Assignment 6, Question 3, Amogha Sekhar, A53301791

In [1]:

```
import numpy as np
```

In [2]:

```
#Read the data files and storing them into lists.

X, Y, P = [], [], []

for fd in open('spectX.txt'):
    fd = fd.split()
    temp = []
    for i in range(len(fd)):
        temp.append(int(fd[i]))
    X.append(temp)

for fd in open('spectY.txt'):
    fd = fd.strip()
    for i in range(len(fd)):
            fd = int(fd)
    Y.append(fd)

#Initializing pi=0.05
for i in range(len(X[0])):
    P.append(0.05)
```

In [3]:

```
#Calculating the log probabilities
import math

def calc_log_prob(X, Y, P):
    T = len(X)
    sum_prob = 0

    for t in range(T):
        prod = 1
        for i in range(len(X[t])):
            prod *= (pow(1 - P[i],X[t][i]))
        if Y[t] == 0:
            sum_prob += np.log(prod)
        elif Y[t] == 1:
            sum_prob += np.log(1 - prod)

    log_prob = sum_prob / T

    return log_prob
```

In [4]:

```
#Calculating the denominator of the expression

def calc_denom(X, P):
    prod = 1
    for i in range(len(X)):
        prod *= (pow((1 - P[i]),X[i]))
    return (1 - prod)
```

In [5]:

```
#Performing M-step of the EM algorithm
def m_step(X, Y, P):
    T = len(X)
    I = len(P)
    denom = []

    for t in range(T):
        denom.append(calc_denom(X[t], P))

    for i in range(I):
        T_i = [1 for x in X if (x[i] == 1)]
        prob = 0

        for t in range(T):
            prob += ((Y[t] * X[t][i] * P[i])/denom[t])
        P[i] = prob/len(T_i)

    return P
```

In [6]:

```
#Computing the mistakes made in each iteration
def calc_mistake(X, Y, P):
    mistake = 0
    T = len(X)

    for t in range(T):
        prod = 1
        for i in range(len(X[t])):
            prod *= pow((1 - P[i]),X[t][i])
        prob = 1 - prod

        if Y[t] == 0 and prob >= 0.5:
            mistake += 1
        elif Y[t] == 1 and prob <= 0.5:
            mistake += 1

    return mistake
```

In [7]:

```
def iterate(X, Y, P):
    list_to_print = [1,2,4,8,16,32,64,128,256]
    print(0, calc_mistake(X, Y, P), calc_log_prob(X, Y, P))
    for i in range(257):
        P = m_step(X, Y, P)
        if i+1 in list_to_print:
            print(i+1, calc_mistake(X, Y, P), calc_log_prob(X, Y, P))
```

In [8]:

```
iterate(X, Y, P)
```

```
0 175 -0.9580854082157914
1 56 -0.49591639407753635
2 43 -0.40822081705839114
4 42 -0.3646149825001877
8 44 -0.3475006162087826
16 40 -0.33461704895854844
32 37 -0.32258140316749784
64 37 -0.3148266983628559
128 36 -0.3111558472151897
256 36 -0.310161353474076
```

In [ ]: