Programming Assignment 4: Clustering

James Santiago

05 November 2010

IST 562 Prof: James Wang

Contents

Clustering Problem 3

Assignment Steps 3

Step 1: Define a Cluster Class and its Members 3

Step 2: Implement a K-means Clustering Algorithm Function 3

Step 3: Data Summarization 4

Step 4: Letter Recognition 4

How to Run the Program 5

Example 1: 7

Example 2: 7

Example 3: 8

References ..... **Error! Bookmark not defined.**

Appendix A: Source Code ..... see PA3 Source.pdf

Clustering Problem

The clustering problem faced is character recognition. The dataset provided by the UC Irvine Machine Learning Repository provides data on the size and shape of 26 English letters from 20 different fonts. The entire dataset contains 20,000 data points each with 16 attributes. The goal of this clustering program is to group the entire dataset into 26 clusters which will represent each of the 26 English characters. With these clusters we would then be able to determine future data point assignment and essentially be able to determine which letter a set of attribute values represent. The k-means clustering algorithm will be implemented to accomplish this goal. The programming language of choice will be C#.

Assignment Steps

The assignment was broken down into the major steps needed to perform the k-means clustering algorithm on the dataset.

Step 1: Define a Cluster Class and its Members

This was not the first step performed in practice but it was later determined that this was essential to the program and should have been performed first. The cluster object implemented the CollectionBase class withing the .NET architecture. This allowed the use of a list wrapper to hold the datapoints for each cluster and also allowed the definition of a few sub-members to define information specific to clusters. Each cluster was essentially an array of datapoints with the added information of cluster mean (centroid value), mean of each attribute and sum of each attribute.

Additionally a second class was made to be a collection of clusters.

Step 2: Implement a K-means Clustering Algorithm Function

This step was broken into two sub-steps. First was generating centroids. I decided to have the centroids be randomly generated. The method to generate the centroids involved randomly assigning every data point to any one of the clusters (the number of clusters is a parameter of this function). This

effectively created a random cluster mean (centroid) for each cluster. The second and most important step was data assignment. This was defined as the same function with a different override. The k-means function always kept two copies of the clusters collection. The centroids from the first cluster collection was used to assign data to the second. This caused the second cluster collection to have different centroids then the first because it was likely that the data points were assigned to different clusters during each data assignment. The second cluster replaced the first and a new cluster collection was created using the new centroids and so on. Each data assignment or update round was done until the centroids stopped moving (the cluster mean didn't change from one data assignment to the next). At this point the cluster collection was returned.

Step 3: Data Summarization

Once the cluster collection is returned the data is summarized in a report format to explain its contents. This step doesn't do an exceptional job of showing how accurate each cluster is to making an effective grouping of each letter but gives a general idea. In future uses of a similar clustering program it would be helpful to also be able to see outliers and a visualization of data point distances from the centroid. Since the data has several attributes it would not be appropriate for the scale of this assignment to design distance visualization. Although, I'd imagine it could be accomplished by displaying data points as polar coordinates around a centroid where the radius would be the Euclidian distance and the angle would be some sort of incremented value. In fact I haven't thought about this until now and actually it wouldn't be too difficult to implement as I already have the chart libraries needed.

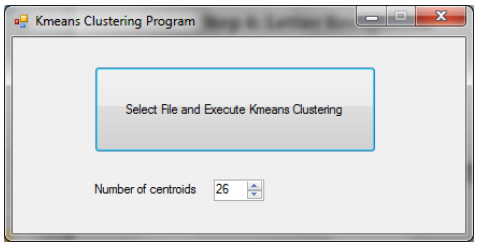
The data that is viewable from this step is the data points as assigned to clusters, the sum and mean of each attribute according to its cluster and a sampling of the letters originally assigned to the data points.

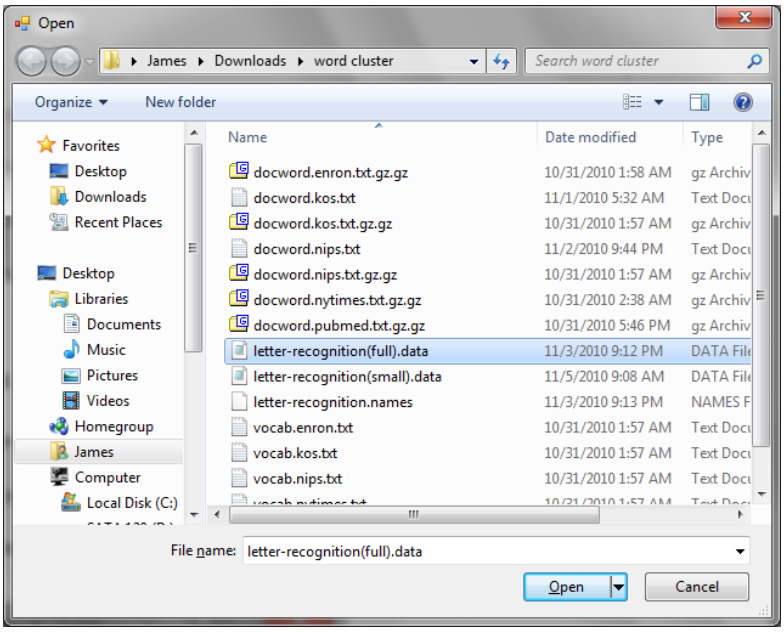
Step 4: Letter Recognition

The last step and solution to the clustering problem is the ability to determine the letter according to its attribute values. This is done by determining the closest centroid to the data point given the previously calculated clusters. It can then be determined what character the data point likely represents according to the character samplings from each cluster.

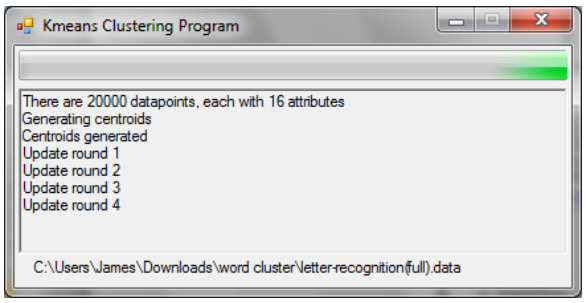
How to Run the Program

The program can be installed and immediately ran by running setup.exe. The dataset is provided as the text file within the provided zip file. Once the program starts the first step is choose the number of clusters to create and then to choose the dataset file.

