Jay Patel
CS422
11/19/18

Map Reduce and Cluster

Map Reduce

We have been studying about map reduce since a long time. This part of the assignment deals with implementing map reduce model for computing even number integer across all files.

For this model, we will require **one map reduce pair.**
The first mapper will take input **FILE** as an input.

Psuedocode :

```
FIRST MAPPER
Input — Key: File ID and value: file content
Output — Key: Mapper ID and value: total number of even integer in file

Mapper ( file ID, content) :
        even_count = 0
        string token  = string.split(\n)
        for string token :
                if even int —> even_count + 1
        emit ( mapper id, even_count)

COMBINER
Input — Key : mapper ID, value : count
Output — 1, value: total number in collection ( total number)

Combine(map ID, count)
        int sum
        int N
        for key value pair
                sum + even count
        emit(1, sum)
REDUCER
This will calculate percentage of even

input— key : 1, value: even int per node
Output — key: 1 , value: percentage

Reduce ( 1, pair( sum , node)
        sum = 0
        total = 0
        for pair(sum, node)
                sum + 1
                node + 1
        emit(1, percentage)
```

We will only require one job since it will finish the task in one cycle. The mapper will count the ven, combiner will combine the total count and the nodes, and then reducer will convert into percentage.

Clustering

A.

We will be observing the Kmeans cluster algorithm from the sci-kit learn library, and use the code provided in the assignment prompt. In this part, we will be observing the situation where the algorithm performs well, and in cases where it does not. For the same, we will be manipulating the *k* Values in the code and observe the changes.

<insert graphs here>

The graphs are distributed based on the type of cluster they have formed, and along with the number of cluster they are distributed in. In the case of k=2 (*Incorrect Number of Blobs*), we can see that the there have formed two different cluster. When we formulate the SSE, the centroid is calculated between those two cluster and we can conclude that with k=2, the cluster would be relatively large.

Taking a look at the elongated graphs, this type of transformation, the data points are more spread out. Taking specific look at k=3 (Anisotropicyl Distributed Blobs), you can notice that the purple colored cluster, the cluster has been classified close to each other, however the yellow cluster is not, since the data points are elongated, its difficult for k-means to detect cluster in that type of distribution. K-means only considers shortest distance.

Looking at all 4 types of graph for k=3 seems to be properly distributed and close to the centroid, and be able to detect its cluster. They are almost equal distances. As we increase the number of cluster, the clusters get defined closely to each other. In case of k=7, the clusters are almost evenly distributed into 7 segments across the graph. As the number of data points increases, the this algorithm fails to understand the densities in the dataset, and therefore we tend to use algorithms like DBSCAN.

Another thing to notice in the data set in k=3, is that we can notice 3 types of densities in the cluster, one with large, one with medium, and one with least dense data points. Such larger dataset are assigned with more weight to avoid inter cluster group formation.

So, how do we predict the appropriate k value? For this, we would use the algorithm called elbow-method. In this process, the algorithm iterates through k values from 1 to 10 and looks at the sum of the squared values of the weight. Looking at the graph, attached, we can notice that the elbow starts dropping at lower angle starting at k=3. Hence, we can say that K=3 is the optimal for this particular dataset.

Conclusion: As observed, python uses euclidian distance to compute distance between the data points. We can conclude that the k-means algorithm works perfectly fine with evenly grouped dataset. It makes two assumptions 1) Shape of the cluster (Spehrical or uneven) 2) Clusters are similar in size. In our case, as the K increases, the data cluster becomes more granular and the chi-squared score tends to move to 0.

B.

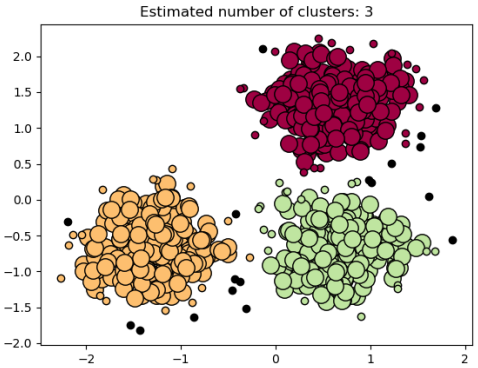
DBSCAN algorithm is a clustering algorithm that groups together points that are close to each other using specific parameters eps and minPoint. It will take out the dense cluster, and leave the noise in the background. The eps provides a threshold for the minimum distance between 2 points. The minPoints parameter tells the algorithm to specify minimum number of close points to declare it as dense. At high density areas, the core points are present.

