Homework #6

ECE 461/661: Introduction to Machine Learning
Prof. Gauri Joshi and Prof. Carlee Joe-Wong
Due: April 12, 2020 at 8:59PM PT / 11:59PM ET

Please remember to show your work for all problems and to write down the names of any students that you collaborate with. The full collaboration and grading policies are available on the course website: https://www.andrew.cmu.edu/course/18-661/.

Your solutions should be uploaded to Gradescope (https://www.gradescope.com/) in PDF format by the deadline. We will not accept hardcopies. If you choose to hand-write your solutions, please make sure the uploaded copies are legible. Gradescope will ask you to identify which page(s) contain your solutions to which problems, so make sure you leave enough time to finish this before the deadline. We will give you a 30-minute grace period to upload your solutions in case of technical problems.

1 EM *[25 points]*

Suppose $X_1, \ldots, X_N$ are i.i.d random variables with the density function $f_X(x, \lambda) = \lambda e^{-\lambda x}$, for $x \geq 0$ and 0 otherwise. We observe $Y_i = \min\{X_i, c_i\}$ for some fixed and known c_i. In this problem, you will estimate the value of λ using the EM algorithm. In particular, the variables X_i's correspond to the missing or latent variables.

(a) Show that the log-likelihood $\mathcal{L}(X_1, \ldots, X_n|\lambda)$ in terms of the latent (unobserved) variables X_i is

$$\mathcal{L}(X_1, \ldots, X_n|\lambda) = n \log \lambda - \lambda \sum_{i=1}^{n} X_i.$$

(b) What is the conditional expectation of X_i assuming that you observe $Y_i = y_i < c_i$?

(c) Now suppose that $Y_i = c_i$. What is the conditional expectation of X_i given that you observe $Y_i = c_i$? Hint: You may want to use the memorylessness property of the exponential distribution.

(d) Suppose we observe $y_1, ..., y_N$. Let I_i be an indicator denoting if $y_i < c_i$. In other words, $I_i = 1$ if $y_i < c_i$ (as in part (b) above)), and $I_i = 0$ if $y_i = c_i$ (as in part (c) above). After t rounds of the EM algorithm, we have got an estimator λ^t. Using your results from parts (a), (b), and (c) above, write down the E-Step for the $t + 1$st round, i.e., derive $Q(\lambda|\lambda^t) = \mathbb{E}[\mathcal{L}|\boldsymbol{y}, \lambda^t]$.

(e) Now write down the M-Step from the $t + 1$st round and calculate the value of λ^{t+1}.

2 3-Dimensional Principal Component Analysis *[25 points]*

In this problem, we will perform PCA on 3-dimensional data step by step. We are given three data points:

$$\boldsymbol{x}_1 = [0, -1, -2], \boldsymbol{x}_2 = [1, 1, 1], \boldsymbol{x}_3 = [2, 0, 1],$$

and we want to find 2 principal components of the given data.

a. First, find the covariance matrix $\boldsymbol{C}_X = X^T X$ where $X = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \end{bmatrix}$. Then, find the eigenvalues and the corresponding eigenvectors of $\boldsymbol{C}_X$. (Feel free to use any numerical analysis program such as numpy, e.g., `numpy.linalg.eig` can be useful. Also make sure to center the data X first.)

b. Using the result above, find the first two principal components of the given data.

c. Now we want to represent the data $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_3$ using a 2-dimensional subspace instead of a 3-dimensional one. PCA gives us the 2-D plane which minimizes the difference between the original data and the data projected to the 2-dimensional plane. In other words, $\boldsymbol{x}_i$ can be represented as:

$$\widetilde{\boldsymbol{x}}_i = a_{i1}\boldsymbol{u}_1 + a_{i2}\boldsymbol{u}_2 + b_3\boldsymbol{u}_3, \tag{1}$$

where $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are the principal components we found in 3.b., and $\boldsymbol{u}_3$ is a vector orthonormal to $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. Figure 1 gives an example of what this might look like.

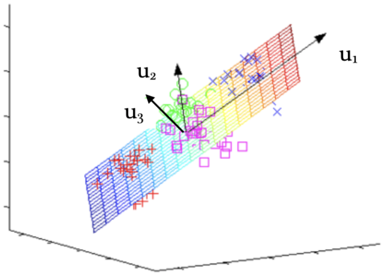

Figure 1: Example of 2-D plane spanned by the first two principal components.

Find a_{i1}, a_{i2} for $i = 1, 2, 3$ and b_3. Then, find the $\widetilde{\boldsymbol{x}}_i$'s and the difference between $\widetilde{\boldsymbol{x}}_i$ and $\boldsymbol{x}_i$, i.e., $||\widetilde{\boldsymbol{x}}_i - \boldsymbol{x}_i||^2$ for $i = 1, 2, 3$. (Again, feel free to use any numerical analysis program to get the final answer. But, show your calculation process.)

Hint: b_3 can be easily obtained from $\overline{\boldsymbol{x}} = \frac{1}{3}(\boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{x}_3)$.

