homework9

November 12, 2019

In this project, the datasets used are the **x1**, **x2** and **xeasy** datasets. The folders containing the three datasets must be in the same directory as this notebook.

In order to run the code for this project, the following packages must be imported first

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.mixture import GaussianMixture
from scipy.stats import multivariate_normal as mvn
import scipy as sp
from numpy.core.umath_tests import matrix_multiply as mm
```

```
/Users/eff/anaconda3/envs/sta5635/lib/python3.6/site-
packages/ipykernel_launcher.py:7: DeprecationWarning: numpy.core.umath_tests is
an internal NumPy module and should not be imported. It will be removed in a
future NumPy release.
  import sys
```

Helper functions

[2]:
```
def report_package_params(dataset_name, model):
    print(f'{dataset_name} Weights: \n{model.weights_}\n')
    print(f'{dataset_name} Means: \n{model.means_}\n')
    print(f'{dataset_name} Covariances: \n{model.covariances_}\n')

def report_two_step_em_params(dataset_name, pis, mus,sigmas):
    print(f'{dataset_name} Weights: \n{pis}\n')
    print(f'{dataset_name} Means: \n{mus}\n')
    print(f'{dataset_name} Covariances: \n{sigmas}\n')

def loglikelihood(k, pis, mus, sigmas, features, n):
    probs = np.zeros((k, n))
    for j in range(k):
        probs[j, :] = pis[j] * mvn(mus[j], sigmas[j]).pdf(features)
    return probs
```

EM Algorithm for Part 2

[3]:
```
# code from https://people.duke.edu/~ccc14/sta-663/EMAlgorithm.html
def em(features, pis, mus, sigmas, tol=0.01, max_iter=100):
    n, p = features.shape
    k = len(pis)

    ll_old = 0
    for i in range(max_iter):
        ll_new = 0

        # E-step
        probs = np.zeros((k, n))
        for j in range(k):
            probs[j, :] = pis[j] * mvn(mus[j], sigmas[j]).pdf(features)

        probs /= probs.sum(0)

        # M-step
        pis = probs.sum(axis=1)
        pis /= n

        mus = np.dot(probs, features)
        mus /= probs.sum(1)[:, None]

        sigmas = np.zeros((k, p, p))
        for j in range(k):
            ys = features - mus[j, :]
            sigmas[j] = (probs[j, :, None, None] * mm(ys[:, :, None], ys[:,␣
↪None, :])).sum(axis=0)
        sigmas /= probs.sum(axis=1)[:, None, None]

        # update complete log likelihoood
        for pi, mu, sigma in zip(pis, mus, sigmas):
            ll_new += pi * mvn(mu, sigma).pdf(features)
        ll_new = np.log(ll_new).sum()

        if np.abs(ll_new - ll_old) < tol:
            break
        ll_old = ll_new

    return pis, mus, sigmas
```

Load datasets

[4]:
```
x1_data = pd.read_csv('x1.txt', header=None, sep=',').dropna(axis=1).values
x2_data = pd.read_csv('x2.txt', header=None, sep=',').dropna(axis=1).values
```

```
xeasy_data = pd.read_csv('xeasy.txt', header=None, sep=',').dropna(axis=1).
 ↪values
```

Problem 1) Scikit Package EM

A) x1 Dataset

Train Model and Report of Parameters

[5]:
```
x1_GM = GaussianMixture(n_components=2, init_params='kmeans') # initialize␣
 ↪model with k=2 and kmeans result
x1_GM.fit(x1_data) # train model
report_package_params("x1", x1_GM) # report mus, sigmas and pis
```

```
x1 Weights:
[0.4456259 0.5543741]

x1 Means:
[[ 1.83408326  0.28654869]
 [-0.28690175  2.17083353]]

x1 Covariances:
[[[ 1.125035   -0.09991553]
  [-0.09991553  1.23349476]]

