import numpy as np

from numpy import \*

import matplotlib.pyplot as plt

#----------------------------------------------------------#

#----------------- UTILITY FUNC ---------------------#

#----------------------------------------------------------#

def init\_S(num\_cliques):

S = np.ones((2,2,2,2,num\_cliques))

return S

def compute\_mu\_hat(mu, data, clique\_nodes):

mu\_hat = mu

for j in range(2):

for k in range(2):

# compute sum

sum = 0

for i in range(30):

(s,t) = clique\_nodes

sum += (data[s][i] == j)\*(data[t][i] == k)

mu\_hat[j][k] = 1/30.0 \* sum

return mu\_hat

def init\_mu(clique\_nodes):

list = [None] \* len(clique\_nodes)

i = 0

for n in clique\_nodes:

c\_size = len(n)

list[i] = np.ones((c\_size,c\_size))

i = i+1

return list

# compute P(x1...xd) = (1/Z) \* prod(psi\_st(x\_s, x\_t))

def get\_likelihood(S, data):

normalizer = 0

for i in range(2):

for j in range(2):

for k in range(2):

for m in range(2):

normalizer += np.prod(S[i,j,k,m])

prob0 = np.zeros((2,1))

prob1 = np.zeros((2,1))

prob2 = np.zeros((2,1))

prob3 = np.zeros((2,1))

for i in range(2):

prob0[i] = np.sum(np.prod(S[i,:,:,:],3))/normalizer

prob1[i] = np.sum(np.prod(S[:,i,:,:],3))/normalizer

prob2[i] = np.sum(np.prod(S[:,:,i,:],3))/normalizer

prob3[i] = np.sum(np.prod(S[:,:,:,i],3))/normalizer

print "p(x0) = \n", prob0

print "p(x1) = \n", prob1

print "p(x2) = \n", prob2

print "p(x3) = \n", prob3

total = 1.0

for i in range(30):

total \*= np.prod(S[data[0][i], data[1][i], data[2][i], data[3][i],:])/normalizer

print "likelihood: ", total

#----------------------------------------------------------#

#----------------- PROBLEM 2(i) ---------------------#

#----------------------------------------------------------#

def problem2i():

cliques = [(0,1), (1,2), (2,3), (0,3)]

num\_c = len(cliques)

S = init\_S(num\_c)

mu\_hat = init\_mu(cliques)

# compute mu\_hat for each clique

for i in range(num\_c):

mu\_hat[i] = compute\_mu\_hat(mu\_hat[i], data, cliques[i])

for n in range(10):

for c in range(num\_c):

if(c == 0): # clique (0,1)

sum00 = np.sum(np.prod(S[0,0,:,:],2))

sum01 = np.sum(np.prod(S[0,1,:,:],2))

sum10 = np.sum(np.prod(S[1,0,:,:],2))

sum11 = np.sum(np.prod(S[1,1,:,:],2))

norm = sum00 + sum01 + sum10 + sum11

S[0,0,:,:,c] = S[0,0,:,:,c]\*mu\_hat[c][0][0]/(sum00/norm)

S[0,1,:,:,c] = S[0,1,:,:,c]\*mu\_hat[c][0][1]/(sum01/norm)

S[1,0,:,:,c] = S[1,0,:,:,c]\*mu\_hat[c][1][0]/(sum10/norm)

S[1,1,:,:,c] = S[1,1,:,:,c] \*mu\_hat[c][1][1]/(sum11/norm)

elif(c == 3): # clique (0,3)

sum00 = np.sum(np.prod(S[0,:,:,0],2))

sum01 = np.sum(np.prod(S[0,:,:,1],2))

sum10 = np.sum(np.prod(S[1,:,:,0],2))

sum11 = np.sum(np.prod(S[1,:,:,1],2))

norm = sum00 + sum01 + sum10 + sum11

S[0,:,:,0,c] = S[0,:,:,0,c] \*mu\_hat[c][0][0]/(sum00/norm)