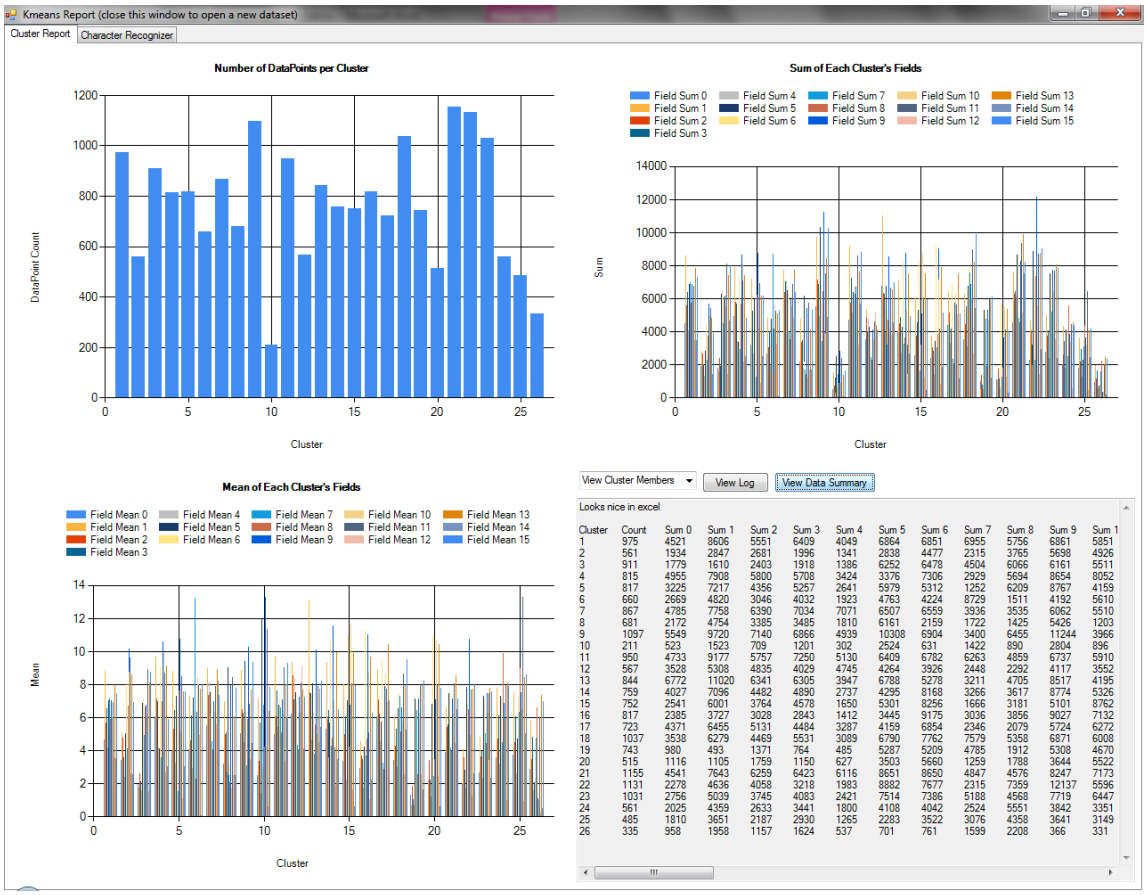




The next step is to

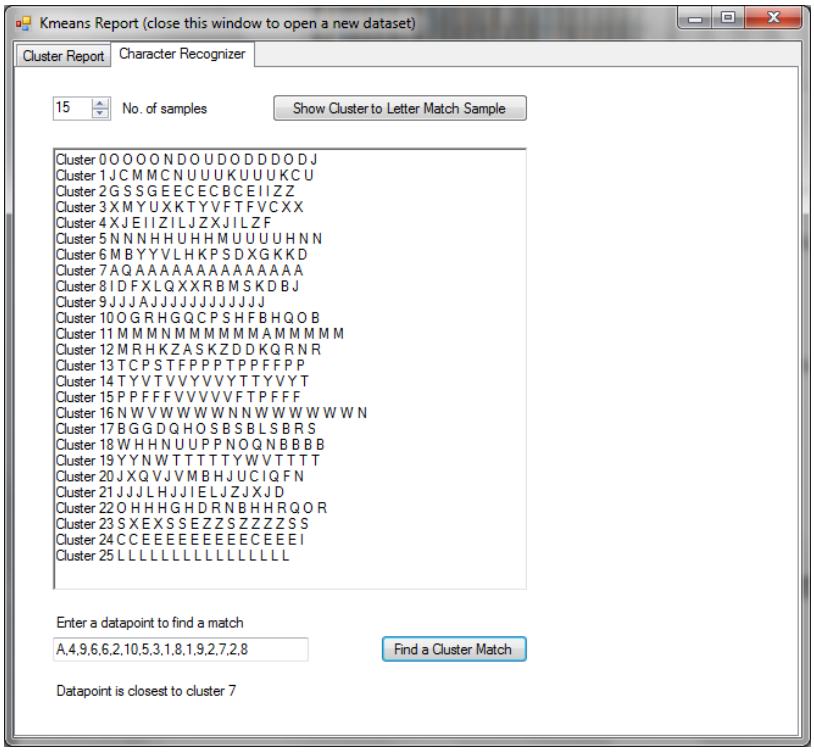


Once the clustering algorithm completes the data report page will display.



From here you can view the charts, data summary, console log and view the data points assigned to each cluster. The sum and mean charts are quite a mess unfortunately. They were originally designed for a dataset that had only three attributes.

The last part is the character recognizer which can be accessed from the tab on the top of the report form.



In this area you can see a sampling of the letter values in each cluster and also fund a cluster match for a given data point. In this example we see that the data point given was closest to cluster 7 which has mostly 'A' values. This indicates that the letter 'A' was successfully determined from its attributes.

Example 1:

```
There are 20000 datapoints, each with 16 attributes
…
It took 52 iterations and a total of 95.7947438 seconds to converge

Datapoint {A,4,9,6,6,2,10,5,3,1,8,1,9,2,7,2,8} is closest to cluster 20

Cluster 20 A Q J A A A A A A A A A A J A
```

Example 2:

```
There are 20000 datapoints, each with 16 attributes
…
It took 88 iterations and a total of 174.5910863 seconds to converge

Datapoint {P,6,10,6,6,4,7,10,5,2,11,5,4,4,11,5,7} is closest to cluster 20

Cluster 12 P P C P F P P P P P P P P P F
```

Example 3:

```
There are 20000 datapoints, each with 16 attributes
…
It took 114 iterations and a total of 193.7511105 seconds to converge

Datapoint {L,3,6,4,4,2,5,4,3,8,3,2,6,1,6,2,5} is closest to cluster 14
```

Cluster 14 L L L L L L L L L L L L L L L L

References

Frank, A. &. (2010). *UCI Machine Learning Repository*. Retrieved from Irvine, CA: University of California, School of Information and Computer Science: http://archive.ics.uci.edu/m

K-means - Interactive demo. (n.d.). Retrieved from http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

k-means clustering. (n.d.). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/K-means_clustering