3 GMM *[25 points]*

In this problem, we have a dataset that has K components $\{C_1, C_2, \ldots, C_K\}$. Each component is generated from a normal distribution $\sim \mathcal{N}(\mu_i, \Sigma_i)$. In other words, each data point x is generated as follows:

- Choose a component $y = 1, 2, \ldots, K$ with a probability $P(y = i) = \pi_i$ for $i = 1, 2, \ldots, K$.
- Draw a sample $x \sim \mathcal{N}(\mu_i, \Sigma_i)$.

Note that our dataset therefore satisfies the assumptions of a Gaussian mixture model. Now we can find the following probability distributions of x:

$$p(x|y = i) \sim \mathcal{N}(\mu_i, \Sigma_i), \quad p(x) = \sum_{k=1}^{K} p(x|y = i)P(y = i).$$

At test time, we wish to assign a cluster to each data point x. We can do so by finding the component that yields the maximum probability for the test point x:

$$\operatorname{argmax}_k p(y = k|x)$$

Now suppose that we are given a set D of N example data points, each consisting of a d dimensional vector. More formally, our data $D = \{x_1, x_2, x_3 \dots x_N\}$ where each $x_i \in \mathbb{R}^d$. Our task is to assign each example into one of the K clusters using the above formulation. The parameters for the above formulation are $\theta = (\pi_1, \pi_2 \dots \pi_K, \mu_1, \mu_2 \dots \mu_K, \Sigma_1, \Sigma_2 \dots \Sigma_K)$, hence the problem reduces to estimating these parameters.

Dataset In the file *gmm_data.txt*, each row is a unique vector of 5 dimensions. There are 3000 unique examples. Hence $N = 3000, d = 5$, and we want to group these points into three clusters ($K = 3$). They are generated as per the generation process outlined above. To visualize the dataset, we have plotted the first 2 dimensions (i.e x_{i1}, x_{i2}) of each i from 1 to 3000 (Figure 2)

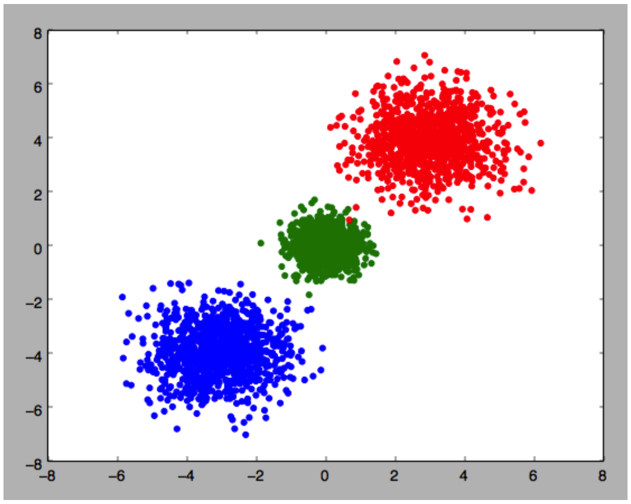

Figure 2: The first two dimensions of the dataset, different colors denote different clusters.

Convention For consistency, we refer to the leftmost cluster (i.e., one with the least μ_{i1}) as the Blue cluster (denoted by blue color), the middle cluster as the Green cluster (denoted by green color), and the rightmost (the one with the largest μ_{i1}) cluster as the Red cluster (denoted by red color). Figure 2 also adheres to this convention.

Problems Now, please answer the following questions. **Please include your code in the final PDF you turn in for full credit.**

(a) Assuming that $\Sigma_i = \sigma_i^2 I$, compute and write below the means μ_1, μ_2, μ_3 and the standard deviations $\sigma_1, \sigma_2, \sigma_3$. We highly recommend that you use the GaussianMixture model from sklearn. (Please use random initialization for this part, and a default convergence threshold/tolerance of 0.001)

(b) From these computed means and std. deviations, cluster all the 3000 points, and plot three figures

- Plot the first two dimensions (x_{i1}, x_{i2}) with their cluster assignments. (similar to Figure 2)
- Plot the third and fourth dimensions (x_{i3}, x_{i4}) with their cluster assignments.
- Plot the fourth and fifth dimensions (x_{i4}, x_{i5}) with their cluster assignments.

(c) For this subquestion, you are not permitted to use any library function that performs the EM algorithm, and you are instead supposed to write your own EM algorithm to estimate the means μ_1, μ_2, μ_3. You can assume that values of π_1, π_2, π_3 and $\sigma_1, \sigma_2, \sigma_3$ are known and reuse the values that you computed in part (a) (They can be found in GaussianMixture.weights_ and GaussianMixture.covariances_). Write down your E step and M step to estimate μ_1, μ_2, μ_3. For the programming, please use a convergence threshold/tolerance of 0.001.

(d) We now relax the assumption that π_1, π_2, π_3 are known. Modify your EM algorithm to simultaneously estimate means μ_1, μ_2, μ_3 and the component probabilities π_1, π_2, π_3. You can still assume that σ_1, σ_2 and σ_3 are known and use their values from the solution above. Write down your E step and M step. For the programming, please use convergence threshold/tolerance of 0.001. For this part too, you are not permitted to use library functions that perform EM algorithm.