CS5785 / ORIE5750 / ECE5414 Homework 3

This homework is due on **Wednexday, October 4th, 2023 at 11:59PM ET**. Upload your homework to Gradescope (Canvas->Gradescope). The homework is split into programming exercises and written exercises. Upload your homework to Gradescope. Your submission will have two parts:

1. A write-up as a single `.pdf` file. Submit this under the `hw3-report` assignment in Gradescope.
2. Source code and data files for all of your experiments (AND figures) in `.ipynb` files (file format for IPython Jupyter Notebook). These files should be placed in a folder titled `hw3` and uploaded to the `hw3-code` assignment in Gradescope.

The write-up should contain a general summary of what you did, how well your solution works, any insights you found, etc. On the cover page, include the class name, and homework number. You are responsible for submitting clear, organized answers to the questions. You could use online LaTeX templates from Overleaf, under "Homework Assignment" and "Project / Lab Report".

Please include all relevant information for a question, including text response, equations, figures, graphs, output, etc. If you include graphs, be sure to include the source code that generated them. Please pay attention to Canvas for announcements, policy changes, etc. and Piazza for homework related questions.

IF YOU NEED HELP

There are several strategies available to you.

- If you get stuck, we encourage you to post a question on Piazza. That way, your solutions will be available to other students in the class.
- The professor and TAs offer office hours, which are a great way to get some one-on-one help.
- You are allowed to use well known libraries such as `scikit-learn`, `scikit-image`, `numpy`, `scipy`, etc. for this assignment (including implementations of machine learning algorithms), unless we explicitly say that you cannot in a particular question. Any reference or copy of public code repositories should be properly cited in your submission (examples include Github, Wikipedia, Blogs).

PROGRAMMING EXERCISES

1. **Implement EM algorithm.** In this problem, you will implement a bimodal GMM model fit using the EM algorithm. Bimodal means that the distribution has two peaks, or that the data is a mixture of two groups. If you want, you can assume the covariance matrix is diagonal (i.e. it has the form $\mathrm{diag}(\sigma_1^2, \sigma_2^2, ..., \sigma_d^2)$ for scalars σ_i) and you can randomly initialize the parameters of the model.

   You will need to use the Old Faithful Geyser Dataset. The data file contains 272 observations of the waiting time between eruptions and the duration of each eruption for the Old Faithful geyser in

Yellowstone National Park.

You should do this without calling the Gaussian Mixture library in `scikit learn`. You can use `numpy` or `scipy` for matrix calculation or generating Gaussian distributions.

(a) Treat each data entry as a 2-dimensional feature vector. Parse and plot all data points on the 2-D plane.

(b) Recall that EM learns the parameter θ of a Gaussian mixture model $P_\theta(x, z)$ over a dataset $D = \{x^{(i)} | i = 1, 2, \ldots n\}$ by performing the E-step and the M-step for $t = 0, 1, 2, \ldots$. We repeat the E-step and M-step until convergence.

In the E-step, for each $x^{(i)} \in D$, we compute a vector of probabilities $P_{\theta_t}(z = k \mid x)$ for the event that each $x^{(i)}$ originates from a cluster k given the current set of parameters θ_t.

Write the expression for $P_{\theta_t}(z = k \mid x)$, which is the posterior of each data point $x^{(i)}$. Recall that by Bayes' rule,

$$P_{\theta_t}(z = k \mid x) = \frac{P_{\theta_t}(z = k, x)}{P_{\theta_t}(x)} = \frac{P_{\theta_t}(z = k, x)}{\sum_{l=1}^{K} P_{\theta_t}(x|z = l) P_{\theta_t}(z = l)}.$$

Note that we have seen this formula in class. We are asking you to write it down and try to understand it before implementing it in part (e).

(c) In the M-step, we compute new parameters θ_{t+1}. Our goal is to find μ_k, Σ_k and ϕ_k that optimize

$$\max_\theta \left(\sum_{k=1}^{K} \sum_{x \in D} P_{\theta_t}(z_k|x) \log P_\theta(x|z_k) + \sum_{k=1}^{K} \sum_{x \in D} P_{\theta_t}(z_k|x) \log P_\theta(z_k) \right)$$

Write down the formula for μ_k, Σ_k, and for the parameters ϕ at the M-step (we have also seen these formulas in class).

(d) Implement and run the EM algorithm. Specifically:

i. Implement the EM algorithm from scratch (e.g., in Python and `numpy`).

ii. Choose a termination criterion for when the algorithm stops repeating the E-step and the M-step. State your termination criterion and explain the reasoning behind it.

iii. Plot the trajectories of the two mean vectors (μ_1 and μ_2) in two dimensions as they change over the course of running EM. You might want to use a scatter plot for this.

(e) If you were to run K-means clustering instead of the EM algorithm you just implemented, do you think you would get different clusters? Please experiment with K-means clustering on the same dataset with $K = 2$. (The KNN library from `scikit learn` is a good way to try). Comment on why you think the results will or will not change.

WRITTEN EXERCISES

1. **True/false questions.** Please provide responses to the following statements, indicating whether they are true or false, and briefly provide explanations for your assessment.

(a) Training and testing split is beneficial in the context of the K-means algorithm.

(b) There exist scenarios where the K-means algorithm will not converge and thus the algorithm will not terminate.

(c) There exist scenarios where you would obtain different cluster assignments by running the same K-means algorithm multiple times, potentially with different cluster initialization.

(d) Training and testing split is beneficial in the context of GMMs for clustering.

(e) GMM for clustering will yield consistent cluster partitions across repeated runs.

2. **Weights for clustering.** In clustering algorithms like K-means, we need to compute distances in the feature space. Sometimes people use weights to value some feature more than others. Show that weighted Euclidean distance for p dimensional data points x_i and $x_{i'}$

$$d_e^{(w)}(x_i, x_{i'}) = \frac{\sum_{l=1}^{p} w_l (x_{il} - x_{i'l})^2}{\sum_{l=1}^{p} w_l} \tag{1}$$

satisfies

$$d_e^{(w)}(x_i, x_{i'}) = d_e(z_i, z_{i'}) = \sum_{l=1}^{p} (z_{il} - z_{i'l})^2, \tag{2}$$

where

$$z_{il} = x_{il} \cdot \left(\frac{w_l}{\sum_{l=1}^{p} w_l} \right)^{1/2}. \tag{3}$$

Thus weighted Euclidean distance based on x is equivalent to unweighted Euclidean distance based on a proper transformed data z.