Data Mining, Homework 3

Out: November 26, 2024, Due: December 08, 2024, Total: 70

Note:

- You must use Python 3 version.
- The entire code will need to be your own, however you are allowed to use numpy libraries for mean, and variance of a vector, co-variance of a matrix, and mutivariate normal distribution pdf.
- Homeworks are individual work, please do not collaborate with others inside or outside of the class. Software will be used to determine code similarity, so do not take a chance.
- Start early and if you need help, post your questions on Canvas or use instructor/TA's office hour.
- Print all fractional numbers rounded with 3 digits after the decimal point.

Clustering via Expectation-Maximization

Write a python script that implements the Expectation-Maximization (EM) algorithm for clustering. Run the code on the `iris.txt` dataset (`https://archive.ics.uci.edu/ml/datasets/iris`. Use the first four attributes for clustering, and use the labels only for the purity-based clustering evaluation (see below). In your implementation, you should estimate the full covariance matrix for each cluster.

For EM initialization, use the first 40 points for cluster 1, the next 40 for cluster 2, and the rest for cluster 3; your initial model parameters (mean, covariance matrix) will be based on the above initial clustering. For convergence testing, you can compare the sum of the Euclidean distance between the old means and the new means over the k clusters. If this distance is less than 0.000001 you can stop the method.

Your program output should consist of the following information:

a. The final mean for each cluster. Print the word "Mean:" and in the following lines, print the mean vectors one in a line, sorted in increasing order of the norm of these vectors.

b. The final covariance matrix for each cluster. Print the word "Covariance Matrices:" and in the following lines, print the matrices with an empty line between them. Sort the covariance matrix in the same order with their corresponding mean vectors as you have printed earlier.

c. Number of iterations the EM algorithm took to converge. Print "Iteration count=" followed by an integer number.

d. Final cluster assignment of all the points, where each point will be assigned to the cluster that yields the highest probability $P(C_i|x_j)$. Print "Cluster Membership:" and in the following lines, print membership. In a single line, print the members of each cluster in ascending sorted order and separated by comma. For k clusters your output should have k different membership lines which are sorted in the same order as the mean and covariance that you have printed earlier.

e. Final size of each cluster. Print "Size:" followed by k integers separated by a single space. The integers are sorted in the same order as above.

Finally, you must print the 'purity score' for your clustering, computed as follows: Assume that C_i denotes the set of points assigned to cluster i by the EM algorithm, and let T_j denote the true assignments of the points based on the last attribute. Purity score is defined as:

$$Purity = \frac{1}{n}\sum_{i=1}^{k} max_j^k\{C_i \cap T_j\}$$

For more on this, see Section 17.1.1 of the textbook. Print "Purity:" followed by a real number.

Deliverables

Submit two files: (1) Your python source file for the EM clustering question, named `lastname-assign3.py`, and (2) A PDF file named `lastname-assign3.pdf` that contains the output of the EM clustering. Make sure that you do not hard code the input file path in the script. Your script should read the filename from the command line as a parameter (without waiting for a prompt), and you should also read the k value (the number of clusters to find) from the command line, e.g., for iris data we will run it as `lastname-assign3.py iris.txt 3`. Submit the assignment via Canvas before due date.

Point Distribution

50 points for the clustering part, and 20 points for purity based validation part.