S[0,:,:,1,c] = S[0,:,:,1,c] \*mu\_hat[c][0][1]/(sum01/norm)

S[1,:,:,0,c] = S[1,:,:,0,c]\*mu\_hat[c][1][0]/(sum10/norm)

S[1,:,:,1,c] = S[1,:,:,1,c]\*mu\_hat[c][1][1]/(sum11/norm)

elif(c == 1): # clique (1,2)

sum00 = np.sum(np.prod(S[:,0,0,:],2))

sum01 = np.sum(np.prod(S[:,0,1,:],2))

sum10 = np.sum(np.prod(S[:,1,0,:],2))

sum11 = np.sum(np.prod(S[:,1,1,:],2))

norm = sum00 + sum01 + sum10 + sum11

S[:,0,0,:,c] = S[:,0,0,:,c]\*mu\_hat[c][0][0]/(sum00/norm)

S[:,0,1,:,c] = S[:,0,1,:,c]\*mu\_hat[c][0][1]/(sum01/norm)

S[:,1,0,:,c] = S[:,1,0,:,c]\*mu\_hat[c][1][0]/(sum10/norm)

S[:,1,1,:,c] = S[:,1,1,:,c]\*mu\_hat[c][1][1]/(sum11/norm)

elif(c == 2): # clique (2,3)

sum00 = np.sum(np.prod(S[:,:,0,0],2))

sum01 = np.sum(np.prod(S[:,:,0,1],2))

sum10 = np.sum(np.prod(S[:,:,1,0],2))

sum11 = np.sum(np.prod(S[:,:,1,1],2))

norm = sum00 + sum01 + sum10 + sum11

S[:,:,0,0,c] = S[:,:,0,0,c]\*mu\_hat[c][0][0]/(sum00/norm)

S[:,:,0,1,c] = S[:,:,0,1,c]\*mu\_hat[c][0][1]/(sum01/norm)

S[:,:,1,0,c] = S[:,:,1,0,c]\*mu\_hat[c][1][0]/(sum10/norm)

S[:,:,1,1,c] = S[:,:,1,1,c]\*mu\_hat[c][1][1]/(sum11/norm)

print "PROBLEM 2(i) RESULTS:"

print "psi\_01 = \n", S[:,:,0,0,0], "\n"

print "psi\_12 = \n", S[0,:,:,0,1], "\n"

print "psi\_23 = \n", S[0,0,:,:,2], "\n"

print "psi\_03 = \n", S[:,0,0,:,3], "\n"

get\_likelihood(S, data)

#----------------------------------------------------------#

#----------------- PROBLEM 2(ii) --------------------#

#----------------------------------------------------------#

def problem2ii():

cliques = [(0,1), (0,2), (0,3), (1,2)]

num\_c = len(cliques)

S = init\_S(num\_c)

mu\_hat = init\_mu(cliques)

# compute mu\_hat for each clique

for i in range(num\_c):

mu\_hat[i] = compute\_mu\_hat(mu\_hat[i], data, cliques[i])

for n in range(10):

for c in range(num\_c):

if(c == 0): # clique (0,1)

sum00 = np.sum(np.prod(S[0,0,:,:],2))

sum01 = np.sum(np.prod(S[0,1,:,:],2))

sum10 = np.sum(np.prod(S[1,0,:,:],2))

sum11 = np.sum(np.prod(S[1,1,:,:],2))

norm = sum00 + sum01 + sum10 + sum11

S[0,0,:,:,c] = S[0,0,:,:,c]\*mu\_hat[c][0][0]/(sum00/norm)

S[0,1,:,:,c] = S[0,1,:,:,c]\*mu\_hat[c][0][1]/(sum01/norm)

S[1,0,:,:,c] = S[1,0,:,:,c]\*mu\_hat[c][1][0]/(sum10/norm)

