CSE250a_HW6.3

November 13, 2017

1 6.3d

In [2]:
```
import numpy as np
```

In [37]:
```
''' Load files and constants '''

spectX_fh = 'hw6_spectX.txt'
spectY_fh = 'hw6_spectY.txt'

spectX = np.loadtxt(spectX_fh)
spectY = np.loadtxt(spectY_fh)

ITERS = 256
T = 267 #examples
n = 23 #inputs
PI0 = 0.05 #initial parameter
ITERS_LIST = [0,1,2,4,8,16,32,64,128,256]

Ti_arr = np.zeros(n)
for i in range(n):
    Ti_arr[i] = np.sum(spectX[:,i])
```

In [47]:
```
''' Functions '''

# P(Y|X)
# give p_arr, x_t (arr), y_t (value)
def likelihood(p, x, y):
    prod = np.prod((1-p)**x)
    out = (1-y)*prod + y*(1-prod)
    return(out)

# E-step of EM algorithm
# give x_t, y_t, p arrays for current iteration of t
def e_step(x, y, p):
    numer = y*x*p
    denom = 1-np.prod((1-p)**x)
    return(numer/denom)
```

```
# main
def EM_algorithm(x_data, y_data):
    mistakes = [] # mistakes in each iteration
    L = [] # log-likelihood for each iteration
    p = np.full(n, PI0) #initialize p_arr with 0.05
    for i in range(ITERS+1):
        L_i = 0
        M_i = 0
        em_sum = 0
        for t in range(T):
            p_yx = likelihood(p, x_data[t], y_data[t])
            L_i += np.log(p_yx)
            em_sum += e_step(x_data[t], y_data[t], p)
            if (p_yx < 0.5):
                M_i += 1
        p = em_sum/Ti_arr
        mistakes.append(M_i)
        L.append(L_i/T)
        if i in ITERS_LIST:
            print('iteration: %d \t number of mistakes M %d \t log-likelihood L %d' % (
    return(mistakes, L)
```

2 Solution

```
In [48]: mistakes_list, log_likelihoods = EM_algorithm(spectX, spectY)
```

```
iteration: 0          number of mistakes M 175          log-likelihood L 0
iteration: 1          number of mistakes M 56          log-likelihood L 0
iteration: 2          number of mistakes M 43          log-likelihood L 0
iteration: 4          number of mistakes M 42          log-likelihood L 0
iteration: 8          number of mistakes M 44          log-likelihood L 0
iteration: 16          number of mistakes M 40          log-likelihood L 0
iteration: 32          number of mistakes M 37          log-likelihood L 0
iteration: 64          number of mistakes M 37          log-likelihood L 0
iteration: 128          number of mistakes M 36          log-likelihood L 0
iteration: 256          number of mistakes M 36          log-likelihood L 0
```