CSE 601- Data Mining

Submitted in fulfillment of

Project 2- Clustering Algorithms

Submitted by:

Srinivasa Sai Kaushal Bondada
sbondada@buffalo.edu

Navinder jeet Mehrok
navinder@buffalo.edu

Submitted to:

Dr. Jing Gao
Assistant Professor

K Means clustering

In this clustering method, we split the given gene data into pre specified k clusters. The goal behind this approach is to find k clusters where the points in a cluster has minimum distance from that cluster mean

Implementation:

This algorithm is implemented in python, where we prepared two classes one for gene and other for centroid (cluster). Gene class instance holds the gene information such as gene number and the feature values in a list. Additionally it will also hold the centroid number it corresponds to. The second class will hold the information about the centroid which includes the centroid number and a list which will hold the mean values for each of the features. It will also hold the old list which contains value of last updated centroid feature mean values

Initially the user assigns a value for k i.e. total number of cluster, corresponding to those k instances of centroid class are created. We assigned random gene values to this k centroid. Random here implies the information of a randomly selected gene is copied into centroid instance. Then we measured the Euclidean distance between the various genes and the centroids. For each gene we compute the distance to all the centroid, the minimum distance centroid number is updated in that gene's centroid number field. In other words each gene is assigned to closest cluster. Once we assigned the values to each of the genes, then we take mean for all the genes in that cluster and updated the mean value for that cluster in all the features.

We repeat the above step till we have all the feature values in all centroids as same.

Advantages:

- As we increase the number of variable or genes the K means algorithm proves to be much faster than other such algorithms such as hierarchical in our case
- K cluster has distance calculation as the core value to predict or generate the next cluster thus it resulted in tighter clusters.

Limitaions:

- As we have to provide the k value in this algorithm it is difficult to predict total cluster values
- The intial step of initializing the cluster centroid is done randomly which may result in different clusters every time.

- K means doesn’t perform well when the shapes of clusters are irregular. Due to distance proximity check the cluster tend to have spherical shape which is not ideal for other shaped clusters.
- Density of a cluster also tend to distort the k means output

K Means clustering on Hadoop Map Reduce:

Hadoop provide a big platform for running the algorithms on big data. The architecture is fault tolerant and provides god reliability when executed on distributed system compared to other frameworks. Here we run the k means clustering algorithm on a single node system.

Implementation:

K means algorithm for map reduce is developed in java. First we will discuss key challenges for k means in map reduce:

1. The map reduce framework is stateless as each job is independent of the other
2. The exit check need to be performed so that to detect when the cluster means are not changing any more

These challenges are addressed in the implementation through the use of Counters and an external stored sequence file. Initially we have a data file with all the gene number, ground truth and feature values. Following are the steps describing the working of this algorithm:

1. The main function calls the initalizeCentroid method which uses the random values to select random genes to assign to the k clusters specified by the user. Then these centroid values are stored in a sequence file in the centroid directory.
2. Next we set the mapper and reduce classes to the job. We also initialized a counter which is going to help us in determining exit criteria.
3. The mapper runs, It first read the cluster centroid from the sequence file. Then it will take each gene and compare its feature values distance(Euclidean) from centroid of the each cluster. It will emit the cluster feature values as key and gene number with its feature values as the value.
4. Input to reducer is going to be the centroid value for each cluster as key and all the genes and their feature values in value. Key is going to be a text however the value is list of text. In reduce class we calculated the new mean for each cluster as centroid value and emit the cluster number as key and gene number as well as feature values(optional) as value. The new generated centroids are written to the sequence file. If the new centroid is same as previous obtained as key in reduce function then it will not increment the counter otherwise it will.
5. Once the job is done, in main function we check if the counter values is less than zero, which means all reducer runs didn't increment the cluster as it has same centroids as in previous run.

6. If so it will stop the execution, otherwise a new iteration begins

Advantages:

- As we increase the number of variable or genes the K means algorithm proves to be much faster than other such algorithms such as hierarchical in our case
- K cluster has distance calculation as the core value to predict or generate the next cluster thus it resulted in tighter clusters.

