Project 3 Report

Neer Patel

***Abstract*—The following different algorithms were used to identify clusters in a set of two-dimensional points: KMeans, KMeans++, Bisecting KMeans, Agglomerative Clustering, DBSCAN, and Mini-Batch KMeans. Each algorithm's performance was determined using a Silhouette Score and further verified using visual inspection as needed. The DBSCAN algorithm performed the best across all of the data files because it was able to handle the noise present in each one. Additionally, it did not require as many clusters to identify the general shapes present in each of the data files.**

***Index Terms*—KMeans, KMeans++, Bisecting KMeans, Agglomerative Clustering, DBSCAN, Mini-Batch KMeans, clustering, Silhouette Score**

I. Introduction

The purpose of this project was to analyze and apply the following clustering algorithms on to five given data sets. The algorithms used were KMeans, KMeans++, Bisecting KMeans, Agglomerative Clustering, DBSCAN, and Mini-Batch KMeans++. The data provided consisted of clusterings in the shapes of being dispersed, a maze, objects, rings, and waves. The two-dimensional data files also consisted of noise – the noise was not removed from the data in order to analyze how well each algorithm could handle it.

II. Results and Discussion

There were two measures used to determine the performance of each algorithm on each image: Silhouette Score and visual inspections. Heavy emphasis was placed on the Silhouette Score when choosing the best parameters, since it numerically defines the algorithms performance. The reason for choosing this numerical metric was to provide a way to compare the performance of each algorithm with the others and between trials. A visual inspection of the plots was used when the Silhouette Score seemed unreasonable. For each of the data files in the algorithms, the bolded entries in the trials table marks the best parameters.

A. Original Data

Below are the images of each of the provided data sets in their respective order: clustersDispersed.txt, clustersMaze.txt, clustersObjects.txt, clustersRings.txt, and clustersWaves.txt.