 [[ 1.58962506  0.36036065]
  [ 0.36036065  2.07693289]]]
```

B) x2 Dataset

Train Model and Report of Parameters

[6]:
```
x2_GM = GaussianMixture(n_components=2, init_params='kmeans') # initialize␣
 ↪model with k=2 and kmeans result
x2_GM.fit(x2_data) # train model
report_package_params("x2", x2_GM) # report mus, sigmas and pis
```

```
x2 Weights:
[0.56552076 0.43447924]
```

```
x2 Means:
[[ 0.03613016 -0.03944809]
 [ 0.16980465 -0.14409042]]

x2 Covariances:
[[[ 1.33265509  0.10632086]
  [ 0.10632086  0.98474808]]

 [[ 9.92200261  0.80274075]
  [ 0.80274075 10.33702241]]]
```

C) xeasy Dataset

Train Model and Report of Parameters

[7]:
```
xeasy_GM = GaussianMixture(n_components=2, init_params='kmeans') # initialize␣
 ↪model with k=2 and kmeans result
xeasy_GM.fit(xeasy_data) # train model
report_package_params("xeasy", x2_GM) # report mus, sigmas and pis
```

```
xeasy Weights:
[0.56552076 0.43447924]

xeasy Means:
[[ 0.03613016 -0.03944809]
 [ 0.16980465 -0.14409042]]

xeasy Covariances:
[[[ 1.33265509  0.10632086]
  [ 0.10632086  0.98474808]]

 [[ 9.92200261  0.80274075]
  [ 0.80274075 10.33702241]]]
```

Problem 2) Two-Step EM

A) x1 Dataset

Train Model

Initialize parameters

[8]:
```
k = 100
n, m = x1_data.shape
while k != 2:
    random_inits = np.random.random_integers(0, n-1, 3)
    pis = [1/3, 1/3, 1/3]
    mus = x1_data[random_inits]
    small_sig_1 = min(np.linalg.norm(mus[0] - mus[1]), np.linalg.norm(mus[0] -␣
 ↪mus[2]))
    small_sig_2 = min(np.linalg.norm(mus[1] - mus[0]), np.linalg.norm(mus[1] -␣
 ↪mus[2]))
    small_sig_3 = min(np.linalg.norm(mus[2] - mus[0]), np.linalg.norm(mus[2] -␣
 ↪mus[1]))
    sigmas = np.array([(small_sig_1**2)*np.eye(2), (small_sig_2**2)*np.eye(2),␣
 ↪(small_sig_3**2)*np.eye(2)])
    pis, mus, sigmas = em(x1_data, pis, mus, sigmas)
    k = pis[pis > 1/12].size
```

```
/Users/eff/anaconda3/envs/sta5635/lib/python3.6/site-
packages/ipykernel_launcher.py:4: DeprecationWarning: This function is
deprecated. Please call randint(0, 599 + 1) instead
  after removing the cwd from sys.path.
```

[9]:
```
pis, mus, sigmas = em(x1_data, pis, mus, sigmas)
```

Report Parameters

[10]:
```
report_two_step_em_params("x1", pis, mus, sigmas)
```

```
x1 Weights:
[0.16296047 0.04523075 0.79180878]

x1 Means:
[[-0.24664456  2.98884348]
 [ 0.05067474  4.62234051]
 [ 0.87920894  0.80197662]]

x1 Covariances:
[[[ 1.63689637  0.33406578]
  [ 0.33406578  0.37176186]]

