EE660_HW7_Q2_Hardik_2678294168

November 28, 2021

[33]:
```
import numpy as np
from scipy import stats
from matplotlib import pyplot as plt
```

[10]:
```
def pdf_gaussian(xsample,mean,variance):
    y=stats.norm(mean, variance).pdf(xsample)
    return y
```

[12]:
```
def yhc_t(x_u,mean1,mean2,var1,var2,pi_1,pi_2,c):
    if c==1:
        num=pdf_gaussian(x_u,mean1,var1)*pi_1
    else:
        num=pdf_gaussian(x_u,mean2,var2)*pi_2
    den=pi_1*pdf_gaussian(x_u,mean1,var1)+pi_2*pdf_gaussian(x_u,mean2,var2)
    value=num/den
    return value
```

[19]:
```
#given information
u_ini_1=1.5
u_ini_2=4.0
prior1=0.5
prior2=0.5
variance1=1
variance2=1
x_h=3
x_i=[(1,1),(2,1),(4,2)]
x_i[2][0]
```

[19]:
```
4
```

[25]:
```
#Q2.e.i: Compute Y_hc(t) for both c=1, c=2
y_h1_t=yhc_t(x_h,u_ini_1,u_ini_2,variance1,variance2,prior1,prior2,1)
y_h2_t=yhc_t(x_h,u_ini_1,u_ini_2,variance1,variance2,prior1,prior2,2)
print("--xx--Q2.e.i: Responsibilities--xx--")
print("\nY_h1(t) = ",y_h1_t)
print("\nY_h2(t) = ",y_h2_t)
```

```
--xx--Q2.e.i: Responsibilities--xx--
```

```
Y_h1(t) =  0.3486451353339458

Y_h2(t) =  0.6513548646660543
```

[27]:
```
#Q2.e.ii: Compute u1(t+1) and u2(t+1) according to derived formulas in part␣
 ↪d(pen&paper)
def M_step(yh1,yh2,x_u,x_l):
    x_l1=[]
    x_l2=[]
    for (i,j) in x_l:
        if j==1:
            x_l1.append(i)
        else:
            x_l2.append(i)
    l1=len(x_l1)
    l2=len(x_l2)
    u1_next=(yh1*x_u+np.sum(x_l1))/(yh1+l1)
    u2_next=(yh2*x_u+np.sum(x_l2))/(yh2+l2)

    return u1_next,u2_next
```

[31]:
```
u1_new,u2_new=M_step(y_h1_t,y_h2_t,x_h,x_i)
print("--xx--Q2.e.ii: New Estimates of Mean--xx--")
print("\nu1(t+1) = ",u1_new)
print("\nu2(t=1) = ",u2_new)
```

```
--xx--Q2.e.ii: New Estimates of Mean--xx--

u1(t+1) =  1.722667824582431

u2(t=1) =  3.6055633597580647
```

[79]:
```
#Q2.f: Run for more iterations untill u1(t+1) and u2(t+1) converges.
#I chose the threshold by first running over 20 iterations and calculating the␣
 ↪differnce at 8th
#iteration where the curves seemed to converge.

u_ini_1=1.5
u_ini_2=4.0
u1_t_next=[]
u2_t_next=[]
yh1_t=[]
yh2_t=[]
d1=10
d2=10

while (d1>=7.401362611680895e-08) and (d2>=4.96864274168729e-08):
    y_h1_t=yhc_t(x_h,u_ini_1,u_ini_2,variance1,variance2,prior1,prior2,1)
```