S[1,1,:,:,c] = S[1,1,:,:,c] \*mu\_hat[c][1][1]/(sum11/norm)

elif(c == 2): # clique (0,3)

sum00 = np.sum(np.prod(S[0,:,:,0],2))

sum01 = np.sum(np.prod(S[0,:,:,1],2))

sum10 = np.sum(np.prod(S[1,:,:,0],2))

sum11 = np.sum(np.prod(S[1,:,:,1],2))

norm = sum00 + sum01 + sum10 + sum11

S[0,:,:,0,c] = S[0,:,:,0,c] \*mu\_hat[c][0][0]/(sum00/norm)

S[0,:,:,1,c] = S[0,:,:,1,c] \*mu\_hat[c][0][1]/(sum01/norm)

S[1,:,:,0,c] = S[1,:,:,0,c]\*mu\_hat[c][1][0]/(sum10/norm)

S[1,:,:,1,c] = S[1,:,:,1,c]\*mu\_hat[c][1][1]/(sum11/norm)

elif(c == 1): # clique (0,2)

sum00 = np.sum(np.prod(S[0,:,0,:],2))

sum01 = np.sum(np.prod(S[0,:,1,:],2))

sum10 = np.sum(np.prod(S[1,:,0,:],2))

sum11 = np.sum(np.prod(S[1,:,1,:],2))

norm = sum00 + sum01 + sum10 + sum11

S[0,:,0,:,c] = S[0,:,0,:,c]\*mu\_hat[c][0][0]/(sum00/norm)

S[0,:,1,:,c] = S[0,:,1,:,c]\*mu\_hat[c][0][1]/(sum01/norm)

S[1,:,0,:,c] = S[1,:,0,:,c]\*mu\_hat[c][1][0]/(sum10/norm)

S[1,:,1,:,c] = S[1,:,1,:,c]\*mu\_hat[c][1][1]/(sum11/norm)

elif(c == 3): # clique (1,2)

sum00 = np.sum(np.prod(S[:,0,0,:],2))

sum01 = np.sum(np.prod(S[:,0,1,:],2))

sum10 = np.sum(np.prod(S[:,1,0,:],2))

sum11 = np.sum(np.prod(S[:,1,1,:],2))

norm = sum00 + sum01 + sum10 + sum11

S[:,0,0,:,c] = S[:,0,0,:,c] \*mu\_hat[c][0][0]/(sum00/norm)

S[:,0,1,:,c] = S[:,0,1,:,c] \*mu\_hat[c][0][1]/(sum01/norm)

S[:,1,0,:,c] = S[:,1,0,:,c] \*mu\_hat[c][1][0]/(sum10/norm)

S[:,1,1,:,c] = S[:,1,1,:,c] \*mu\_hat[c][1][1]/(sum11/norm)

print "\nPROBLEM 2(ii) RESULTS:"

print "psi\_01 = \n", S[:,:,0,0,0], "\n"

print "psi\_02 = \n", S[:,0,:,0,1], "\n"

print "psi\_03 = \n", S[:,0,0,:,2], "\n"

print "psi\_12 = \n", S[0,:,:,0,3], "\n"

get\_likelihood(S, data)

#----------------------------------------------------------#

#----------------- PROBLEM 2(iii) -------------------#

#----------------------------------------------------------#

def problem2iii():

cliques = [(0,1), (0,2), (0,3), (1,2), (1,3), (2,3)]

num\_c = len(cliques)

S = init\_S(num\_c)

mu\_hat = init\_mu(cliques)

# compute mu\_hat for each clique

for i in range(num\_c):

mu\_hat[i] = compute\_mu\_hat(mu\_hat[i], data, cliques[i])

for n in range(10):

for c in range(num\_c):

if(c == 0): # clique (0,1)

sum00 = np.sum(np.prod(S[0,0,:,:],2))

sum01 = np.sum(np.prod(S[0,1,:,:],2))

sum10 = np.sum(np.prod(S[1,0,:,:],2))

sum11 = np.sum(np.prod(S[1,1,:,:],2))

norm = sum00 + sum01 + sum10 + sum11

S[0,0,:,:,c] = S[0,0,:,:,c]\*mu\_hat[c][0][0]/(sum00/norm)

S[0,1,:,:,c] = S[0,1,:,:,c]\*mu\_hat[c][0][1]/(sum01/norm)

S[1,0,:,:,c] = S[1,0,:,:,c]\*mu\_hat[c][1][0]/(sum10/norm)

S[1,1,:,:,c] = S[1,1,:,:,c] \*mu\_hat[c][1][1]/(sum11/norm)

elif(c == 1): # clique (0,2)

sum00 = np.sum(np.prod(S[0,:,0,:],2))

sum01 = np.sum(np.prod(S[0,:,1,:],2))

sum10 = np.sum(np.prod(S[1,:,0,:],2))

sum11 = np.sum(np.prod(S[1,:,1,:],2))

norm = sum00 + sum01 + sum10 + sum11

S[0,:,0,:,c] = S[0,:,0,:,c]\*mu\_hat[c][0][0]/(sum00/norm)

S[0,:,1,:,c] = S[0,:,1,:,c]\*mu\_hat[c][0][1]/(sum01/norm)

S[1,:,0,:,c] = S[1,:,0,:,c]\*mu\_hat[c][1][0]/(sum10/norm)

S[1,:,1,:,c] = S[1,:,1,:,c]\*mu\_hat[c][1][1]/(sum11/norm)

elif(c == 2): # clique (0,3)

sum00 = np.sum(np.prod(S[0,:,:,0],2))

sum01 = np.sum(np.prod(S[0,:,:,1],2))

sum10 = np.sum(np.prod(S[1,:,:,0],2))

sum11 = np.sum(np.prod(S[1,:,:,1],2))

norm = sum00 + sum01 + sum10 + sum11

S[0,:,:,0,c] = S[0,:,:,0,c] \*mu\_hat[c][0][0]/(sum00/norm)

S[0,:,:,1,c] = S[0,:,:,1,c] \*mu\_hat[c][0][1]/(sum01/norm)

S[1,:,:,0,c] = S[1,:,:,0,c]\*mu\_hat[c][1][0]/(sum10/norm)

S[1,:,:,1,c] = S[1,:,:,1,c]\*mu\_hat[c][1][1]/(sum11/norm)

elif(c == 3): # clique (1,2)

sum00 = np.sum(np.prod(S[:,0,0,:],2))

sum01 = np.sum(np.prod(S[:,0,1,:],2))

sum10 = np.sum(np.prod(S[:,1,0,:],2))

sum11 = np.sum(np.prod(S[:,1,1,:],2))

norm = sum00 + sum01 + sum10 + sum11

S[:,0,0,:,c] = S[:,0,0,:,c] \*mu\_hat[c][0][0]/(sum00/norm)

S[:,0,1,:,c] = S[:,0,1,:,c] \*mu\_hat[c][0][1]/(sum01/norm)

S[:,1,0,:,c] = S[:,1,0,:,c] \*mu\_hat[c][1][0]/(sum10/norm)

S[:,1,1,:,c] = S[:,1,1,:,c] \*mu\_hat[c][1][1]/(sum11/norm)

elif(c == 4): # clique (1,3)

sum00 = np.sum(np.prod(S[:,0,:,0],2))

sum01 = np.sum(np.prod(S[:,0,:,1],2))

sum10 = np.sum(np.prod(S[:,1,:,0],2))

sum11 = np.sum(np.prod(S[:,1,:,1],2))

norm = sum00 + sum01 + sum10 + sum11

S[:,0,:,0,c] = S[:,0,:,0,c] \*mu\_hat[c][0][0]/(sum00/norm)

S[:,0,:,1,c] = S[:,0,:,1,c] \*mu\_hat[c][0][1]/(sum01/norm)

S[:,1,:,0,c]= S[:,1,:,0,c] \*mu\_hat[c][1][0]/(sum10/norm)

