HW3

2016312107 문경진

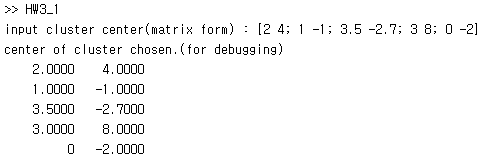
**1. Implement the K-means clustering algorithm using MATLAB. Your function should have the following function prototype.**

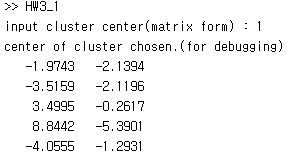
**Function prototype:  
- Input: k, initial centers, X  
- Output: cluster label**

Here is the procedure of HW3\_1.m.

1. Load x.mat.
2. set k value, dimension(which will be the size of centroid matrix), and centroid. We can set centroid by typing centroid matrix (it should be 5\*2 matrix). If the size of matrix is not 5\*2, are randomly set.

Ex)

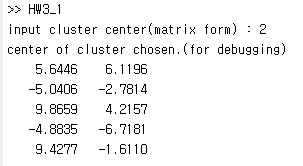


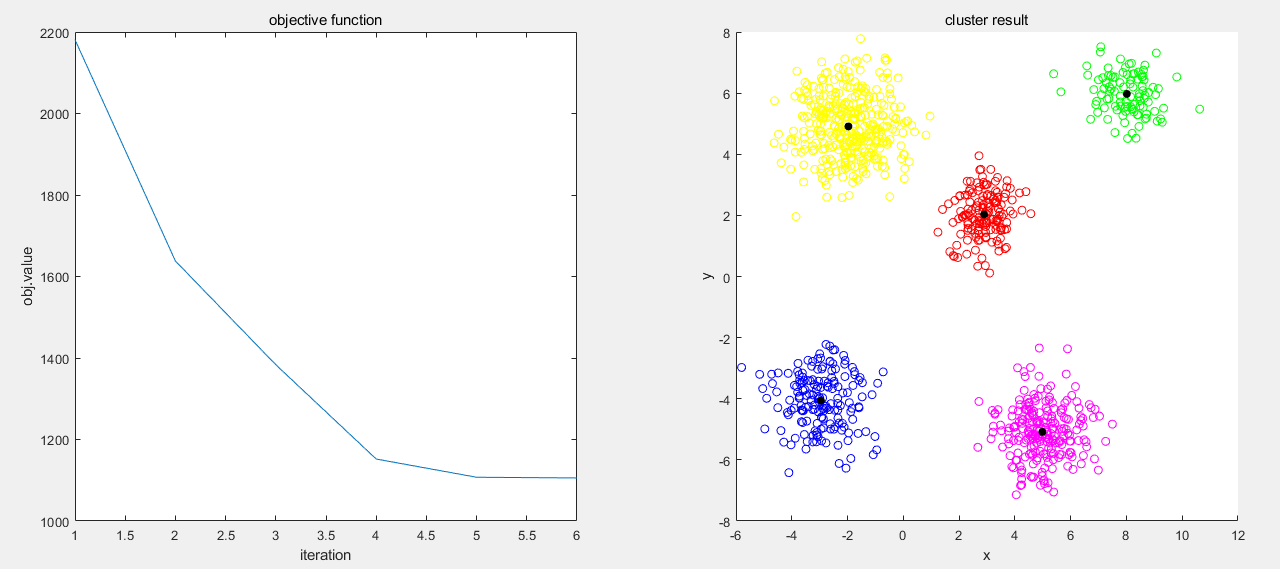


1. Initially put data into cluster by putDataIntoCluster(). This function calculates distance between each of X vectors and centroid vector, and put data into cluster which number is argmin() of distance vector. The clustered data is stored in clusteredData, which has [clusterNum distance] for all X data. After that, set beforeLabel the first column of clutseredData.
2. in the loop, re-calculate centroid based on clusteredData. It is calculated by setNewCentroid(centroid, clusteredData, X), which returns mean of data in the each clutser.
3. Also in the loop, re-allocate cluster. and then get clusteredData and compare two results, beforeLabel and afterLabel. Objective value is also calculated in this time. If two are not same, repeat. If not, iteration is terminated. It means, the procedure is over.
4. Show graphs of objective function and data.