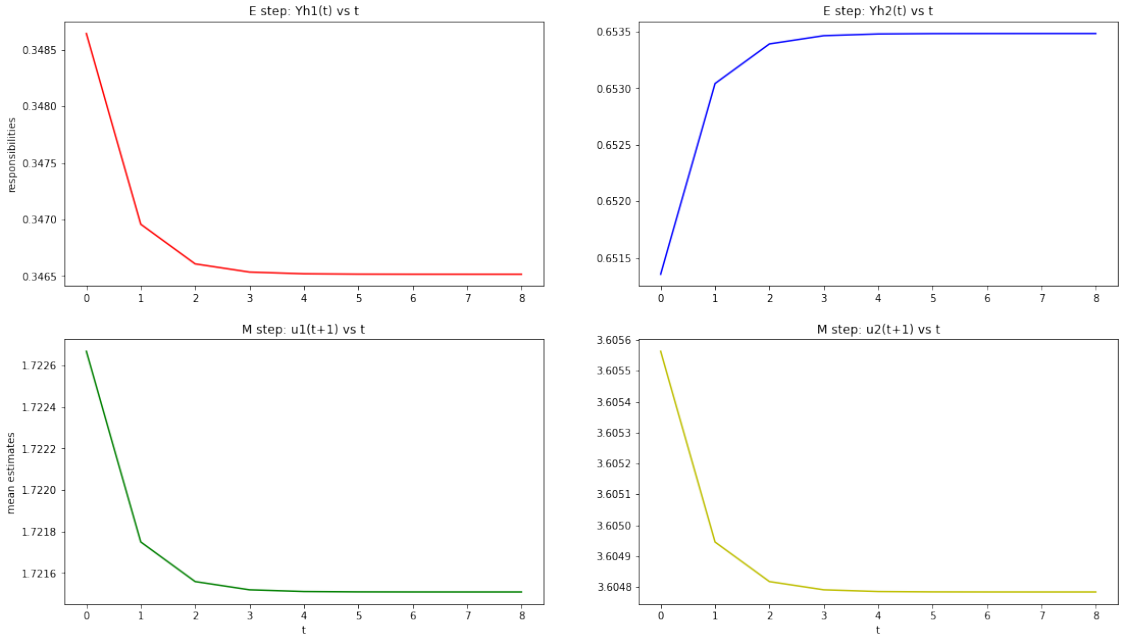
```
    y_h2_t=yhc_t(x_h,u_ini_1,u_ini_2,variance1,variance2,prior1,prior2,2)
    yh1_t.append(y_h1_t)
    yh2_t.append(y_h2_t)

    u1_new,u2_new=M_step(y_h1_t,y_h2_t,x_h,x_i)
    u1_t_next.append(u1_new)
    u2_t_next.append(u2_new)

    d1=abs(u1_new-u_ini_1)
    d2=abs(u2_new-u_ini_2)

    u_ini_1=u1_new
    u_ini_2=u2_new

t=len(u1_t_next)
fig, axs = plt.subplots(2, 2)
fig.set_size_inches(18.5, 10.5)
axs[0, 0].plot(range(t), yh1_t,'r',label='yh1(t)')
axs[0,0].set(ylabel='responsibilities')
axs[0, 0].set_title('E step: Yh1(t) vs t')
axs[0, 1].plot(range(t), yh2_t,'b',label='yh2(t)')
axs[0, 1].set_title('E step: Yh2(t) vs t')
axs[1, 0].plot(range(t),u1_t_next,'g',label='u1(t+1)')
axs[1,0].set(xlabel='t',ylabel='mean estimates')
axs[1, 0].set_title('M step: u1(t+1) vs t')
axs[1, 1].plot(range(t),u2_t_next,'y',label='u2(t+1)')
axs[1,1].set(xlabel='t')
axs[1, 1].set_title('M step: u2(t+1) vs t')
plt.show()
```



[80]:
```
#Q2.f: Report the final answers for u1, u2, yh1, yh2
print("--xx--Q2.f: Final answers--xx--")
print("\nu1(t+1) = ",u1_t_next[-1])
print("\nu2(t+1) = ",u2_t_next[-1])
print("\nYh1(t) = ",yh1_t[-1])
print("\nYh2(t) = ",yh2_t[-1])
```

```
--xx--Q2.f: Final answers--xx--

u1(t+1) =  1.7215089545577031

u2(t+1) =  3.6047836838807368

Yh1(t) =  0.3465162393547647

Yh2(t) =  0.6534837606452354
```

[76]:
```
#Q2.f: I repeat the same task, this time running through the iterations till␣
 ↪the time difference
#in u(t+1)-u(t) equals to 0 for both class.

u_ini_1=1.5
u_ini_2=4.0
u1_t_next=[]
u2_t_next=[]
yh1_t=[]
```