Limitaions:

- As we have to provide the k value in this algorithm it is difficult to predict total cluster values
- The intial step of initializing the cluster centroid is done randomly which may result in different clusters every time.
- K means doesn't perform well when the shapes of clusters are irregular. Due to distance proximity check the cluster tend to have spherical shape which is not ideal for other shaped clusters.
- Density of a cluster also tend to distort the k means output

Hierarchical clustering

Implementation:

In hierarchical clustering, we implemented the Min based clustering algorithm. Here the goal for a point is to find the cluster with the minimum distance of the closest point in the cluster. We store the all the genes in a list where each value in list represents a gene list which further store the gene feature value list in it.

So as start, we build a matrix to hold for distance between clusters. We find the minimum distance between two clusters. One important point to note here is that as we are using a matrix we need to exclude those cases where row and column have same values as they will corresponds to the same gene. Eg. I=2 and j=3 will have distance between gene 2 and 3, however the i=2 and j=2 will have distance between 2 and 2 which will always be 0. Once we have the minimum distance then we can combine the genes in a cluster. Next time we will be evaluating the boundary values for these clusters.

Advantages:

- This type of clustering is flexible, you can obtain the desired number of clusters by cutting the dendogram at proper level. No initial assumptions for number of clusters are required here.
- Hierarchical clustering relevant to practical taxonomies like the retail websites
- It provides different techniques to obtain desired number of clusters such as Max, Min and ward's approaches

Limitations:

- There is no minimization of a standard objective function
- As it is based upon merging of clusters, once two clusters are merged then there is no way of going back
- As hierarchical clustering includes various approaches, they are sensitive to different adverse factors such as noise, outliers, irregular clusters.

Density Based Clustering

Implemenation:

Density based clustering works on the concept of neibhourhood. We started with min neighbor values and epsilon values as provided at start.

We find the corepoint who have number of neighbor greater than the min value within epsilon. Then we procede to the non visited neighbor genes and calculate the neighbor, if total neighbor >min value then we add its neighbors to new cluster then iterate on neighbor. However if it has less neighbors then it is termed as border gene, add it to but don't iterate on its neighbors.

For each gene belonging to data set, check if gene is not classified yet and if it satisfy the requirement of core element. Then we iteratively find all genes that are density reachable from it and assign those points to new clusters.

Advantages:

- Since is works on neighbourhood concept for each element in cluster it adapts well to cluster who have different shapes as well as sizes.
- It is resistant to noise and outlier in a cluster thus providing accurate results.

Limitations:

- As it highly depends upon the input parameters which in include minimum points in neighbourhood and epsilon. Thus change in any of them shifts the whole cluster
- Furthermore if a cluster has varying densities in it then density based approach may mislead the output clusters.

Results

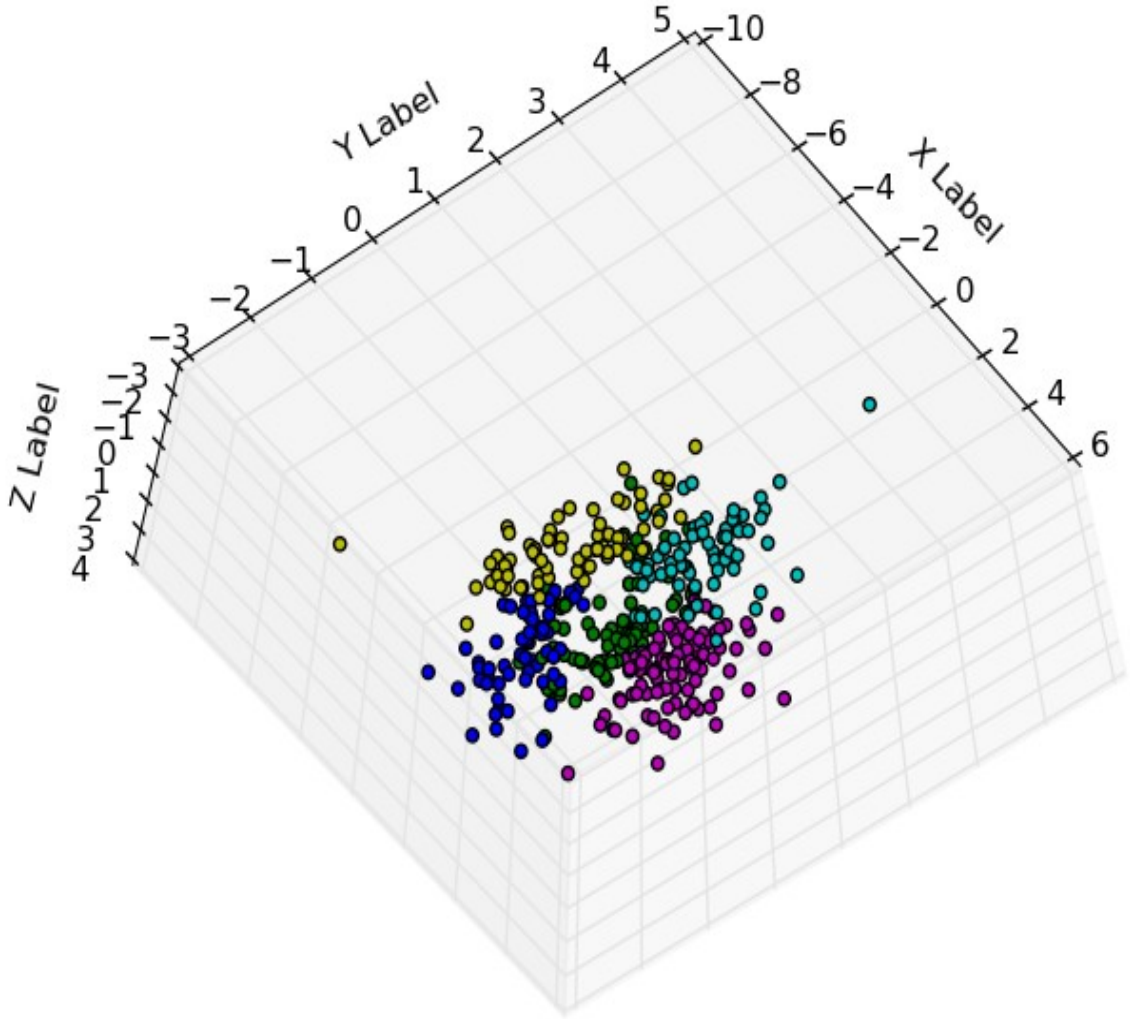
	DBSCAN_Cho.txt (e=4,minpt=8)	DBSCAN_iyer.txt (e=4,minpt=8)	KMEANS_cho (n=5)	KMEANS_iyer (n=10)	Hirarchical_cho (n=5)	Hirachical_iyer (n=10)
Jaccard	0.19598187906243844	0.2505390748687236	0.35717223227504524	0.3232622306148078	0.22839497757358454	0.1582430969664285 8
Rand index	0.50685924454347 76	0.6085959392268294	0.7901822867727992	0.7237970885446 09	0.2402749067089049 5	0.1882868355974245
Correlation	-0.350799570606	-0.408067881492	-0.4503249042	-0.375393690844	-0.308898691207	-0.674469761187

PCA Plots

Plot for Kmeans

For Data set : cho.txt

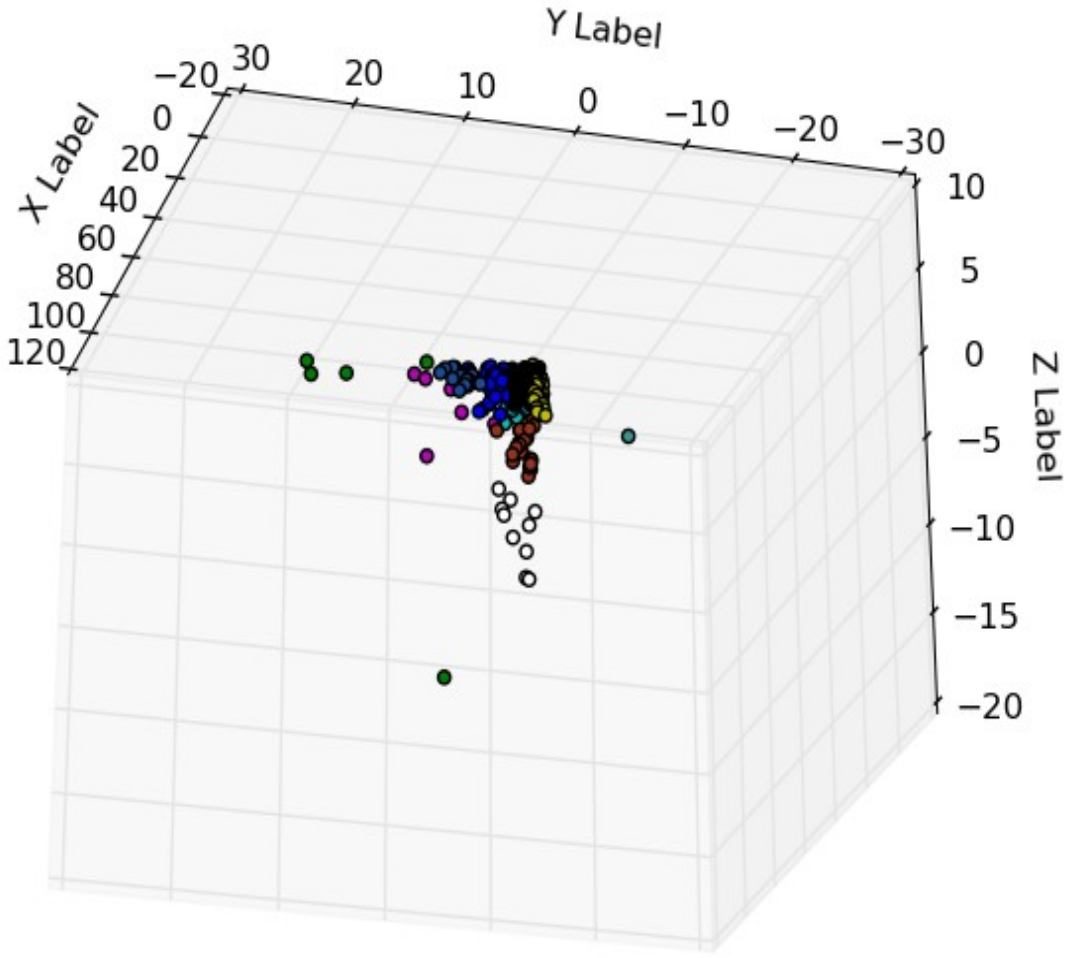
Total clusters(K value)= 5



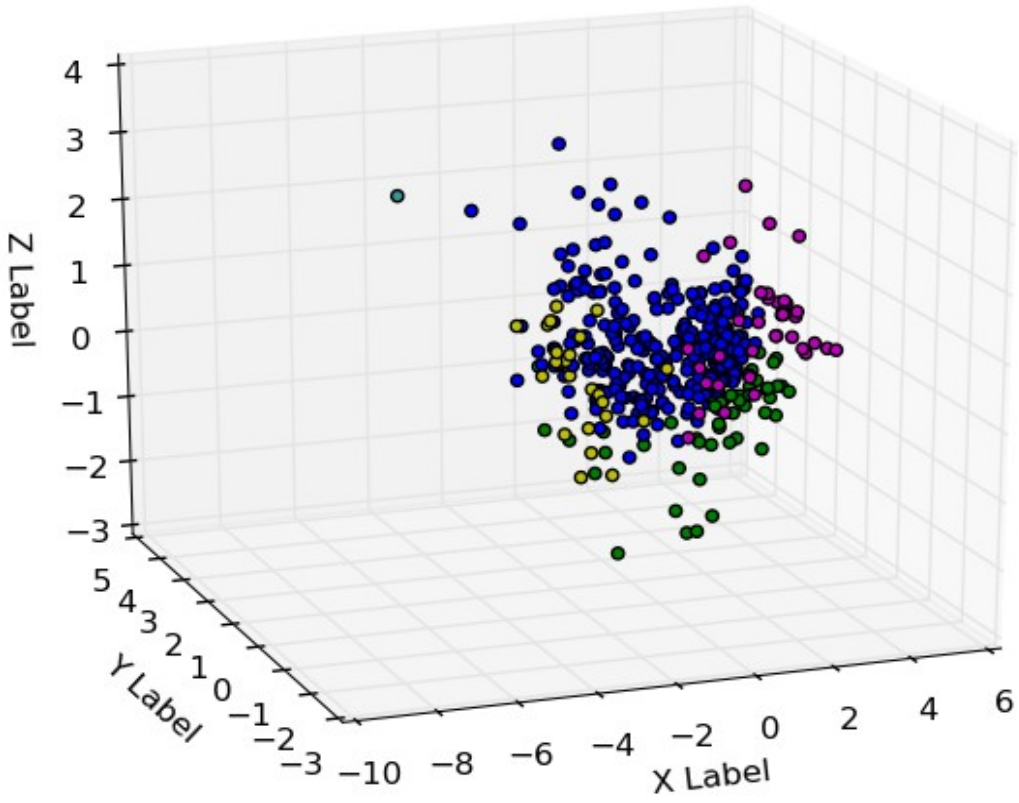
It clearly differentiates the clusters

For data set : Iyer.txt

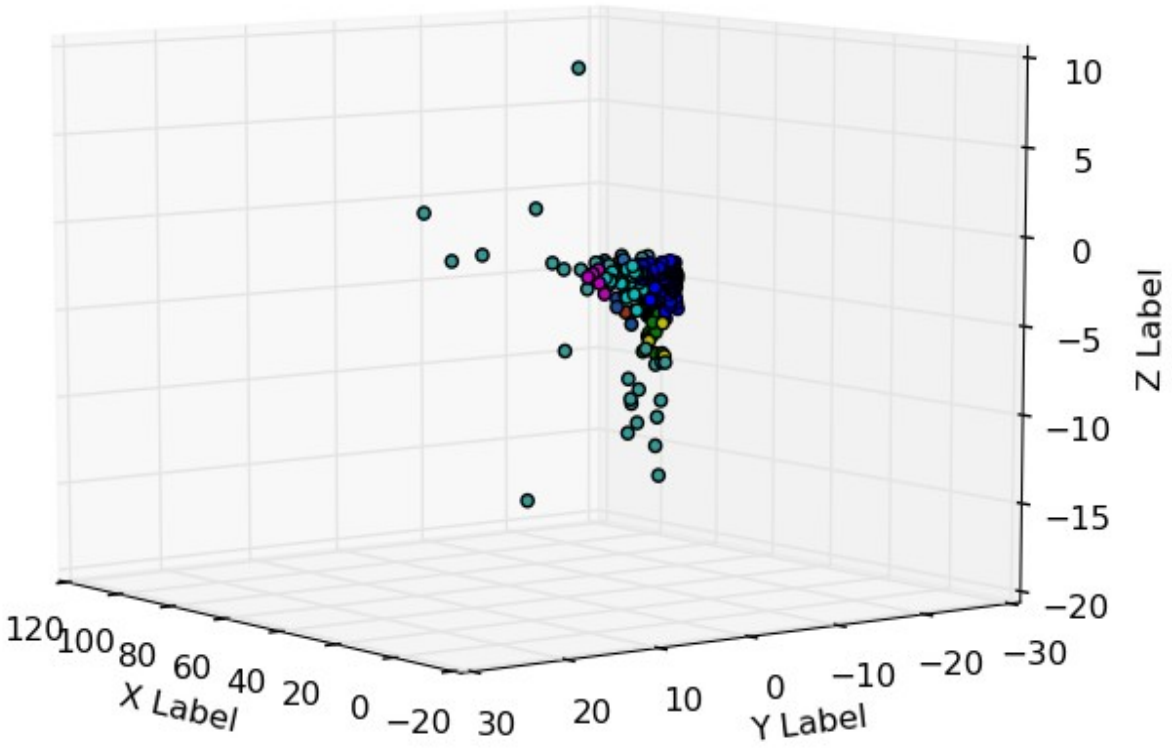
K value: 10



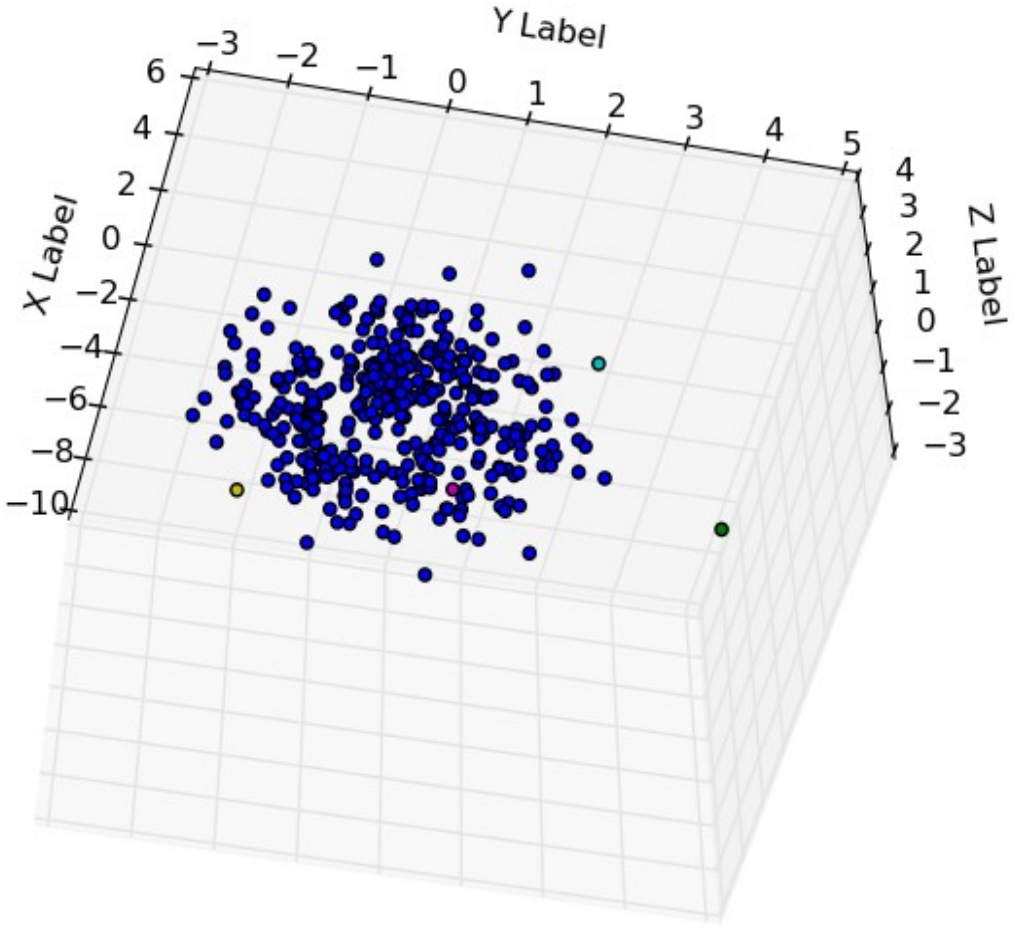
DB scan plot for cho.txt with epsilon value as 4 and min points as 8



DBSCAN for iyer.txt with epsilon as 4 and minpoints as 8



Hirarchical for cho.txt with 5 clusters:



Hirarchical for iyer.txt with 10 clusters: