

GMM3

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 from mpl_toolkits.mplot3d import Axes3D
4 import matplotlib.animation as animation
5 def f(x,mu,sigma):
6     F=1/(2*np.pi*np.sqrt(float(np.linalg.det(sigma)))) * np.exp(-1/2 *np.dot(np.dot(np.transpose(x-mu),np.linalg
7     .inv(sigma)),(x-mu)))
8     return F
9 import random
10 random.seed(18)
11 donnees=[]
12 for loop in range(10):
13     donnees.append(np.array(([random.randint(0,10),random.randint(30,40)])))
14     donnees.append(np.array(([random.randint(7,17),random.randint(24,34)])))
15 pourcentage_y=[]
16 pourcentage_z=[]
17 listemoyenne_y=[np.array(([15,50]))]
18 listemoyenne_z=[np.array(([30,20]))]
19 listeMCov_y=[np.array([[4,1],[1,4]])]
20 listeMCov_z=[np.array([[6,2],[2,6]])]
21 for i in range(20):
22     pourcentage_y.append(f(donnees[i],listemoyenne_y[0],listeMCov_y[0])/(f(donnees[i],listemoyenne_y[0],
23     listeMCov_y[0])+f(donnees[i],listemoyenne_z[0],listeMCov_z[0])))
24     pourcentage_z.append(f(donnees[i],listemoyenne_z[0],listeMCov_z[0])/(f(donnees[i],listemoyenne_y[0],
25     listeMCov_y[0])+f(donnees[i],listemoyenne_z[0],listeMCov_z[0])))
26 #On a calcule le pourcentage de chaque point des donnees
27 moyenne_y=np.array([0,0])
28 moyenne_z=np.array([0,0])
29 variance0_y,variance1_y,cov_y=0,0,0
30 variance0_z,variance1_z,cov_z=0,0,0
31 coeff_y=0
32 coeff_z=0
33 #On initialise les variables a recalculer en fonction des points ponderes
34 for i in range(20):
35     coeff_y+=pourcentage_y[i]
36     coeff_z+=pourcentage_z[i]
37 for i in range(20):
38     moyenne_y=moyenne_y+donnees[i]*pourcentage_y[i]/coeff_y
39     moyenne_z=moyenne_z+donnees[i]*pourcentage_z[i]/coeff_z
```

GMM3

```
39 listemoyenne_y.append(moyenne_y)
40 listemoyenne_z.append(moyenne_z)
41 #la moyenne est faite au tour de la variance et covariance
42 for i in range(20):
43     variance0_y+=((donnees[i][0]-moyenne_y[0])**2)*pourcentage_y[i]/coeff_y
44     variance1_y+=((donnees[i][1]-moyenne_y[1])**2)*pourcentage_y[i]/coeff_y
45     cov_y+=(donnees[i][0]-moyenne_y[0])*(donnees[i][1]-moyenne_y[1])*pourcentage_y[i]/coeff_y
46     variance0_z+=((donnees[i][0]-moyenne_z[0])**2)*pourcentage_z[i]/coeff_z
47     variance1_z+=((donnees[i][1]-moyenne_z[1])**2)*pourcentage_z[i]/coeff_z
48     cov_z+=(donnees[i][0]-moyenne_z[0])*(donnees[i][1]-moyenne_z[1])*pourcentage_z[i]/coeff_z
49 MCov_y=np.array([[variance0_y,cov_y],[cov_y,variance1_y]]) #Matrice de covariance de y
50 listeMCov_y.append(MCov_y)
51 MCov_z=np.array([[variance0_z,cov_z],[cov_z,variance1_z]]) #Matrice de covariance de z
52 listeMCov_z.append(MCov_z)
53 #On a utilise des variables pour MCov car le calcul matriciel ne fonctionne pas
54
55
56 while abs(listemoyenne_y[len(listemoyenne_y)-1][0]-listemoyenne_y[len(listemoyenne_y)-2][0])>0.01:
57     coeff_y=0
58     coeff_z=0
59     pourcentage_y=[]
60     pourcentage_z=[] #on reinitialise les pourcentages
61     for i in range(20):
62         pourcentage_y.append(f(donnees[i],moyenne_y,MCov_y)/(f(donnees[i],moyenne_y,MCov_y)+f(donnees[i],
moyenne_z,MCov_z)))
63         pourcentage_z.append(f(donnees[i],moyenne_z,MCov_z)/(f(donnees[i],moyenne_y,MCov_y)+f(donnees[i],
moyenne_z,MCov_z)))
64         for i in range(20):
65             coeff_y+=pourcentage_y[i]
66             coeff_z+=pourcentage_z[i]
67         moyenne_y=np.array([0,0])
68         moyenne_z=np.array([0,0])
69         variance0_y,variance1_y,cov_y=0,0,0
70         variance0_z,variance1_z,cov_z=0,0,0
71         for i in range(20):
72             moyenne_y=moyenne_y+donnees[i]*pourcentage_y[i]/coeff_y
73             moyenne_z=moyenne_z+donnees[i]*pourcentage_z[i]/coeff_z
74         listemoyenne_y.append(moyenne_y)
75         listemoyenne_z.append(moyenne_z) #On ajoute les nouvelles valeurs de moyenne aux listes associees
76         for i in range(20):
77             variance0_y+=((donnees[i][0]-moyenne_y[0])**2)*pourcentage_y[i]/coeff_y
```

GMM3

```
78     variance1_y+=((donnees[i][1]-moyenne_y[1])**2)*pourcentage_y[i]/coeff_y
79     cov_y+=(donnees[i][0]-moyenne_y[0])*(donnees[i][1]-moyenne_y[1])*pourcentage_y[i]/coeff_y
80     variance0_z+=((donnees[i][0]-moyenne_z[0])**2)*pourcentage_z[i]/coeff_z
81     variance1_z+=((donnees[i][1]-moyenne_z[1])**2)*pourcentage_z[i]/coeff_z
82     cov_z+=(donnees[i][0]-moyenne_z[0])*(donnees[i][1]-moyenne_z[1])*pourcentage_z[i]/coeff_z
83     MCov_y=np.array([[variance0_y,cov_y],[cov_y,variance1_y]])
84     listeMCov_y.append(MCov_y)
85     MCov_z=np.array([[variance0_z,cov_z],[cov_z,variance1_z]])
86     listeMCov_z.append(MCov_z) #On ajoute les nouvelles matrices de covariance aux listes associees
87
88 moyenne_y=listemoyenne_y[len(listemoyenne_y)-1]
89 moyenne_z=listemoyenne_z[len(listemoyenne_z)-1]
90 MCov_y=listeMCov_y[len(listeMCov_y)-1]
91 MCov_z=listeMCov_z[len(listeMCov_z)-1]
92
93
94 #La partie graphique
95 a=np.linspace(-20,60,100)
96 A=[]
97 X=[]
98 Y=[]
99 X_deux=[]
100 Y_deux=[]
101 courbe_y,courbe_z=[],[]
102 for i in range(len(a)):
103     for j in range(len(a)):
104         A.append([a[i],a[j]])
105
106 for i in range(len(A)):
107     X.append(A[i][0])
108     Y.append(A[i][1])
109 for i in range(len(A)):
110     X_deux.append(A[i][0])
111     Y_deux.append(A[i][1])
112
113 for i in range(len(A)):
114     elem_y=f(np.array([[A[i][0]], [A[i][1]]]),moyenne_y,MCov_y)
115     elem_z=f(np.array([[A[i][0]], [A[i][1]]]),moyenne_z,MCov_z)
116     courbe_y.append(elem_y[0][0])
117     courbe_z.append(elem_z[0][0])
118 #On rentre l'integralite des valeurs sur le linspace a 2 dimensions
```

GMM3

```
119 for i in range(len(courbe_y)):
120     if courbe_y[i]<0.01*max(courbe_y):
121         X[i],Y[i]=0,0
122 courbe_y=[elem for elem in courbe_y if elem>=0.01*max(courbe_y)]
123 X=[elem for elem in X if elem!=0]
124 Y=[elem for elem in Y if elem!=0]
125 #On enleve les points plus petits que 1% du max
126 for i in range(len(courbe_z)):
127     if courbe_z[i]<0.01*max(courbe_z):
128         X_deux[i],Y_deux[i]=0,0
129 courbe_z=[elem for elem in courbe_z if elem>=0.01*max(courbe_z)]
130 X_deux=[elem for elem in X_deux if elem!=0]
131 Y_deux=[elem for elem in Y_deux if elem!=0]
132 #On fait de meme pour la seconde courbe
133
134 fig=plt.figure()
135 ax = fig.gca(projection='3d')
136 ax.plot(X,Y,courbe_y)
137 ax.plot(X_deux,Y_deux,courbe_z)
138 plt.title("GMM en 3D")
139 plt.show()
140
141 """
142 for angle in range(0, 360):
143     ax.view_init(30, angle)
144     plt.draw()
145     plt.pause(.001)
146 """
```