Assignment 4: Gaussian Mixture Models

Important Information

School of Engineering Technology and Applied Science

Information and Communication Engineering Technology

Unsupervised and Reinforcement Learning (COMP257)

- **Gaussian Mixture Models (15%)**
- **Due Date: Friday of Week 7 by 11:59 pm EST (late penalty at 10 points per day)**
- **Upload your assignment here: Assignment 4: Gaussian Mixture Models (/d2l/common/dialogs/quickLink/quickLink.d2l?ou=694716&type=dropbox&rcode=CENCOL-3439544)**

Instructions

- You are free to choose any toolkits to solve the problems at hand (e.g., TensorFlow, Sci-Learn, etc.)
- All written reports and codes are to be maintained on a repository of your choice such as Github. The course instructor will discuss and exchange with you information to get access to your code.
- The video presentation will be required as part of the submission that documents the steps taken to obtain the results.
- **IMPORTANT NOTES:**
  - 1 point will be deducted for each incident that does not conform to the requirements (e.g., code not properly formatted, comments not relevant to support documentation of code, missing code documentation, etc.).
  - All points will be deducted for submission of *nonsensical code* (i.e., code that doesn't contribute to the relevancy of the task at hand). This is question-specific.

Questions

- **Read the question below carefully.**

Question 1

[100 points]

In this assignment, you train a Gaussian mixture model on the Olivetti faces dataset.

1. Use PCA preserving 99% of the variance to reduce the dataset's dimensionality. [10 points]
2. Determine the most suitable *covariance_type* for the dataset. [15 points]
3. Determine the minimum number of clusters that best represent the dataset using either AIC or BIC. [15 points]
4. Plot the results from (2) and (3). [15 points]
5. Output the hard clustering for each instance. [2.5 points]
6. Output the soft clustering for each instance. [2.5 points]
7. Use the model to generate some new faces (using the *sample()* method), and visualize them (use the *inverse_transform()* method to transform the data back to its original space based on the PCA method used). [15 points]
8. Modify some images (e.g., rotate, flip, darken). [15 points]
9. Determine if the model can detect the anomalies produced in (8) by comparing the output of the *score_samples()* method for normal images and for anomalies). [10 points]