CS534 — Implementation Assignment 3 — Due May 30th, 2012

General instruction.

1. You are recommended to use Matlab for your implementation. However, other languages (such as Java, C/C++ and R) are also accepted.
2. You can work solo or in team of 2 people. Each team will only need to submit one copy of their solution.
3. Your source code and report will be submitted through the TEACH site (Due at midnight on the due date).

`https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth`

4. You will be graded based on your code as well as the report. **The clarity and quality of the report will be worth 10 pts**. So please write your report in clear and concise manner. Clearly label your figures, legends, and tables if any. Please also provide appropriate comments and notes for your source code.
5. There are three datasets for this assignment. They are all simple $2-d$ data. The files provided are comma separated, with each row corresponding to one data point.

Kmeans clustering (total 70 points)

In this assignment you will implement Kmeans clustering and experiment with your implementation for the following tasks. In your report should contain clear answers to the following questions and should be organized in a clean and easy-to-read fashion.

1. Initialization (10 pts) Run your Kmeans algorithm with $k = 5$ on the provided **data1** with randomly initialized cluster centers for 200 times. Each time, you will randomly pick five points in the data to serve as the initial centers and run your kmeans algorithm to convergence. You will repeat this for 200 times and record all the centers that are found.

a. Create a single figure in which the original data points are plotted in one color and the found cluster centers in the 200 runs in a different color. **Make sure that the cluster centers are clearly visible from the other points when printed in black and white.**

b. Report the minimum, maximum, mean and standard deviation of the within-cluster sum of squared distances for the clustering found by your algoirthm in the 200 random runs. Please comment on your results regard the sensitivity of the kmeans algorithm to the initialization.

2. Finding K (10 pts) Identifying how many clusters are in the data is a very challenging task. In this task, you will try out a commonly used heuristic on the provided data2. We will consider a variety of different k values ($k = 2, 3, ..., 15$). For each k value, you will run your kmeans algorithm with 10 different random initializations and record the lowest within-cluster sum of squared distances obtained for that k value. You will then plot them as a function of k.

a. What trend do you observe as the k increase from 2 to 15? Do you expect this trend to be generally true for all data as we increase k?

b. A commonly used heuristic is to look for the "knee" in this curve, which is the value of k where the rate decreasing sharply reduces. Find the knees in your plot and provide an explanation to your finding. Why do we see multiple knees? what do they correspond to?

3. Feature normalization (10 pts) Kmeans clustering and many other clustering methods are distance based. Distances are sensitive to feature processing such as normalization. For the given data3, please do the following.

a. Apply your implemented kmeans algorithm with $k = 2$ for 200 different random initializations. Record all the cluster centers that are found in these 200 runs. Plot the original data in one color and plot the cluster centers in the same figure using a different color(and shape if that helps to differentiate them).

b. Now normalize the features of the given data set. That is, first center the data to have zero mean(by subtracting the mean from the data), then rescale each feature dimension to have unit variance. Redo part a with this normalized dataset. Report the difference that you observe.

The results should be different with and without normalization. Note that this should not be taken to mean that data always need to be normalized. In some cases, the difference in scale can be truly meaningful and normalization could remove such useful information. In other cases, it is crucial to properly scale the data. The decision should be made in a case-by-case fashion.