Week 8 Project

ACADEMIC HONESTY

As usual, the standard honour code and academic honesty policy applies. We will be using automated **plagiarism detection** software to ensure that only original work is given credit. Submissions isomorphic to (1) those that exist anywhere online, (2) those submitted by your classmates, or (3) those submitted by students in prior semesters, will be detected and considered plagiarism.

INSTRUCTIONS

In this assignment you will implement the K-means and EM Gaussian mixture models. We will give you n data points $\{x_1, \ldots, x_n\}$ where each $x_i \in \mathbb{R}^d$.

Recall that with K-means we are trying to find K centroids $\{\mu_1, \ldots, \mu_K\}$ and the corresponding assignments of each data point $\{c_1, \ldots, c_n\}$, where each $c_i \in \{1, \ldots, K\}$ and c_i indicates which of the K clusters the observation x_i belongs to. The objective function that we seek to minimize can be written

$$\mathcal{L} = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbf{1}(c_i = k) \|x_i - \mu_k\|^2.$$

We also discussed using the EM algorithm to learn the parameters of a Gaussian mixture model. For this model, we assume a generative process for the data as follows,

$$x_i | c_i \sim Normal(\mu_{c_i}, \Sigma_{c_i}), \quad c_i \sim Discrete(\pi).$$

In other words, the ith observation is first assigned to one of K clusters according to the probabilities in vector π, and the value of observation x_i is then generated from one of K multivariate Gaussian distributions, using the mean and covariance indexed by c_i. The EM algorithm discussed in class seeks to maximize

$p(x_1, \ldots, x_n | \pi, \mu, \Sigma) = \prod_{i=1}^{n} p(x_i | \pi, \mu, \Sigma)$

over all parameters $\pi, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K$ using the cluster assignments $c_1, \ldots, c_n$ as the hidden data.

More details about the inputs we provide and the expected outputs are given below.

WHAT YOU NEED TO SUBMIT

You can use either Python or Octave coding languages to complete this assignment. Octave is a free version of Matlab. Your Matlab code should be able to directly run in Octave, but you should not assume that advanced built-in functions will be available to you in Octave. Unfortunately we will not be supporting other languages in this course.

.

Depending on which language you use, we will execute your program using one of the following two commands.

.

Either

```
$ python hw3_clustering.py X.csv
```

Or

```
$ octave -q hw3_clustering.m X.csv
```

.

You must name your file as indicated above for your chosen language. If both files are present, we will only run your Python code. We will create and input the csv data file to your code. You should implement both the K-means and the EM-GMM algorithms either in the "hw3_clustering" file or have your "hw3_clustering" call your implementations of these algorithms located in different files.

.

The csv files that we will input into your code are formatted as follows:.

1. **X.csv:** A comma separated file containing the data. Each *row* corresponds to a single vector x_i.

WHAT YOUR PROGRAM OUTPUTS

You should write your K-means and EM-GMM codes to learn 5 clusters. Run both algorithms for 10 iterations. You can initialize your algorithms arbitrarily. We recommend that you initialize the K-means centroids by randomly selecting 5 data points. For the EM-GMM, we also recommend you initialize the mean vectors in the same way, and initialize $\boldsymbol{\pi}$ to be the uniform distribution and each $\boldsymbol{\Sigma_k}$ to be the identity matrix.

.

When executed, you will have your code write several output files each described below. It is required that you follow the formatting instructions given below. Where you see [iteration] and [cluster] below, replace these with the iteration number and the cluster number.

.

`centroids-[iteration].csv`: This is a comma separated file containing the K-means centroids for a particular iteration. The $\boldsymbol{k}$th row should contain the $\boldsymbol{k}$th centroid, and there should be 5 rows. There should be 10 total files. For example, "centroids-3.csv" will contain the centroids after the 3rd iteration.

.

`pi-[iteration].csv`: This is a comma separated file containing the cluster probabilities of the EM-GMM model. The $\boldsymbol{k}$th row should contain the $\boldsymbol{k}$th probability, $\boldsymbol{\pi_k}$, and there should be 5 rows. There should be 10 total files. For example, "pi-3.csv" will contain the cluster probabilities after the 3rd iteration.

.

`mu-[iteration].csv`: This is a comma separated file containing the means of each Gaussian of the EM-GMM model. The $\boldsymbol{k}$th row should contain the $\boldsymbol{k}$th mean , and there should be 5 rows. There should be 10 total files. For example, "mu-3.csv" will contain the means of each Gaussian after the 3rd iteration.

.

`Sigma-[cluster]-[iteration].csv`: This is a comma separated file containing the covariance matrix of one Gaussian of the EM-GMM model. If the data is $\boldsymbol{d}$-dimensional, there should be $\boldsymbol{d}$ rows with $\boldsymbol{d}$ entries in each row. There should be 50 total files. For example, "Sigma-2-3.csv" will contain the covariance matrix of the 2nd Gaussian after the 3rd iteration.

.

Note on Correctness

Please note that for both of these problems, there are multiple potential answers depending on your initialization. However, the K-means and EM-GMM algorithms have known deterministic properties that we discussed in class, and so in this sense we can distinguish between correct and incorrect answers. We strongly suggest that you test out your code on your own computer before submitting. The UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/) has a good selection of datasets for clustering.

Sample starter code to read the inputs and write the outputs: Download hw3_clustering.py

---

USE OF VOCAREUM

This assignment uses Vocareum for submission and grading. Vocareum comes equipped with an editing environment that you may use to do your development work. You are **NOT** required to use the editor. In particular, you are free to choose your favorite editor / IDE to do your development work on. When you are done with your work, you can simply upload your files onto Vocareum for submission and grading.

However, your assignments will be graded on the platform, so you **MUST** make sure that your code passes at least the submission test cases. In particular, do not use third-party libraries and packages. We do not guarantee that they will work on the platform, even if they work on your personal computer. For the purposes of this project, everything that comes with the standard Python or Matlab libraries should be more than sufficient.

To check the formatting of your submission, select to have your code submitted, but not graded. We will output the results of the formatting check to SubmissionReport.txt. We can guarantee a very low grade if you do not pass this submission test. Once your outputs satisfy the formatting requirements and you are confident in your code, select to have it graded.

- **You will have unlimited opportunities to submit your code for grading**
- **You can test your code in the terminal without submitting it**
- **You will get graded once you click "Submit"**

WORK ON PROJECT (ML.T) (External resource) (25.0 / 25.0 points)

Your email address will be used to identify your submission entry.

Launch Project