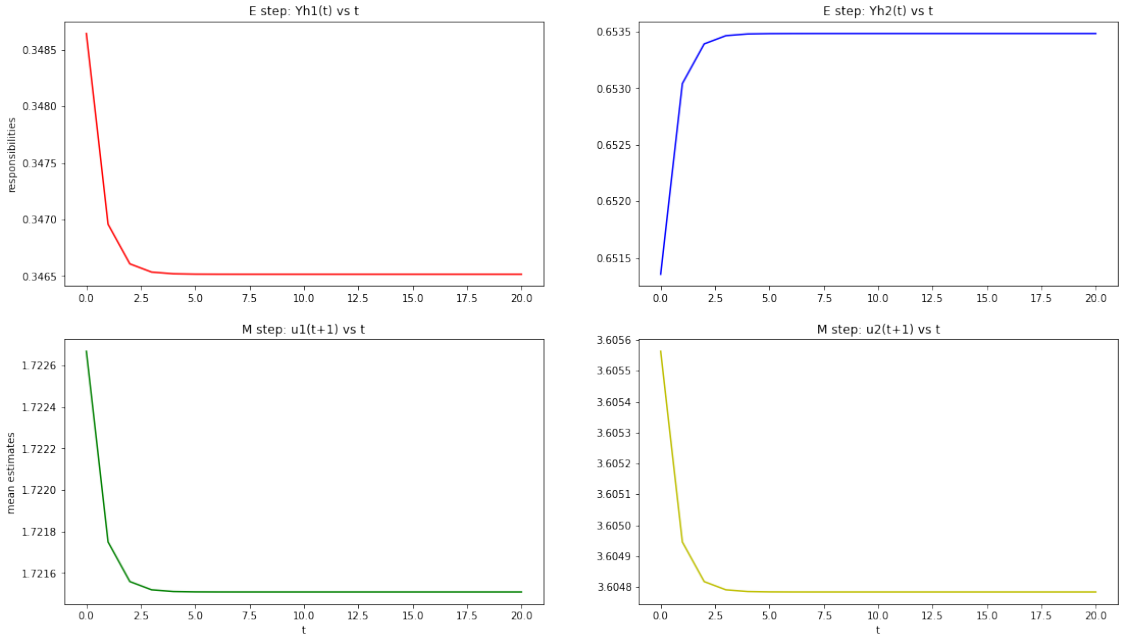
```
yh2_t=[]
d1=10
d2=10
while (d1!=0) and (d2!=0):
    y_h1_t=yhc_t(x_h,u_ini_1,u_ini_2,variance1,variance2,prior1,prior2,1)
    y_h2_t=yhc_t(x_h,u_ini_1,u_ini_2,variance1,variance2,prior1,prior2,2)
    yh1_t.append(y_h1_t)
    yh2_t.append(y_h2_t)

    u1_new,u2_new=M_step(y_h1_t,y_h2_t,x_h,x_i)
    u1_t_next.append(u1_new)
    u2_t_next.append(u2_new)

    d1=u1_new-u_ini_1
    d2=u2_new-u_ini_2

    u_ini_1=u1_new
    u_ini_2=u2_new

t=len(u1_t_next)
fig, axs = plt.subplots(2, 2)
fig.set_size_inches(18.5, 10.5)
axs[0, 0].plot(range(t), yh1_t,'r',label='yh1(t)')
axs[0,0].set(ylabel='responsibilities')
axs[0, 0].set_title('E step: Yh1(t) vs t')
axs[0, 1].plot(range(t), yh2_t,'b',label='yh2(t)')
axs[0, 1].set_title('E step: Yh2(t) vs t')
axs[1, 0].plot(range(t),u1_t_next,'g',label='u1(t+1)')
axs[1,0].set(xlabel='t',ylabel='mean estimates')
axs[1, 0].set_title('M step: u1(t+1) vs t')
axs[1, 1].plot(range(t),u2_t_next,'y',label='u2(t+1)')
axs[1,1].set(xlabel='t')
axs[1, 1].set_title('M step: u2(t+1) vs t')
plt.show()
```


E step: Yh1(t) vs t
responsibilities
E step: Yh2(t) vs t
M step: u1(t+1) vs t
mean estimates
t
M step: u2(t+1) vs t
t

[ ]: