Machine Learning Assignment 1 Report

Kasturi Chandra Shekhar

UTA-ID:1001825454

The University of Texas at Arlington

Computer and Information Sciences

CSE 6363 Machine Learning
Prof.Won Hwa Kim

0.1 Problem 1

0.1.1 Problem 1.1

K-means Clustering is a vector quantization algorithm to cluster the given data in to k clusters depending on the Euclidean distances between the samples where we use cluster centers a.k.a centroids to model the data. Given a set of data $(x1, x2, x3...)$ where each observation is an d-dimensional vector partitioning the data into k cluster sets $clusters = (c1, c2, c3...)$

The first part of implementing the K-means Clustering algorithm involves defining a function named $cluster = mykmeans(X, k)$ where X = 2D Gaussian random samples for different clusters and k = Number of Clusters returning the clustered data which is computed using $l2 - norm$.Euclidean distances() to compute the distances between the sampled data.

$$cluster = mykmeans(X, k)$$

0.1.2 Problem 1.2

The 2D-Gaussian random sample are generated using the mean and sigma values as given below:

$$\mu_1 = \begin{bmatrix} -3 & 0 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} 3 & 0 \end{bmatrix}$$

$$\mu_3 = \begin{bmatrix} 0 & -3 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & 0.75 \\ 0.75 & 1 \end{bmatrix}$$

We use multi-variatenormal to generate the Gaussian random samples with $N = 300$ for each cluster and then function $mykmeans(X, k)$ is called.The initial dataset with randomly generated centroids and the 900 samples is plotted in a scatter plot as shown below:

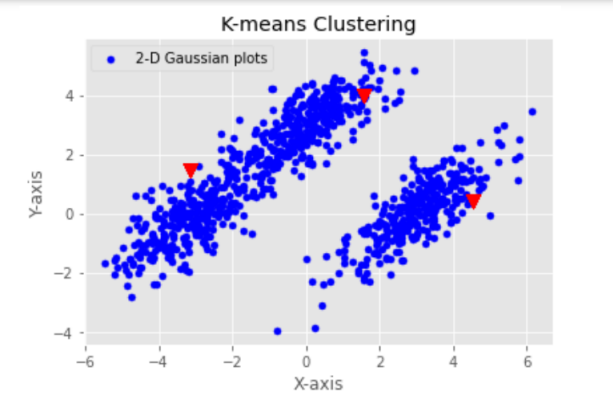

Figure 1: Initial dataset with K=3.

The data is clustered by calculating the $Euclidean, distance(x, y)$ between the samples and each centroid.The sample is assigned to the cluster of the nearest centroid and the new centroids are obtained.For k=3 the clustered data is visualised as below:

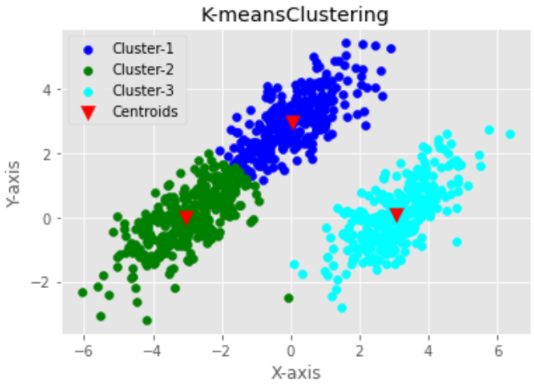

Figure 2: Clustered dataset with K=3.

Accuracy: The accuracy for the k-means algorithm is calculated by comparing the Cluster assignments for the initial untrained data with the trained data. The below diagram depicts the accuracy obtained when clustered using $mykmeans(X, k)$ where k=3

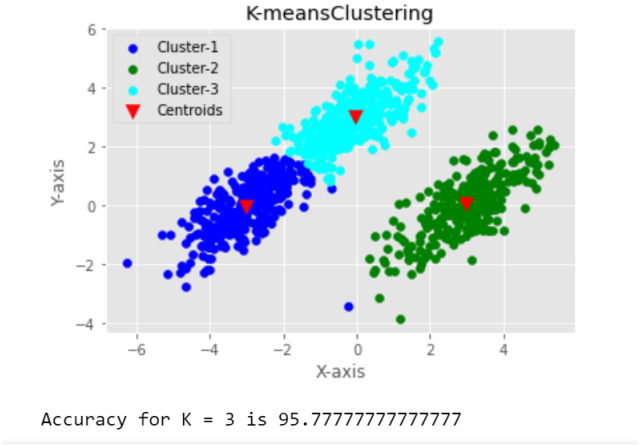

Figure 3: Accuracy of Clustered dataset with K=3.