The following table shows the three variation combination of the two parameters; eps and minPoints.

Epsilon	min_sample	Cluster	Homogenity	Completeness	Vmeasure	Silhoutte Coefficient
0.17	10	3	0.791	0.629	0.7	0.444
0.3	10	3	0.953	0.882	0.917	0.626
0.4	10	1	0.001	0.06	0.002	0.061
0.17	5	3	0.903	0.755	0.823	0.564
0.3	5	1	0.001	0.019	0.002	0.106
0.4	5	1	0.001	0.144	0.003	0.023
0.17	3	8	0.938	0.741	0.828	0.184
0.3	3	2	0.003	0.048	0.005	-0.082
0.4	3	1	0.001	0.144	0.003	0.023

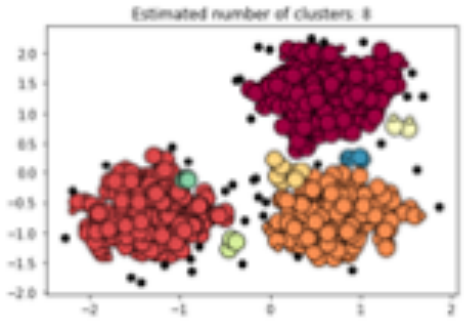
In this table, we look at the different values that defines the appropriate outcome

Metric	Definition	Function	Value Returned(range)	Expected Value for perfect results
Homogenity	If all the clusters contain only data points which belonging to single class	homogeneity_score (labels_true, labels_pred)	0.0 – 1.0	1.0
Completeness	All the data points that are members of a given class are elements of the same cluster	completeness_score (labels_true, labels_pred)	0.0 – 1.0	1.0
V-Measure	harmonic mean between homogeneity and completeness	v_measure_score (labels_true, labels_pred)	0.0 – 1.0	1.0
Silhouette Coefficient	how far away the datapoints in one cluster are, from the datapoints in another cluster	silhouette_score(X, labels, metric='euclidean', sample_size=None, random_state=None, **kwds)	-1 to 1	best value = 1 worst value = -1 values near 0 – overlapping clusters



The initial code states the default value of eps = 0.3 and min_samples = 10, with which, we get out of 3 clusters. Looking at the other details such as homegeniety is 0.953 which indicates that its not perfect labeling. Thats why it further splits classes into more clusters which are almost homogeneous. Same thing goes with the completeness of the cluster which is 0.883.

The silhoutte value indicates that there are inter cluseter splits present.
In the graph above, we can notice that denser circles are where the core points exist. We know this because all the point surrounded those larger circles are under eps = 0.3. since these points are reachable with min_sample, they are considered into one cluster.



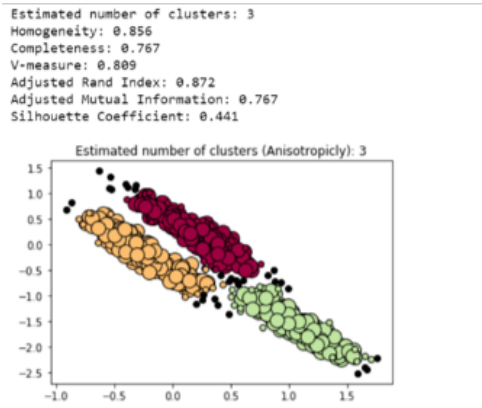
The graph above indicates epsilon = 0.17 and min_sample = 3, looking at the table we notice that it give the cluster has good values which are closer to one. However you might notices noises into the cluster. In the process, number of cluster that were formed, were 8, and it depicts that there were many inter- cluster mean and near cluster mean are formed resulting into over lapping.

Conclusion :

We can gauge the cluster formation based on the parameters, and we can analyze that higher epsilon values always less than 2, regardless of the value of min samples. Larger min values result in more significant cluster using dataset with noise. For this particular dataset, the best values that we got the closest results to 1 is eps to be 0.3 and min_sample to be 10.

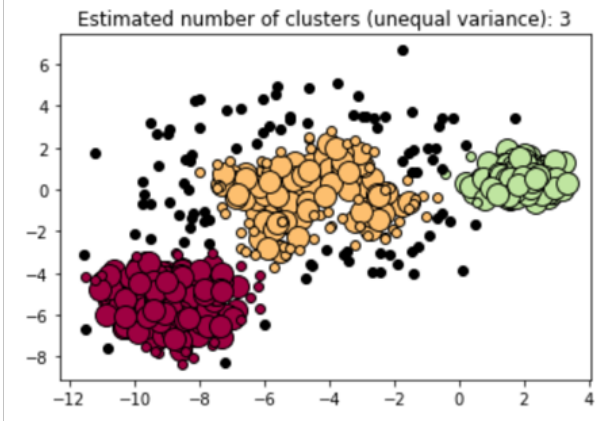
C.

