ELEC 425 - Fall 2018
Machine Learning and Deep Learning
Assignment II

Due: 9:00 p.m., Monday November 5^{th}, 2018

Important Notes

Please kindly understand this assignment is an **"individual"** assignment. You are allowed to discuss the general idea of the assignment but your discussion may not include the specific answers to any of the problems. Please kindly understand that any form of plagiarism may lead to a mark of 0 for your assignment.

Submission Guide

- Submit a **zip** file to "Assignment 2" Dropbox folder on onQ, which should contain:
  - A **pdf** report that includes your answers to the questions.
  - **All your source code.** For example, if you use Matlab, you should include all your .m files.
  - **A README file** telling us what each program/code file is for and how to run your code.
- **Late submission:** If you are late, for every 2 hours, you will lose **15%** of your assignment marks.

K-median and K-medoids

1 Implement K-medians (3 marks)

Although K-means is one of the most popular algorithms for clustering, it is very sensitive to local structures of data and the choices of initial centres can sometime lead to very different clusters.

It's well known that a median is less sensitive to extreme values. The difference between K-means and K-median is that we use Manhattan distance (also called cityblock distance in latest Matlab version) as our distance metric, and we use a cluster's median but not mean as its centre.

Based on Lab 2 solution (published on onQ already), implement the K-medians algorithm. Run your code to cluster the Lab 2 data with 2, 3, and 4 clusters separately. For each case, try different initialization and in your report **show the pictures of different clusters you find**, since different initialization may lead to different clusters. **If you use Matlab, name your K-medians code as k_medians.m and the main program (the code calling k_medians.m) as k_medians_main.m**. Submit both files. If you use python or other programming languages, submit your source code and write your README file clearly so our TA can run your code.

2 Prove the EM Updating Algorithm Used in K-medians (3 marks)

As discussed in class, the K-means algorithm minimizes the following loss function ($r_{nk} = 1$ iff a data point $\mathbf{x}_n$ belongs to cluster k, and 0 otherwise).

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \mu_k||^2 \tag{1}$$

Similarly, please prove that the K-medians algorithm you implemented above actually minimizes the following error function:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} |\mathbf{x}_n - \mu_k| \tag{2}$$

(Hints: Similar to K-means, you can separate the proof into 2 steps. In addition to our K-means slides, you may refer to our lecture slides on constant regression models, to obtain the derivative with regard to μ_k.)

3 K-medoids (4 marks)

As discussed in our class, the K-medoids algorithm always selects one of the data points in a cluster as the centre (instead of using a mean or median). Once data points have been assigned to clusters in a E-step, you need to find one data point that minimizes the sum of points-to-centre distances within a cluster in the M-step. Use Manhattan/cityblock distance to implement the K-medoids algorithm. Run your code to cluster the Lab 2 data with 2, 3, and 4 clusters separately. For each case, try different initialization and in your report **show the pictures of different clusters you find**. **If you use Matlab, name your K-medoids code as k_medoids.m and the main program (the code calling k_medoids.m) as the k_medoids_main.m**. Submit both files. If you use python or other programming languages, submit your source code and write your README file clearly so our TA can run your code.