Assignment 2: KMeans and PCA

Due: March 22 2024 11:59PM

Implement KMeans

Two datasets (Iris, YeastGene) are provided. In each dataset, each row corresponds to an object and each column corresponds to an attribute. The attribute values are comma-separated. You don't need to do normalization for any of the datasets.

The initial centroids to be used in K-means for Iris and YeastGene datasets are provided. In each file, each row denotes the initial centroid of a cluster.

In this part, you are asked to implement K-means algorithm. The provided template (*kmeans_template.py*) is for Python 3.

In *kmeans_template.py*, you are asked to fill in two functions: *assignCluster* and *getCentroid*. In *assignCluster*, each object is assigned to the cluster whose centroid is the closest to the object, based on Euclidean distance. In *getCentroid,* cluster centroids are updated based on current assignment.

If you are not sure about whether it is OK to use a certain function, please post your question on Piazza.

Please take the following steps:

1. Implement K-means algorithm as follows:
   Repeat T times:
   a. For each object x_i
      i. Calculate Euclidean distance between x_i and each of the K centroids
      ii. Assign x_i to the cluster whose centroid is the closest to x_i
   b. For each cluster
      i. Calculate its centroid as the mean of all the objects in that cluster

2. Test your K-means on Iris dataset. Use the provided initial centroids and set the number of iterations (T) as 12, and the number of clusters (K) as 3. The final cluster centroids should be:

   Cluster 1: 5.006,3.418,1.464,0.244
   Cluster 2: 6.8538,3.0769,5.7154,2.0538
   Cluster 3: 5.8836,2.741,4.3885,1.4344

3. After running the filled *kmeans_template.py* code file, you can obtain a data file *Iris_with_cluster.csv*, which contains the Iris data and the assigned clusters.

Implement PCA

Next, you should implement the PCA to project the clustered Iris data to 2 dimensions and plotting function to draw the scatter plot. You will use **different colors for different clusters from KMeans** in the scatter plot.

To do so, please take the following steps:

1. The template based on Python is provided. You are asked to complete the required functions (pca and plot) in that template. Please do not change the input and output provided in the template. The details of the template are explained as follows: There are four functions in the template: ***loadDataSet, pca, plot*** and ***the main function***.

2. The loadDataSet function is to load the dataset from the csv file. The input of this function is the filename of the dataset and the outputs are the data matrix (*dataMat*) and corresponding cluster labels (*labelMat*). Each row in the *dataMat* represents an observation and each column in the *dataMat* represents an attribute. Each entry of *labelMat* is the cluster label corresponding to each row of *dataMat*.

3. ***You need to implement PCA algorithm in the pca function.***
    a. The input of the pca function is *dataMat* obtained from the *loadDataSet* function and the number of dimensions after PCA transformation which is set to be 2.
    b. The output of the pca function is the two-dimensional data(*lowDDataMat*) after PCA transformation.

4. ***In the plot function you need to plot all observations as scatter plots and color the data points according to their cluster labels. You also need to save the figure.*** The input to the plot function is the data matrix after PCA transformation (*lowDDataMat*) obtained from the pca function, the label vector (*labelMat*) obtained from the loadDataSet function and the name of the saved figure (figname).

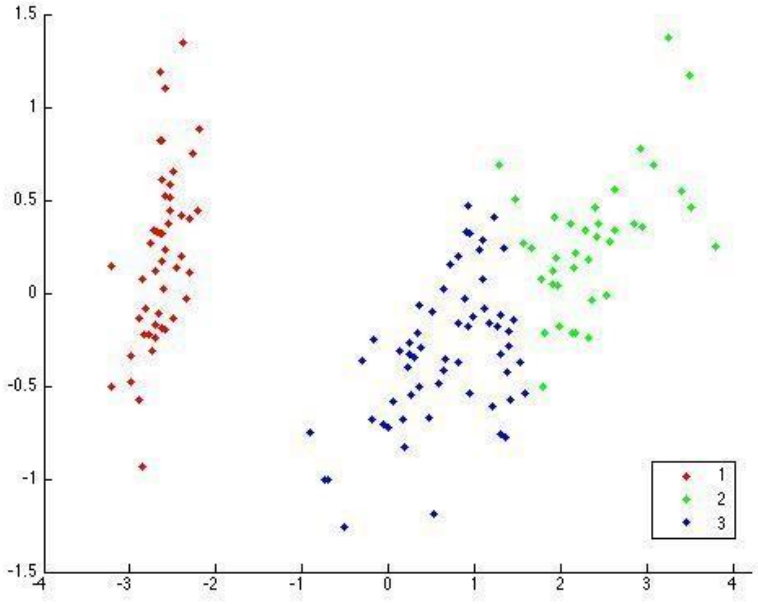
5. These functions are called in the main function. To run the template, you can use the command line and then type the following command:

Python pca_template.py [filename]

You can also run the template in an IDE and specify the configuration.

The parameter filename is an optional parameter to specify the name of the data file you want to read. This file should be the one you saved from KMeans. If it is not specified, the default value (*'Iris_with_cluster.csv'*) will be used.

The final scatter plot should look like the one below:



If you get the correct final cluster centroids and the plot, then repeat above steps on the YeastGene dataset. Use the provided initial centroids and set the number of iterations (*T*) as 7, and the number of clusters (*K*) as 6.

Prepare Submission

Your final submission should be a zip file named as Assignment2.zip. In the zip file, you should include:

1. The Python codes.

2. Report: A WORD or PDF file named as Assignment2.docx or Assignment2.pdf. The report should consist of the following parts:

    a. The cluster centroids obtained on YeastGene dataset after the *T=7* iterations.

    b. The scatter plot obtained on YeastGene dataset after applying PCA and plotting points using different colors for different clusters.

c. The codes of your K-means clustering algorithm implementations (i.e., the two functions: *assignCluster* and *getCentroid*) and PCA and visualization implementations (i.e., the two functions: *pca* and *plot*)

3. Submit the zip file under Assignment 2 on Brightspace.

Please refer to Course Syllabus for late submission policy and academic integrity policy. This assignment must be done independently. Running your submitted code should be able to reproduce the results in the report.