Portland State University

CS 446/546: Machine Learning, Winter 2020

Programming Assignment #1

A. Rhodes

Note: This assignment is **due by Tuesday, 2/18 at 600pm**; you will turn in the assignment by email to our grader, as instructed below.

Each student will turn in an individual assignment (so that we have something upon which to base your individual grade); you will write the core algorithms for this assignment yourself (e.g. k-means, GMM as discussed in class) – otherwise, you are welcome to use any basic math/graphing etc., libraries.

Data set: The data set is 2d data (for ease of visualization) simulated from 3 Gaussians, with considerable overlap. There are 500 points from each Gaussian, ordered together in the file.

Assignment #1: K-Means

Implement the standard version of the K-Means algorithm as described in lecture. The initial starting points for the K cluster means can be K randomly selected data points. You should have an option to run the algorithm *r* times from *r* different randomly chosen initializations (e.g., *r* = 10), where you then select the solution that gives the lowest sum of squares error over the *r* runs. Run the algorithm for several different values of K and report the sum of squares error for each of these models. Please include a 2-d plot of several different iterations of your algorithm with the data points and clusters.

Assignment #2: GMM

Use the following general outline to execute the EM algorithm for GMM (please refer to lecture slides as needed for details of the EM algorithm for GMMs)

- Initialize the parameters randomly

Execute E and M steps as long as the convergence condition is not satisfied:

– E-step: compute membership probabilities using the current θ current values.

– M-step: compute new parameters θ_{new} using the membership probabilities from the E-step

- After each EM iteration compute the log-likelihood of the data using θ_{new} (see lecture notes). This will allow you to print out the log-likelihood values from each EM iteration, as the algorithm is running, to monitor its convergence.

• Check for convergence by computing the value of the log-likelihood after each iteration and halting when it appears not to be changing in a significant manner from one iteration to the next. If the convergence criterion is not satisfied, then execute another EM iteration.

You should run your algorithm multiple times from *r* different randomly-chosen starting conditions (e.g. r = 10) and pick the solution that results in the highest log-likelihood (since EM in general only finds local maxima).

***Note that The EM algorithm for Gaussian mixtures is a non-trivial algorithm to get working properly: please try and debug it carefully**. Check that the likelihood is non-decreasing at each step (if the log-likelihood ever decreases you have a bug in your code).

Please print out your final parameters for the Gaussians in your GMM and check that the estimated parameters are roughly equal to the true parameters (obtained using the labels). For these data sets you could also impose a maximum number of iterations to halt the algorithm (e.g., 500) if it gets that far and still has not converged. Report your results for each and include a 2-d plot.

Report: Your report should include a short description of your experiments, along with the plots and discussion paragraphs requested above and any other relevant information to help shed light on your approach and results.

Here is what you need to turn in:

- Your report.
- Readable code.

How to turn it in (read carefully!):

- Send these items in electronic format to arhodes@pdx.edu by 6pm on the due date. No hard copy please!
- The report should be in pdf format and the code should be in plain-text format.
- Put "[CS 546] PROGRAMMING #1: your_name" in the subject line.

If there are any questions, don’t hesitate to ask me.