CS 307 Introduction to Machine Learning
Fall 2022
Assignment 5: Instance-Based Learning, Clustering, MLE, and EM

Part I: *Due by Monday, December 5, 11:59 p.m.*

Part II: *Due by Tuesday, December 13, 11:59 p.m.*

Instructions:

1. Your solution to this assignment must be submitted via gradescope.
2. For the written problems, submit your solution as a **single PDF** file to the *Assignment 5 - Part 1 (CS 307)* folder.
   - We prefer that you type your solution. If you choose to hand-write your solution, scan your PDF using a **scanner** and upload it. Make sure your final PDF is **legible**. **Regrades due to non-compliance will receive a 30% score penalty**.
   - Verify that both your answers and procedure are **correct, ordered, clean, and self-explanatory** before writing. Please ask yourself the following questions before submitting:
     - Are my answers and procedure legible?
     - Are my answers and procedure in the same order as they were presented in the assignment? Do they follow the specified notation?
     - Are there any corrections or scratched out parts that reflect negatively on my work?
     - Can my work be easily understood by someone else? Did I properly define variables or functions that I am using? Can the different steps of my development of a problem be easily identified, followed, and understood by someone else? Are there any gaps in my development of the problem that need any sort of justification (be it calculations or a written explanation)? Is it clear how I arrived to each and every result in my procedure and final answers? Could someone describe my submission as messy?
3. **IMPORTANT:** As long as you follow these guidelines, your submission should be in good shape; if not, we reserve the right to penalize answers and/or submissions as we see fit.

Part I: Written Problems (60 points)

1. **Cross-Validation (20 points)**

   Below is a dataset of instances each with attribute $x \in \Re$ and binary class value $y \in \{+, -\}$. We will use k-nearest-neighbors with Euclidean distance to predict y for a given x.

x	-1.2	-1.0	-0.2	0.1	0.2	0.4	1.6	1.7	1.9	2.0
y	$-$	$-$	$+$	$+$	$+$	$-$	$+$	$-$	$+$	$+$

   For the questions below, give your answer as the **total number of misclassifications**.

   (a) **(5 pts)** What is the leave-one-out cross-validation error of 1-nearest neighbor on this dataset?

   (b) **(5 pts)** What is the leave-one-out cross-validation error of 3-nearest neighbors on this dataset?

   (c) **(5 pts)** What is the 5-fold cross-validation error of 1-nearest neighbor on this dataset? The 5 folds should be created as follows. 1st fold: -1.2 and -1.0; 2nd fold: -0.2 and 0.1; 3rd fold: 0.2 and 0.4; 4th fold: 1.6 and 1.7; 5th fold: 1.9 and 2.0.

   (d) **(5 pts)** What is the 2-fold cross-validation error of 3-nearest neighbors on this dataset? The first fold should contain the leftmost five points, and the second fold should contain the remaining five points.

2. **Clustering I (10 points)**

   Using agglomerative clustering,

   (a) what is the minimum height of the tree required to cluster a set of n points? Give an example of a set of points that requires a tree of minimum height.

   (b) what is the maximum height of the tree required to cluster a set of n points? Give an example of a set of points that requires a tree of maximum height.

3. **Clustering II (18 points)**

   Consider the following sample of ten points in one dimension:

$$Sample = \{-1.2, -1.0, -0.2, 0.1, 0.2, 0.4, 1.6, 1.7, 1.9, 2.0\}$$

   (a) **(5 pts)** Draw the clustering tree produced by single-link clustering (i.e., $d(C_i, C_j) = \min_{x \in C_i, y \in Cj} ||x - y||$).

   (b) **(5 pts)** Draw the clustering tree produced by complete-link clustering (i.e., $d(C_i, C_j) = \max_{x \in C_i, y \in Cj} ||x - y||$).

   (c) **(8 pts)** Apply 2-means to cluster these ten points, using as seeds -1.2 and -1.0. Show the clusters produced by 2-means after each iteration until convergence.