When Clustered the data with k=3 the accuracy value is above 90 percent because initially we plotted the dataset using different three values of mean and variance because the initial cluster $values = 1, 2, 3$ will have the highest probability to be fitted right with value k=3 also generating 3 clusters with $values = 1, 2, 3$. The accuracy for other k values are less when compared to k=3 is because it partitions the data into varaiate number of clusters when compared to the generated dataset.

The cluster centers are accurate because the centroids are defined where the distance between the samplesand the centroids in that cluster is mimimum or zero.

While executing the algorithm the accuracies for k values other than 3 are less when compared to the other k values as the initial dataset is generated with three values of means and variance.

0.1.3 Problem 1.3

When changing the parameters of the mean and sigma the 2D ranodm Gaussian samples generated are much more scattered when compared to the dataset generated with the iniital parameters

$$\mu_1 = \begin{bmatrix} -2 & 0 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} 2 & 0 \end{bmatrix}$$

$$\mu_3 = \begin{bmatrix} 0 & 2 \end{bmatrix}$$

The implementation of k-means with this dataset is harder when compared to the previous dataset beacuse there plots in the dataset are more scattered which results in difficulty for selecting the centroids as there are many points which are similar to becoming the centroids. The visualisation for k=3 is as shown below:

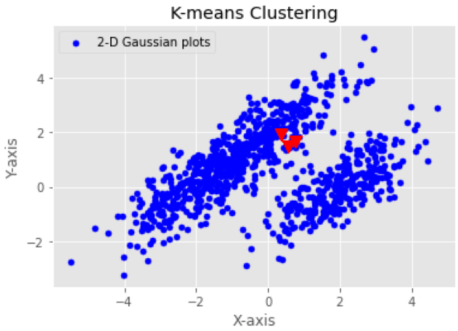

Figure 4: Initial dataset with K=3.

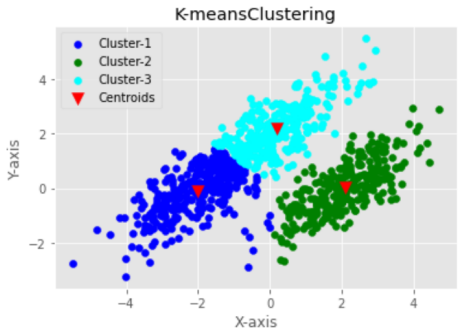

Figure 5: Clustered dataset with K=3.

Accuracy when k=3 is obtained around 90 percent with only difficulty for the centroids updation which takes some more time as there are multiple near-centroid becoming points.