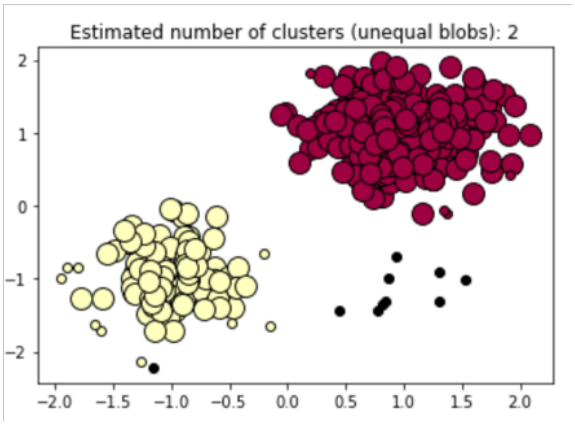
In part A, we used various transformation over our dataset to describe various scenarios in our graph, however we failed to understand the effect of densities in the dataset.



Analyzing the graph above, we came to see that, when eps = 0.15 and min_sample = 10. The graph provides optimal results compared to the datsets in part A k means. This graph is for the Anisotropicly transformation. The value of k did not matter in this case, since the dataset cluster is not spherical in shape. Due to elliptical shape, the data points get closer to each other which in turn gets better score due to the value of epsilon. Selecting smaller epsilon value will result into smaller radius for threshold for min distance.

```
Estimated number of clusters: 3
Homogeneity: 0.920
Completeness: 0.769
V-measure: 0.838
Adjusted Rand Index: 0.834
Adjusted Mutual Information: 0.769
Silhouette Coefficient: 0.578
```





For unequal variance transformation, the best result is provided at eps = 0.93 and min_sample = 12 in reference to part A. We also notice a lot of outliers and noise with this eps value.

Conclusion:
Compared to k-means, the DBSCAN for such transformation does a better job since it detects clusters densities and also detects outlier.

D.

What is 20 News Group?

- The 20 News Group is a dataset consisting of approximately 20,000 news related documents, that are partitioned evenly among 20 groups. This dataset was originally collected by Ken Lang.
- The 20 Groups are divided based on different topics they talk about. Some of the data groups are closely related to each other which can cause ambiguity while performing Kmeans.

Trying different values for K
This section will show the variation in different measures that we obtain at each cluster change.

Metric	Definition	Function	Value Returned(range)	Expected Value for perfect results
Homogenity	If all the clusters contain only data points which belonging to single class	homogeneity_score (labels_true, labels_pred)	0.0 – 1.0	1.0
Completeness	All the data points that are members of a given class are elements of the same cluster	completeness_score (labels_true, labels_pred)	0.0 – 1.0	1.0
V-Measure	harmonic mean between homogeneity and completeness	v_measure_score (labels_true, labels_pred)	0.0 – 1.0	1.0
Silhouette Coefficient	how far away the datapoints in one cluster are, from the datapoints in another cluster	silhouette_score(X, labels, metric='euclidean', sample_size=None, random_state=None, **kwds)	-1 to 1	best value = 1 worst value = -1 values near 0 – overlapping clusters

Once we have defined all the metric, now we will try different values of k and then see the difference.

K value	Homogenity	Completeness	Vmeasures	Silhouette Coef.
5	0.503	0.494	0.499	0.008
10	0.493	0.338	0.401	0.008
20	0.517	0.285	0.367	0.013
50	0.546	0.247	0.340	0.017

Observing the table, we can see that as the cluster increases in size, the value of completeness decreases, the value of homogenity stays the same, and value of silhouette coef. Starts growing, which indicate less overlap.

Now we will check the different transformation and observe the change.

Transformation	Homogenity	Completeness	Vmeasures	Silhouette Coef.
LSA = 10 This will preprocess documents with latent sementicanalysis	0.695	0.225	0.374	0.186
idf This will disable the idf	0.396	0.446	0.418	0.010
hashing This trasformation will use the hashing feature vectorizer	0.320	0.449	0.374	0.005
Max num = 10 This will provide number of dimensions.	0.127	0.132	0.129	0.212