40.319 Statistical and Machine Learning
Spring 2020 Homework 3

Due on Friday 3 APR, 4 PM, Total: 30 pts

Submit your written/typed solutions as a PDF using Gradescope.Upload your Python files using this link:
`https://www.dropbox.com/sh/jwmk8ezrqtncon1/AABJJxAq1VZlX2JOyNSxcWNEa?dl=0`

Problem 1 (10 points)

In this problem, we will implement the EM algorithm for clustering. Start by importing the required packages and preparing the dataset.

```
import numpy as np
import matplotlib.pyplot as plt

from numpy import linalg as LA
from matplotlib.patches import Ellipse
from sklearn.datasets.samples_generator import make_blobs
from scipy.stats import multivariate_normal

K = 3
NUM_DATAPTS = 150

X, y = make_blobs(n_samples=NUM_DATAPTS, centers=K, shuffle=False,
                                    random_state=0, cluster_std=0.6)
g1 = np.asarray([[2.0, 0], [-0.9, 1]])
g2 = np.asarray([[1.4, 0], [0.5, 0.7]])
mean1 = np.mean(X[:int(NUM_DATAPTS/K)])
mean2 = np.mean(X[int(NUM_DATAPTS/K):2*int(NUM_DATAPTS/K)])
X[:int(NUM_DATAPTS/K)] = np.einsum('nj,ij->ni',
                    X[:int(NUM_DATAPTS/K)] - mean1, g1) + mean1
X[int(NUM_DATAPTS/K):2*int(NUM_DATAPTS/K)] = np.einsum('nj,ij->ni',
                    X[int(NUM_DATAPTS/K):2*int(NUM_DATAPTS/K)] - mean2, g2) + mean2
X[:,1] -= 4
```

(a) Randomly initialize a numpy array *mu* of shape (K, 2) to represent the mean of the clusters, and initialize an array *cov* of shape (K, 2, 2) such that *cov*[k] is the identity matrix for each k. *cov* will be used to represent the covariance matrices of the clusters. Finally, set π to be the uniform distribution at the start of the program.

(b) Write a function to perform the E-step:

```
def E_step():
        gamma = np.zeros((NUM_DATAPTS, K))
```

```
    ...
    ...
    return gamma
```

(c) Write a function to perform the M-step:

```
def M_step(gamma):
    ...
    ...
```

(d) Now write a loop that iterates through the E and M steps, and terminates after the change in log-likelihood is below some threshold. At each iteration, print out the log-likelihood, and use the following function to plot the progress of the algorithm:

```
def plot_result(gamma=None):
    ax = plt.subplot(111, aspect='equal')
    ax.set_xlim([-5, 5])
    ax.set_ylim([-5, 5])
    ax.scatter(X[:, 0], X[:, 1], c=gamma, s=50, cmap=None)

    for k in range(K):
        l, v = LA.eig(cov[k])
        theta = np.arctan(v[1,0]/v[0,0])

        e = Ellipse((mu[k, 0], mu[k, 1]), 6*l[0], 6*l[1],
                theta * 180 / np.pi)
        e.set_alpha(0.5)
        ax.add_artist(e)

    plt.show()
```

(e) Use sklearn's KMeans module

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

to perform K-means clustering on the dataset, and compare both clustering results.

NOTE: Append the print-outs from your program (plots of images with their iteration numbers and log-likelihoods) to your PDF submission on Gradescope. Upload the final script as a file named `[student-id]-em.py` using the Dropbox link at the start of this assignment.

Problem 2 (3 points)

Let p and q be distributions on $\{1, 2, 3, 4, 5\}$ such that $p_1 = \frac{1}{8}$, $p_2 = \frac{1}{2}$, $p_3 = p_4 = p_5 = \frac{1}{8}$, and $q_1 = \frac{1}{4}$, $q_2 = q_3 = \frac{1}{8}$, $q_4 = q_5 = \frac{1}{4}$.

(a) Compute the cross-entropy $H(p, q)$ in bits. Is $H(q, p) = H(p, q)$?

(b) Compute the entropies $H(p)$ and $H(q)$ in bits.

(c) Compute the KL-divergence $D_{KL}(p \,|\, q)$ in bits.

Show all working and leave your answers in fractions.

Problem 3 (8 points)

(a) Perform singular value decomposition (SVD) on the following matrix

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = U\Sigma V^T.$$

(b) For a general design matrix X, why are the columns of the transformed matrix $T = XV$ orthogonal?

Problem 4 (4 points)

In this problem, we will perform principal component analysis (PCA) on sklearn's diabetes dataset. Start by importing the required packages and load the dataset.

```
import numpy as np
from sklearn import decomposition
from sklearn import datasets

X = datasets.load_diabetes().data
```

You can find out more on how to use sklearn's PCA module from:
https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

For this problem, make sure the design matrix is first normalized to have zero mean and unit standard deviation for each column.

(a) Write code to print the matrix V that will be used to transform the dataset, and print all the singular values.

(b) Now perform PCA on the dataset and print out the 3 most important components for the first 10 data-points.

NOTE: As this problem is short, **present and include your code and results in your PDF submission in Gradescope**. You **DO NOT** need to upload py file for this question through the Dropbox link at the start of this assignment.

Problem 5 (5 points)

An AR(2) model assumes the form

$$r_t = \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + a_t,$$

where a_t is a white noise sequence. Show that if the model is stationary, then

(a) $E(r_t) = \frac{\phi_0}{1-\phi_1-\phi_2}$ (assume $\phi_1 + \phi_2 \neq 1$);

(b) the ACF is given by

$$\rho(1) = \frac{\phi_1}{1-\phi_2}, \qquad \rho(s) = \phi_1 \rho(s-1) + \phi_2 \rho(s-2), \quad \forall\, s \geq 2.$$