Visualising for $k = 2, 4, 5$

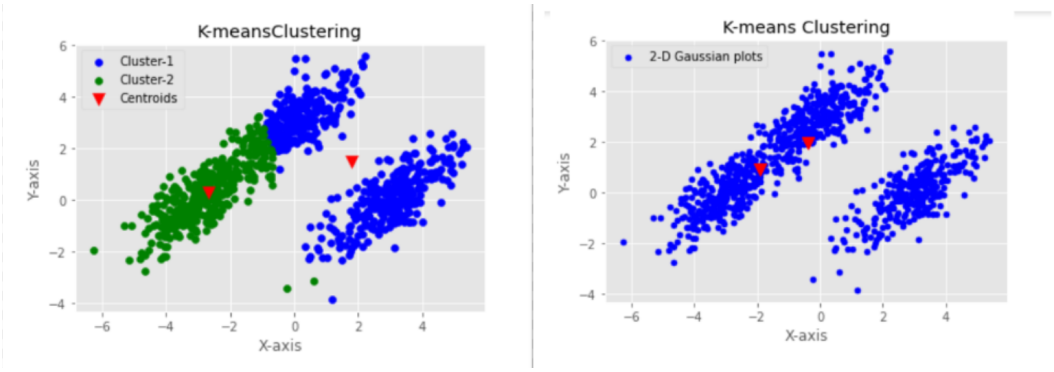

Figure 6: when k=2

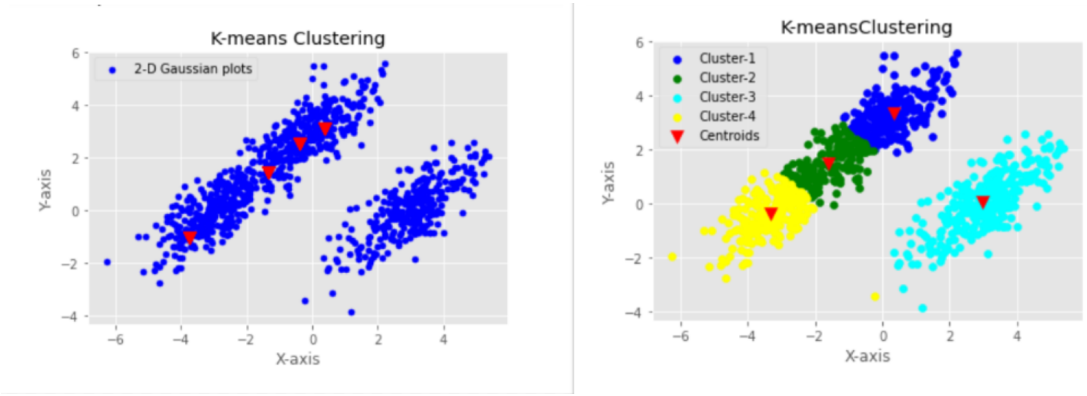

Figure 7: when k=4

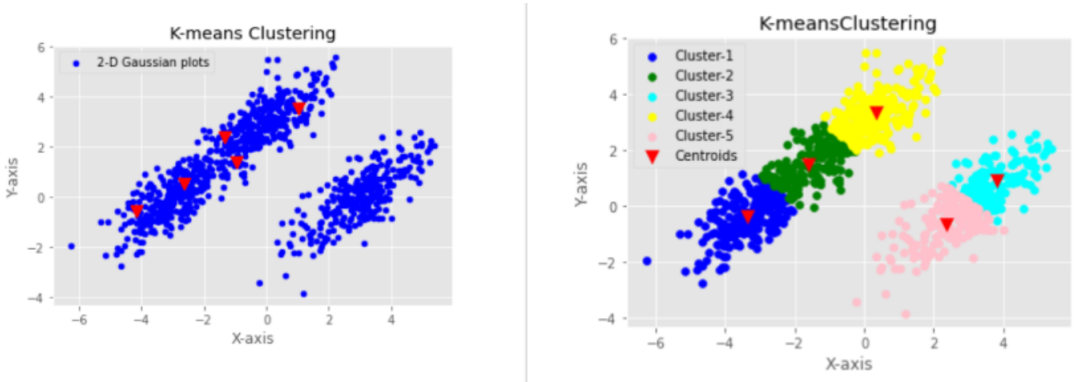

Figure 8: when k=5

0.2 Problem 2

K-Nearest neighbors is an instance-based learning or lazy-learning algorithm used for classification as well as regression. Given a set of data $(x1, x2, x3...)$ where each observation is an d-dimensional vector in which it determines the k-closest samples in the feature space.The output is the determined whether the algorithm is used for either classification or regression problem.In the Problem 2.1 the k-nearest neighbors is used as classification problem, in the Problem 2.2 it is used as a regression problem. In both the methods the $l2 - norm$ or the Euclidean distances are used to calculate the most nearest neighbors to the sample space.

0.2.1 Problem 2.1 KNN-Classifier

In this problem the dataset is generated the same way as we did in the first problem with the help of mean and variances.The training data consists of 2D gaussian random samples as the first two columns and the third column is assigned random class labels with $values = 0, 1$.

$$\mu = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & 0.75 \\ 0.75 & 1 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 1 & -0.5 \\ 0.5 & 1 \end{bmatrix}$$

The 2D gaussian random sample data with N=200 for training data and N=50 for testing data with appended randomly assigned class labels is then passed to the $myknnclassify(train, test, k)$ to predict the class labels as 0,1 by computing the Euclidean distance with all the other samples in the dataset and determining the k-nearest or closest datapoints as neighbors. The k-nearest neighbors classification accuracy is calculated by matching the actual class labels with predicted class labels.The outputs for different values $k = 1, 2, 3, 4, 5, 10, 20$ is obtained as follows when called.It is observed that the accuracy value fork=3 is highest when compared to all other values of k suggesting the optimal number of classes required to classify the given dataset.

$$myknnclassify(train, test, k)$$

```
Accuracy for k =  1 47.0
Accuracy for k =  2 56.0
Accuracy for k =  3 57.0
Accuracy for k =  4 49.0
Accuracy for k =  5 48.0
Accuracy for k =  10 46.0
Accuracy for k =  20 48.0
```