S[:,1,:,1,c] = S[:,1,:,1,c] \*mu\_hat[c][1][1]/(sum11/norm)

elif(c == 5): # clique (2,3)

sum00 = np.sum(np.prod(S[:,:,0,0],2))

sum01 = np.sum(np.prod(S[:,:,0,1],2))

sum10 = np.sum(np.prod(S[:,:,1,0],2))

sum11 = np.sum(np.prod(S[:,:,1,1],2))

norm = sum00 + sum01 + sum10 + sum11

S[:,:,0,0,c] = S[:,:,0,0,c]\*mu\_hat[c][0][0]/(sum00/norm)

S[:,:,0,1,c] = S[:,:,0,1,c]\*mu\_hat[c][0][1]/(sum01/norm)

S[:,:,1,0,c] = S[:,:,1,0,c]\*mu\_hat[c][1][0]/(sum10/norm)

S[:,:,1,1,c] = S[:,:,1,1,c]\*mu\_hat[c][1][1]/(sum11/norm)

print "\nPROBLEM 2(iii) RESULTS:"

print "psi\_01 = \n", S[:,:,0,0,0], "\n"

print "psi\_02 = \n", S[:,0,:,0,1], "\n"

print "psi\_03 = \n", S[:,0,0,:,2], "\n"

print "psi\_12 = \n", S[0,:,:,0,3], "\n"

print "psi\_13 = \n", S[0,:,0,:,4], "\n"

print "psi\_23 = \n", S[0,0,:,:,5], "\n"

get\_likelihood(S, data)

if \_\_name\_\_ == "\_\_main\_\_":

data = np.loadtxt('Pairwise.dat')

problem2i()

problem2ii()

problem2iii()

PROBLEM 2(i) RESULTS:

psi\_01 =

[[ 1.33373973 0.79959393]

[ 0.79955093 1.06711618]]

psi\_12 =

[[ 1.12519723 0.87480285]

[ 0.42836484 1.57163529]]

psi\_23 =

[[ 0.98864293 1.01161901]

[ 1.00771637 0.99240004]]

psi\_03 =

[[ 1.12558905 0.8743853 ]

[ 0.85647053 1.14356057]]

p(x0) = [[ 0.53333333] [ 0.46666667]]

p(x1) = [[ 0.53333333] [ 0.46666667]]

p(x2) = [[ 0.4] [ 0.6]]

p(x3) = [[ 0.5] [ 0.5]]

likelihood: 2.59471673569e-35

PROBLEM 2(ii) RESULTS:

psi\_01 =

[[ 1.49020152 0.67305716]

[ 0.67084813 1.22356212]]

psi\_02 =

[[ 0.51395982 1.51371315]

[ 1.18240255 0.85681415]]

psi\_03 =

[[ 1.125 0.875 ]

[ 0.85714286 1.14285714]]

psi\_12 =

[[ 1.53933358 0.65947607]

[ 0.44626239 1.41900182]]

p(x0) = [[ 0.53333254] [ 0.46666746]]

p(x1) = [[ 0.53333333] [ 0.46666667]]

p(x2) = [[ 0.4] [ 0.6]]

p(x3) = [[ 0.49999989] [ 0.50000011]]

likelihood: 9.33813428809e-35

*(next page)*

PROBLEM 2(iii) RESULTS:

psi\_01 =

[[ 1.59897249 0.59933317]

[ 0.59390748 1.36168386]]

psi\_02 =

[[ 0.48539894 1.55522356]

[ 1.24804092 0.81400052]]

psi\_03 =

[[ 1.2694663 0.75439129]

[ 0.72468223 1.30500213]]

psi\_12 =

[[ 1.61830979 0.61726944]

[ 0.38527698 1.48500045]]

psi\_13 =

[[ 0.63372356 1.39682085]

[ 1.4497392 0.5884504 ]]

psi\_23 =

[[ 1.30319188 0.78766027]

[ 0.84976384 1.1881377 ]]

p(x0) = [[ 0.53333103] [ 0.46666897]]

p(x1) = [[ 0.53334089] [ 0.46665911]]

p(x2) = [[ 0.4] [ 0.6]]

p(x3) = [[ 0.5] [ 0.5]]

likelihood: 5.95935676296e-34