 [[ 1.13194552  0.12308863]
  [ 0.12308863  0.20889399]]
```

```
 [[ 2.50968912 -0.58675936]
  [-0.58675936  1.70332654]]]
```

B) x2 Dataset

Train Model

Initialize parameters

[11]:
```
k = 100
n, m = x2_data.shape
while k != 2:
    random_inits = np.random.random_integers(0, n-1, 3)
    pis = [1/3, 1/3, 1/3]
    mus = x2_data[random_inits]
    small_sig_1 = min(np.linalg.norm(mus[0] - mus[1]), np.linalg.norm(mus[0] -␣
 ↪mus[2]))
    small_sig_2 = min(np.linalg.norm(mus[1] - mus[0]), np.linalg.norm(mus[1] -␣
 ↪mus[2]))
    small_sig_3 = min(np.linalg.norm(mus[2] - mus[0]), np.linalg.norm(mus[2] -␣
 ↪mus[1]))
    sigmas = np.array([(small_sig_1**2)*np.eye(2), (small_sig_2**2)*np.eye(2),␣
 ↪(small_sig_3**2)*np.eye(2)])
    pis, mus, sigmas = em(x2_data, pis, mus, sigmas)
    k = pis[pis > 1/12].size
```

```
/Users/eff/anaconda3/envs/sta5635/lib/python3.6/site-
packages/ipykernel_launcher.py:4: DeprecationWarning: This function is
deprecated. Please call randint(0, 599 + 1) instead
  after removing the cwd from sys.path.
```

[12]:
```
pis, mus, sigmas = em(x2_data, pis, mus, sigmas)
```

Report Parameters

[13]:
```
report_two_step_em_params("x2", pis, mus, sigmas)
```

```
x2 Weights:
[0.05716833 0.5012003  0.44163137]

x2 Means:
```

```
[[-2.07676046 -2.64303131]
 [ 0.02229069 -0.06107485]
 [ 0.45685564  0.21917695]]

x2 Covariances:
[[[2.40698275e+00 1.51767226e-01]
  [1.51767226e-01 1.26465221e+01]]

 [[9.97517494e-01 5.54089832e-02]
  [5.54089832e-02 8.82597219e-01]]

 [[9.28662653e+00 8.34212335e-03]
  [8.34212335e-03 7.85776950e+00]]]
```

C) xEasy Dataset

Train Model

Initialize parameters

[14]:
```
k = 100
n, m = xeasy_data.shape
while k != 2:
    random_inits = np.random.random_integers(0, n-1, 3)
    pis = [1/3, 1/3, 1/3]
    mus = xeasy_data[random_inits]
    small_sig_1 = min(np.linalg.norm(mus[0] - mus[1]), np.linalg.norm(mus[0] -␣
 ↪mus[2]))
    small_sig_2 = min(np.linalg.norm(mus[1] - mus[0]), np.linalg.norm(mus[1] -␣
 ↪mus[2]))
    small_sig_3 = min(np.linalg.norm(mus[2] - mus[0]), np.linalg.norm(mus[2] -␣
 ↪mus[1]))
    sigmas = np.array([(small_sig_1**2)*np.eye(2), (small_sig_2**2)*np.eye(2),␣
 ↪(small_sig_3**2)*np.eye(2)])
    pis, mus, sigmas = em(xeasy_data, pis, mus, sigmas)
    k = pis[pis > 1/12].size
```

```
/Users/eff/anaconda3/envs/sta5635/lib/python3.6/site-
packages/ipykernel_launcher.py:4: DeprecationWarning: This function is
deprecated. Please call randint(0, 499 + 1) instead
  after removing the cwd from sys.path.
```

[15]:
```
pis, mus, sigmas = em(xeasy_data, pis, mus, sigmas)
```

Report Parameters

[16]:
```
report_two_step_em_params("xeasy", pis, mus, sigmas)
```

```
xeasy Weights:
[0.4092564  0.01920679 0.5715368 ]

xeasy Means:
[[ 0.02906585  3.06797625]
 [ 1.6113283  -1.16412929]
 [ 3.06790619 -0.14447968]]

xeasy Covariances:
[[[ 1.0184625  -0.05977653]
  [-0.05977653  0.96021343]]

 [[ 0.03102199 -0.12959524]
  [-0.12959524  0.78159524]]

 [[ 0.96510282  0.12274953]
  [ 0.12274953  0.91380089]]]
```