Figure 9: Accuracies for myknnclassify(train,test,k)

0.2.2 Problem 2.2 KNN-Regressor

In this problem the K-Nearest Neighbor algorithm is used as a regressor function which uses the same dataset to obtain the k-nearest neighbors and then calculate the average of those class labels of k-nearest neighbors to predict the class label of the training dataset.The training dataset consists of 2D gaussian data samples as the first two columns and target value as the third column with parameters specified below:

$$\mu_0 = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\Sigma_0 = \begin{bmatrix} 1 & 0.75 \\ 0.75 & 1 \end{bmatrix}$$

$$y = 2x_1 + x_2 + \epsilon$$

The training data consists of N=300 for training data and N=100 for testing data and Gaussian noise with sigma=0.5 The regression algorithm calculates the k-nearest neighbors using the $l2 - norm$ or Euclidean distance between each sample to find the k-nearest neighbors and averages the target labels of those k-nearest neighbors to generate a regression value. The regressor unlike the classifier does not try to classify the data but regresses the data to fit a straight line depending on the k-nearest neighbors.The function

$$myknnregress(X, test, k)$$

is called to find the K-nearest neighbors and return the regressed value for the training data with specified value of k. The funcation returns the accuracies for different values of $k = 1, 2, 3, 4, 5, 20, 50, 100$.

```
Accuracy for k: 1
81.45041840988067
Accuracy for k: 2
82.10818520577797
Accuracy for k: 3
84.55020153974931
Accuracy for k: 4
84.28863129095129
Accuracy for k: 5
87.04368931352546
Accuracy for k: 10
82.69858983250626
Accuracy for k: 20
84.00536889633797
Accuracy for k: 50
60.12415066150986
Accuracy for k: 100
34.355312173888606
```

Figure 10: Accuracies for myknnregress(X,test,k)

It is observed that the accuracy value for k=5 is the highest with 87.0 percent which indicates that when tried to plot a straight line the number of nearest neighbors to be taken into consideration to predict the regressed value in order for an accurate fit is k=5.Accuracy values keep on decreasing with increasing value of k after k=5

0.2.3 Problem 2.3 KNN- Locally weighted Regressor

Locally weighted regression (LWR) is a memory-based non-parametric method that performs a regression around a point of interest using only training data that are "local" to that point.The K-nearest neighbor using locally weighted regression is used when the plots in the dataset are in non-linear fashion making it difficult to plot a straight line as an accurate fit to the dataset. Locally weighted regressor uses weights to be appended to training data where higher order weights are appended to the closer datapoints and lower-order weights to the points which are far away.We used a multivariate smoothing procedure which is an extension of uni-variate locally weighted regression.The amount of smoothing can be defined by using a kernel smoothing function which smoothens the curves to be the accurate fit by finind accurate weights to be appended to the training data.

$$kernelsmoothing = w = \frac{ae^{|X-X_0|}}{2c}$$

After obtaining the values of w using the least squares method we then find predicted values using the $Y = XW$

$$(X^T X)^{-1} X^T Y = (X^T X)^{-1} (X^T X) W^*$$

The same sample data in Problem 2.2 is generated to perform the regression instead of the target values the weights are being appened to the predicted values to obtain a perfect smoothed curve for the dataset using the similar $l2 - norm$ Euclidean distance to find the k-nearest neighbours.The dataset is then passed to the function

$$[value\ weight] = myLWR(X, test, k)$$

with $k = 1, 2, 3, 5, 10, 20, 50, 100$.The accuracies are measured for the untrained test data and the trained test data set.It can be observed that the accuracy values for K=3 is the highest when compared to other values of k determining the optimal number of neighbors to consider before regressing the value with weights.The accuracy values keep on decreasing for the larger values of k as it becomes really extensive to consider 100 nearest neighbors to regress the weighted target value.

```
[1, 2, 3, 4, 5, 10, 20, 50, 100]
Accuracy for k value: 1
100.0
Accuracy for k value: 2
76.02005864602863
Accuracy for k value: 3
84.55610256444994
Accuracy for k value: 4
82.19073382989843
Accuracy for k value: 5
77.55434737410121
Accuracy for k value: 10
68.16406358238447
Accuracy for k value: 20
64.97433210941965
Accuracy for k value: 50
48.18081120403139
Accuracy for k value: 100
26.17180184356638
```

Figure 11: Accuracies for myLWR(X,test,k)