
```

```
    m = np.random.rand()
    if(bitTransmitted==0):
      if(m<=e0):
          return 1
      else:
          return 0
    else:
       if(m<=e1):
           return 0
       else:
           return 1

count=0
for i in range(0,100000):
    bitTransmitted = transmitSignal()
    if(receiveSignal(bitTransmitted)==1):
        count +=1

probabilityOfSuccess = count/100000
print('The probability of received signal being 1 when 1 is
transmitted is %s' %probabilityOfSuccess)
```

Conditional Probability (R=1|S=1)	
Ans	**0.96972**

3. Conditional probability P(S=1|R=1)

Introduction

The experiment checks the probability of transmitted signal being 1 only if the received signal is 1. The experiment does not consider the cases when the received signal was 0.

Methodology

Since the experiment is focused only on cases where received signal is equal to 1, therefore the experiment is considered successful only if the transmitted signal is equal to 1. If the transmitted signal is not equal to 1, the experiment is repeated again until 1 is transmitted. If the transmitted signal is equal to 1 given that received signal was 1, the experiment is considered a "success". The experiment is repeated 100,000 times and the probability of getting "success" is calculated.

Source code

```
"""
Ajay Kc
013213328
EE 381
Project 1 Part 3

The program calculates the probability of transmitted
signal being 1 when 1 is received as the signal
"""

import numpy as np
p = 0.4 #Probability that generated bit is 0
e0 = 0.02 #Probability of transmission error when
transmitted signal is 1
e1 = 0.03 #Probability of transmission error when
transmitted signal is 0

def transmitSignal():

    m = np.random.rand()
    if ( m <=p):
        return 0
    else:
        return 1

def receiveSignal(bitTransmitted):

    m = np.random.rand()
```

```
    if(bitTransmitted==0):
      if(m<=e0):
          return 1
      else:
          return 0
    else:
       if(m<=e1):
           return 0
       else:
           return 1

successCount=0
count = 0
while (count<100000):
    bitTransmitted = transmitSignal()
    if(receiveSignal(bitTransmitted)==1):
            count +=1
            if(bitTransmitted==1):
                successCount+=1

probabilityOfSuccess = successCount/100000
print('The probability of transmitted signal being 1 when 1
is transmitted is %s' %probabilityOfSuccess)
```

Conditional Probability (S=1|R=1)	
Ans	**0.98651**

4. Enhanced Transmission method

Introduction

The experiment is similar to the first experiment and follows the same concept. The only difference is the signal is transmitted three times and received three times. Based on the three signals received, voting and majority rule is applied on the three received signals to decide whether 1 or 0 should be selected as the received signal.

Methodology

Firstly, the transmitted signal is generated only once. Then, the received signal is generated three times based on the given probabilities. Out of the three signals, the majority is calculated and declared as the final received signal. Finally, if the transmitted signal and the received signal are not equal, the experiment is considered a "failure" and the experiment is repeated 100,000 times to calculate the probability of getting a "failure"

Source code

```
"""
Ajay Kc
013213328
EE 381
Project 1 Part 4

The program calculates the probability of a transmitted
signal being received incorrectly
when a 0 or 1 is transmitted and the same bit is trasmitted
three times
"""

import numpy as np
p = 0.4 #Probability that generated bit is 0
e0 = 0.02 #Probability of transmission error when
transmitted signal is 1
e1 = 0.03 #Probability of transmission error when
transmitted signal is 0

def transmitSignal():

    m = np.random.rand()
    if ( m <=p):
        return 0
    else:
        return 1
```

```
def receiveSignal(bitTransmitted):

    bits=""
    for i in range (0,3):
        m = np.random.rand()
        if(bitTransmitted==0):
            if(m<=e0):
              bit = 1
            else:
              bit = 0
        else:
            if(m<=e1):
                bit = 0
            else:
                bit = 1

        bits = bits + str(bit)

    return bits

count=0
for i in range(0,100000):
    bitTransmitted = transmitSignal()
    bitsReceived = receiveSignal(bitTransmitted)
    if(bitsReceived.count("0")>bitsReceived.count("1")):
        bit = 0
    else:
        bit = 1

    if(bit!=bitTransmitted):
        count +=1

probabilityOfFailure = count/100000
print('The probability of transmitted bit being received
incorrectly is %s' %probabilityOfFailure)
```

Probability of error with enhanced transmission	
Ans	**0.00218**