**(1) Visualize the clustering results on the given data matrix X.mat. Set k=5, and use a different color for each cluster. Also visualize the center of each cluster in the plot.**

**(2) Plot the K-means objective function value vs. the number of iterations.**





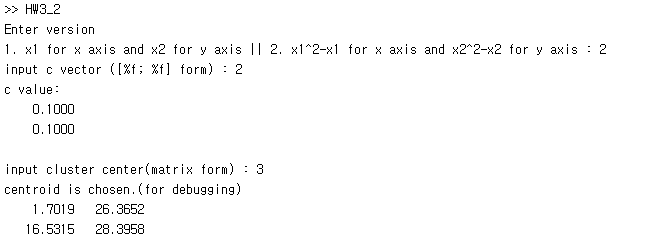
**2. Implement the kernel K-means clustering algorithm using MATLAB. Set k=2 and use the Gaussian RBF kernel. Visualize the clustering results on the given data matrix Y.mat. Use a different color for each cluster. Describe what value you used for the parameter c for the Gaussian RBF kernel. Your function should have the following function prototype.**

**Function prototype:  
- Input: k, initial centers, c, Y  
- Output: cluster label**

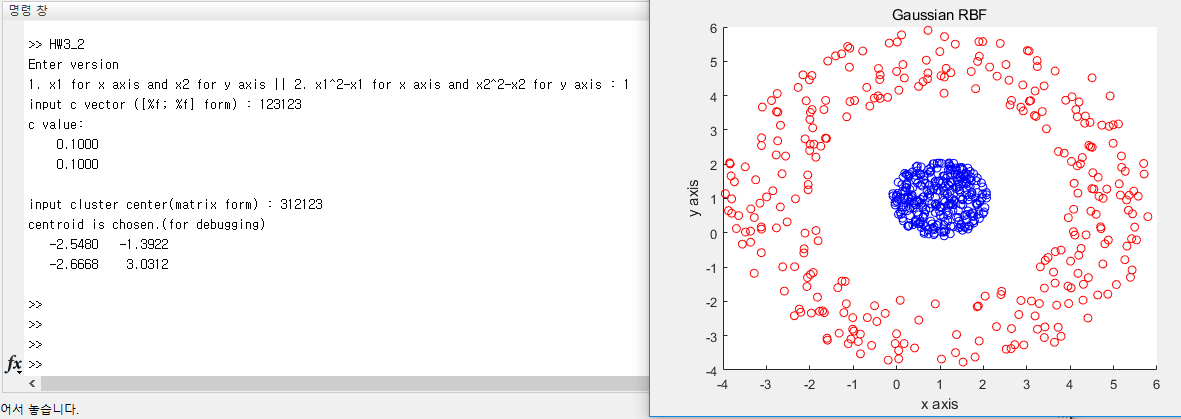
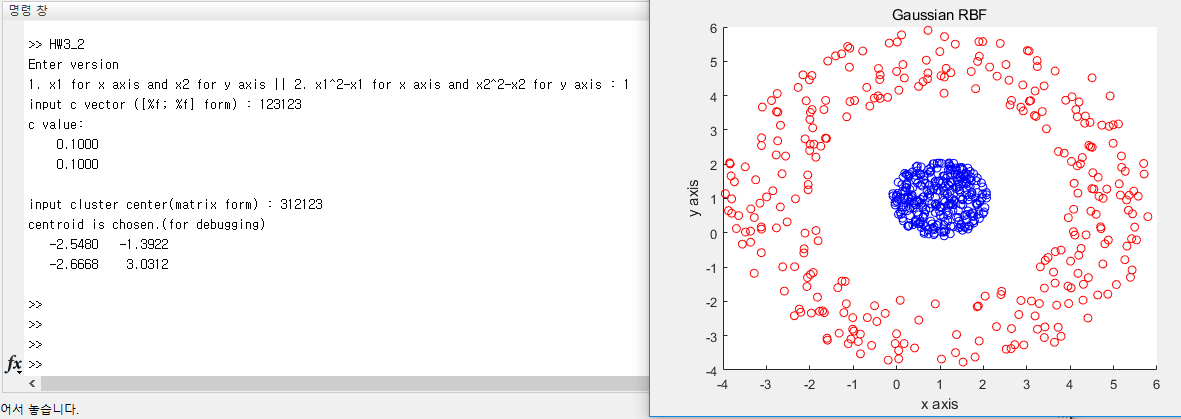
Here is the procedure of HW3\_2.m

1. Load y.mat
2. Input version. 1 for question 1, 2 for question 2.
3. Input c value (2\*1 vector form). If input vector size is not equal to [2 1], initial c vector is set [0.1; 0.1]. (ex: available input : [0.25; 0.4] (O), 1 (X), [0.5 0.5] (X))
4. Input cluster centroid. If input vector size is not equal to [2 2], centroid is set randomly.

Ex)



1. Randomly set initial label, and get kernel matrix. Kernel matrix is calculated in getKernelMatrix().
2. Until objective value is not changed, repeat to set label. First count the amount of data in each cluster, and calculate new label. And then calculate objective function by kernelObjFunc().
3. Plot result.
4. **Draw a plot which has x1 for the x axis and x2 for the y axis.**



**(2) Draw a plot which has x12-x1 for the x axis and x22-x2 for the y axis.**

