

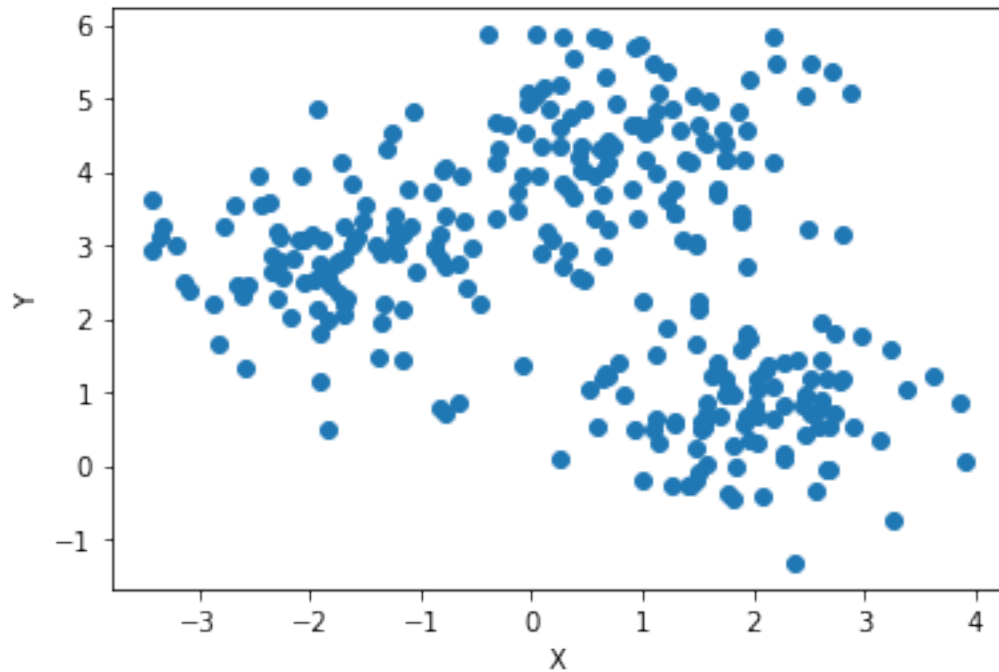
VE414_HW4

November 30, 2021

```
[1]: #Your turn  
#The samples are coming from 3 gaussian distributions,  
#please find the source of each cluster & the corresponding parameter  
#mu & sigma  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from scipy.stats import multivariate_normal as mvn  
from numpy.core.umath_tests import matrix_multiply as mm  
  
df = pd.read_csv('raw_data.csv',index_col=0)  
plt.scatter(df['X'],df['Y'])  
plt.xlabel('X');  
plt.ylabel('Y');
```

<ipython-input-1-306fdb4aa462>:9: DeprecationWarning: numpy.core.umath_tests is an internal NumPy module and should not be imported. It will be removed in a future NumPy release.

```
from numpy.core.umath_tests import matrix_multiply as mm
```



```
[2]: # The initial value of parameters
mus = np.array([[ -2, 3], [ 1, 5], [ 2, 1]])
sigmas = np.array([[ [1,0],[0,1]], [ [1,0],[0,1]], [ [1,0],[0,1]]])
pis = np.array([1/3, 1/3, 1/3])
xs = df.values

def em_gmm_vect(xs, pis, mus, sigmas, tol = 0.01, max_iter = 100):
    n, p = xs.shape
    k = len(pis)
    ll_old = 0
    for i in range(max_iter):
        ws = np.zeros((k,n))
        for j in range(k):
            ws[j, :] = pis[j] * mvn(mus[j], sigmas[j]).pdf(xs)
        ws /= ws.sum(0)

        pis = ws.sum(axis = 1)
        pis /= n
        mus = np.dot(ws, xs)
        mus /= ws.sum(1)[:, None]
        sigmas = np.zeros((k, p, p))
        for j in range(k):
            ys = xs - mus[j,:]
            sigmas[j] = (ws[j,:, None, None] * mm(ys[:, :, None], ys[:, None, :])).
            ↪sum(axis = 0)
```

```

        sigmas /= ws.sum(axis=1)[: ,None,None]
        ll_new = 0
        for pi, mu, sigma in zip(pis, mus, sigmas):
            ll_new += pi*mvn(mu,sigma).pdf(xs)
        ll_new = np.log(ll_new).sum()
        if np.abs(ll_new-ll_old) < tol:
            break
        ll_old = ll_new

    print("Total iteration: %d" %(i+1))
    print(" mu_1:",mus[0], "\n mu_2: ", mus[1], "\n mu_3: ", mus[2])
    print("\n sigma_1:", sigmas[0], "\n sigma_2: ", sigmas[1], "\n sigma_3: ",
    ↪sigmas[2])

em_gmm_vect(xs, pis, mus, sigmas, tol = 0, max_iter = 100)

```

Total iteration: 100

mu_1: [-1.74427754 2.75730959]

mu_2: [0.81315808 4.27595904]

mu_3: [1.93899422 0.7726094]

sigma_1: [[0.62294001 -0.02641632]

[-0.02641632 0.63883231]]

sigma_2: [[0.7917954 0.0651846]

[0.0651846 0.74009291]]

sigma_3: [[0.59709477 -0.01084667]

[-0.01084667 0.48586406]]