Boston University
Department of Electrical and Computer Engineering

ENG EC 414 Introduction to Machine Learning

HW 8

Issued: Fri 30 Oct 2020 **Due:** 10:00am Fri 6 Nov 2020 in Gradescope (non-code) + Blackboard

Important: Before you proceed, please read the documents pertaining to *Homework formatting and submission guidelines* in the Homeworks section of Blackboard. **In particular, for computer assignments you are prohibited from using any online code or built-in MATLAB functions except as indicated in the problem or skeleton code (when provided).**

Important: To obtain full grade, please clearly motivate all your answers.

Note: Problem difficulty = number of coffee cups ☕

Problem 8.1 [36pts] *(k-means implementation)* In this problem we will implement k-means clustering and explore the impact of initialization and number of clusters on one synthetic and one real-world dataset. We will also explore a dataset where k-means will fail to produce meaningful clusters. You are provided skeleton code to assist you in implementing this clustering method.

(a) [8 points] *(Implement k-means)* Implement the k-means algorithm we saw in class in Matlab with prototype

```
[c,obj,y] = k_means_clustering(X, c0, T)
```

where `c` are the returned centers $\in \mathbb{R}^{k\times d}$, `obj` is the final value of the objective function, `y` are the inferred "labels", `X` is the matrix of the training samples $\in \mathbb{R}^{m\times d}$, `c0` are the initial centers $\in \mathbb{R}^{k\times d}$, `T` is the maximum number of iterations. Note that there is no need to pass k because the algorithm can infer it from the size of `c0`.

(b) [4pts] *(Synthetic training set generation)* Generate 3 two-dimensional Gaussian clusters of data points having the following mean vectors and covariance matrices: $\boldsymbol{\mu}_1 = [2,2]^\top$, $\boldsymbol{\mu}_2 = [-2,2]^\top$, $\boldsymbol{\mu}_3 = [0,-3.25]^\top$, and $\Sigma_1 = 0.02\cdot I_2$, $\Sigma_2 = 0.05\cdot I_2$, $\Sigma_3 = 0.07\cdot I_2$, where I_2 is the 2×2 identity matrix. You can use `mvnrnd` to generate multivariate Gaussian noise. Let each data cluster have 50 points. Plot the generated Gaussian data. Color the data points in the 1st, 2nd, and 3rd clusters with red, green, and blue colors, respectively. You can use `gscatter` to easily plot the points in different colors. Use your implementation of k-means on this dataset with $k = 3$ and the following **initialization**: $\boldsymbol{c}_1^{\text{initial}} = [3,3]^\top$, $\boldsymbol{c}_2^{\text{initial}} = [-4,-1]^\top$, $\boldsymbol{c}_3^{\text{initial}} = [2,-4]^\top$. and the maximum number of iterations equal to 10. Plot the clusters produced by your k-means algorithm, plotting the points of each cluster with a different color.

(c) [4pts] *(Effect of different initialization)* Using the same synthetic training dataset from part (b), re-run your k-means algorithm implementation for $k = 3$ using the following (different) **initialization:** $\boldsymbol{c}_1^{\text{initial}} = [-.14, 2.61]^\top$, $\boldsymbol{c}_2^{\text{initial}} = [3.15, -0.84]^\top$, $\boldsymbol{c}_3^{\text{initial}} = [-3.28, -1.58]^\top$. Create a new plot of the resulting clusters. Discuss what you observe.

(d) [10pts] *(Best of multiple random initializations)* To reduce the possibility selecting an initialization which results in a "bad" clustering, the k-means algorithm is typically run multiple times using different random initializations. The best clustering result, i.e., the one having the smallest objective function is saved and used as the final output. Run your implementation of the k-means algorithm on the same synthetic training dataset from part (b) for 10 different random initializations. Report the objective function for each of the 10 trials. Identify the trial which yields the smallest objective function value. Report its objective function value and create a plot of the clustering produced by it.

(e) [6pts] *(Clustering a real-world dataset)* Here we examine a real-world digits, MNIST. The inputs are 28 x 28 images of digits flattened to vectors of size 784. Ignore the labels and apply your implementation of the k-means algorithm over the training data with $k = 10$ selecting the best of 10 different random initializations as the final output. Reshape the centers as images and plot the images corresponding to the 10 centers. Hint: you can reshape and display the first training sample with the following command `imagesc(reshape(Xtr(1,:),28,28)')`. Also, if your implementation of k-means is too slow, take a subset of the training set.

(f) [4pts] *(Failure of k-means)* Here we examine the performance of the k-means algorithm on a dataset composed of 3 concentric rings. Use `sample_circle.m` to generate a dataset with 3 concentric ring clusters and 500 points for each cluster. Plot the dataset. Apply your implementation of the k-means algorithm on this dataset using $k = 3$ and choosing the best of 10 different random initializations. Create a plot of the best clustered results. Discuss what you observe.

Code-submission via Blackboard: You must prepare 1 files: `k_means_clustering.m` for Problem 8.1(a). Place them in a **single** directory which should be zipped and uploaded into Blackboard. Your directory must be named as follows: <yourBUemailID>_hwX where X is the homework number. For example, if your BU email address is `charles500@bu.edu` then for homework number 8 you would submit a single directory named: `charles500_hw8.zip` which contains all the MATLAB code (and only the code).

Three corresponding skeleton code files are provided for your reference. Reach-out to the TAs via Piazza and their office/discussion hours for questions related to coding.