

250HW6 Q3

November 13, 2018

```
In [54]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from numpy.linalg import *
import math

In [61]: X = pd.read_csv('spectX.txt', sep=" ", header=None)
Y = pd.read_csv('spectY.txt', sep=" ", header=None)
X = X.values[:, :23]
Y = Y.values

In [62]: p = 1/23*np.ones((23,1))

In [57]: def CalcProb(index, x, y, p):
    xi = x[index]
    pi = p[index,0]
    denom = 1 - Calc_product(x,p)
    return y*xi*pi/denom

def Calc_product(x,p):
    result = 1
    for i in range(x.shape[0]):
        result *= np.power((1-p[i,0]),x[i])
    return result

def EMUpdate(X,Y,p):
    result = np.ones((23,1))
    for i in range(p.shape[0]): #for each pi
        pi = 0
        Ti = sum(X[:,i])
        for j in range(X.shape[0]): #for each row
            prob = CalcProb(i, X[j,:], Y[j,0], p)
            pi += prob
        pi /= Ti
        result[i,0] = pi
    return result

In [58]: def ComputeLogLikelihood(X,Y,p):
    T = X.shape[0]
```

```

    result = 0
    for i in range(T):
        if Y[i,0] == 1:
            result += np.log(1 - Calc_product(X[i,:],p))
        else:
            result += np.log(Calc_product(X[i,:],p))
    result /= T
    return result

def CalcNumMistakes(X,Y,p):
    T = X.shape[0]
    count = 0
    for i in range(T):
        if Y[i,0]==0 and (1 - Calc_product(X[i,:],p)) >= 0.5:
            count += 1
        elif Y[i,0]==1 and (1 - Calc_product(X[i,:],p)) <= 0.5:
            count += 1
        else:
            count = count
    return count

In [63]: # 256 iterations of EM
it = 257
for i in range(it):
    likelihood = ComputeLogLikelihood(X,Y,p)
    mistakes = CalcNumMistakes(X,Y,p)
    p = EMUpdate(X,Y,p)
    if i in (0,1,2,4,8,16,32,64,128,256):
        print('iteration: %d' % i + 'likelihood: %.5f' % likelihood + 'mistakes: %d' %

iteration: 0likelihood: -1.04456mistakes: 195
iteration: 1likelihood: -0.50494mistakes: 60
iteration: 2likelihood: -0.41076mistakes: 43
iteration: 4likelihood: -0.36513mistakes: 42
iteration: 8likelihood: -0.34766mistakes: 44
iteration: 16likelihood: -0.33468mistakes: 40
iteration: 32likelihood: -0.32259mistakes: 37
iteration: 64likelihood: -0.31483mistakes: 37
iteration: 128likelihood: -0.31116mistakes: 36
iteration: 256likelihood: -0.31016mistakes: 36

```