4. **Maximum Likelihood Estimation (12 points)**

   Because of the pandemic, you have been ordering food online everyday. So far, you have ordered food from your favorite fast food shop, Burger Queen, n times, where the delivery time of your ith order was x_i minutes. Assume that the delivery time is given by the following probability distribution with parameter β. In other words, for a non-negative integer x,

   $$P(\text{delivery time} = x|\beta) = \frac{1}{\beta}e^{-\frac{x}{\beta}}$$

   Compute β such that the likelihood of observing $\{ x_1, x_2, \ldots, x_n \}$ is maximized.

Part II: Programming (40 points)

In this problem you will write a program that uses the EM algorithm to induce a soft clustering on a given set of data points in the 1-dimensional space by fitting a mixture of univariate Gaussians to the points.

Your program should take as input a *data* file containing the data points to be clustered. Sample data files are available from the assignment page of the course website. In each file, only lines containing non-space characters are relevant. Each relevant line defines a single data point.

IMPORTANT:

- To implement the EM algorithm, you should use Python (3.6.9), C++ (g++ 7.5.0), or Java (openjdk 11.0.13 2021-10-19). Do **not** use any non-standard libraries (except numpy (1.19.5) and pandas (1.1.5)) in your code.
- The number of Gaussians in the mixture and the number of data points should be determined at runtime.
- Your program should allow exactly three arguments to be specified in the **command line** invocation of your program: (1) the path to a data file, (2) the number of Guassians in the mixture, and (3) the number of EM iterations. Your program should take these three arguments in the same order as they are listed above. There should be no graphical user interface (GUI). Any program that does not conform to the above specification will receive no credit.
- Your algorithm should process the data instances in the same order as they appear in the data set.
- The model parameters should be initialized as follows. We begin by assigning each data point to exactly one cluster. Specifically we assign data point i to cluster i mod k, where k is the number of clusters. We then initialize the parameters of Gaussian k based on the points assigned to cluster k.
- You may submit as many source files as needed, but you must make sure you provide a main code entry that follows the following naming convention. Specifically, if you are using:
  - **Python**

    * Make sure that your primary source file is `main.py` and that your code runs successfully after executing `python main.py <path_to_train_file> <num_gaussians> <num_iter>`.

- **C++**
    * Make sure that your primary source file is `main.cpp` and that your code runs successfully after executing `g++ main.cpp -o a.out -std=c++17` and `./a.out <path_to_train_file><num_gaussians> <num_iter>`.

- **Java**
    * Make sure that your primary source file is `Main.java` and that your code runs successfully after executing `javac Main.java` and `java Main <path_to_train_file> <num_gaussians> <num_iter>`.

What to Do

Use EM to learn the mixture model. After each EM iteration, print to `stdout` (1) the parameters re-estimated in the M-step, including the prior, the mean, and the variance associated with each mixture component. Assuming that the number of mixture components is 4, the output for the first two iterations may look like this (do not worry about the actual values shown below; they are all made up):

```
After iteration 1:
Gaussian 1:  mean = 49.5524, variance = 1156.7818, prior = 0.2497
Gaussian 2:  mean = 52.1171, variance = 938.9100, prior = 0.2516
Gaussian 3:  mean = 39.7707, variance = 630.6709, prior = 0.2470
Gaussian 4:  mean = 52.2414, variance = 929.9241, prior = 0.2516

After iteration 2:
Gaussian 1:  mean = 50.0747, variance = 1156.7818, prior = 0.2491
Gaussian 2:  mean = 52.5949, variance = 938.9100, prior = 0.2530
Gaussian 3:  mean = 38.1149, variance = 630.6709, prior = 0.2447
Gaussian 4:  mean = 52.7329, variance = 929.9241, prior = 0.2531
```

To facilitate the grading process, please follow the format of the sample output provided in the assignment page of the course website when printing to `stdout`. In particular, when printing the means, the variances, and the priors, print exactly the four digits after the decimal place.

Submission

Once you are done, sign in to gradescope. You will be able to see *Assignment 5 - Part 2 (CS 307)* under the Assignments section. Directly submit all your source files to this submission folder. Do not create any folder and do not rename the files or upload the files in a zip file or folder (your homework will not be graded otherwise).

Grading

We will be using an output-based auto-grader for this submission, so make sure you follow the formatting from the example test files: be careful not to insert extra lines, tabs instead of spaces, etc ... When you submit, your code will be **graded using hidden test cases**, so we encourage you to test your code thoroughly. More information about the autograder will be available on Piazza shortly.