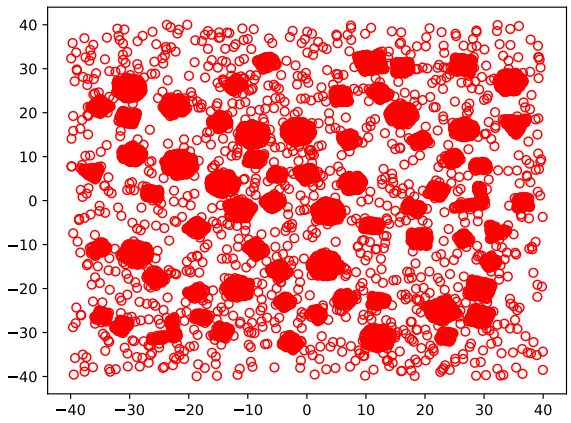

Fig. 1. clustersDispersed.txt

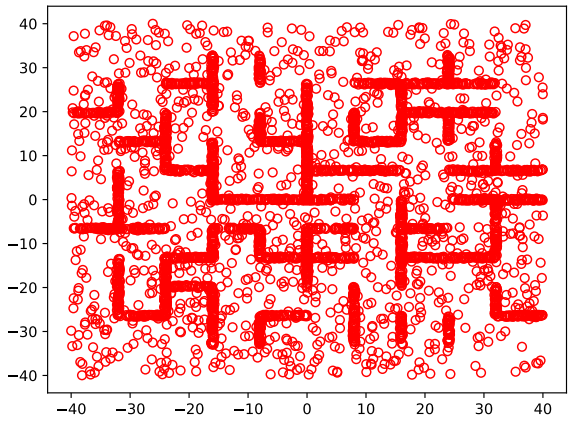

Fig. 2. clustersMaze.txt

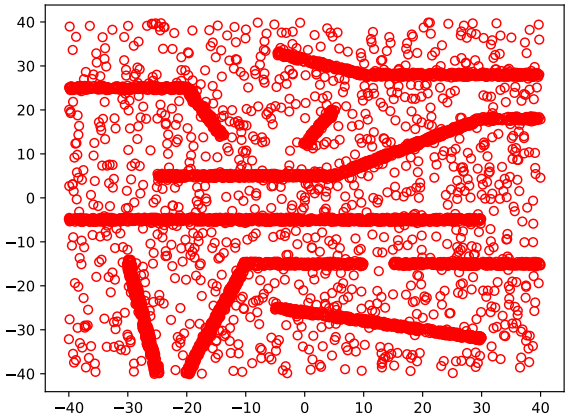

Fig. 3. clustersObjects.txt

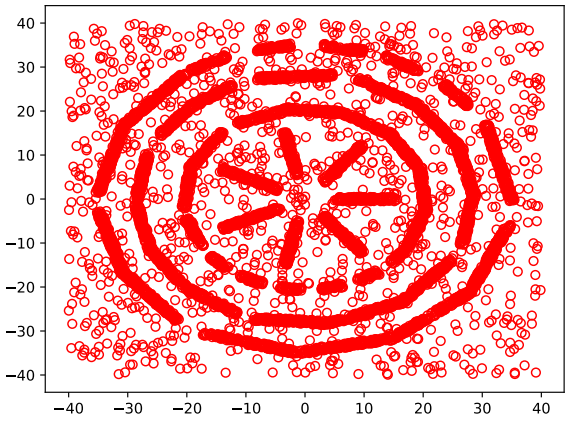

Fig. 4. clustersRings.txt

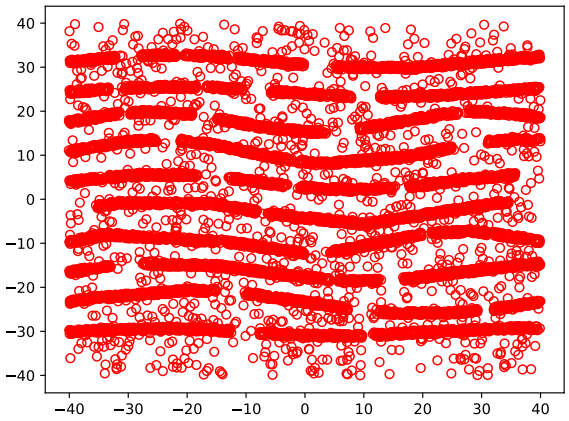

Fig. 5. clustersWaves.txt

B. KMeans

The main parameters that were analyzed for KMeans were the number of clusters, the number of iterations to run the algorithm, the maximum number of iterations, tolerance, and the KMeans algorithm to use (lloyd vs elkan). After running a few test trials, lloyd had a much better runtime and practically the same accuracy, so it was favored. Additionally, considering both runtime and accuracy, n_init, max_iter, and told were determined. The biggest focus for algorithm was determining the number of clusters.

1) Dispersed: The table representing some of the trials is as follows:

clustersDispersed.txt

n_clusters	n_init	max_iter	tol	algo	silhouette_score
63	10	300	1.00E-04	lloyd	0.557
63	50	300	1.00E-04	lloyd	0.563
63	100	300	1.00E-04	lloyd	0.588
63	200	300	1.00E-04	lloyd	0.574
63	100	300	1.00E-04	elkan	0.569
63	100	500	1.00E-04	lloyd	0.568
60	100	300	1.00E-04	lloyd	0.581
61	100	300	1.00E-04	lloyd	0.555
62	100	300	1.00E-06	lloyd	0.576
64	100	300	1.00E-06	lloyd	0.578
65	100	300	1.00E-06	lloyd	0.578
66	100	300	1.00E-06	lloyd	0.582
63	100	300	1.00E-03	lloyd	0.566
50	100	300	1.00E-03	lloyd	0.554
70	100	300	1.00E-03	lloyd	0.582
100	100	300	1.00E-03	lloyd	0.552
63	**300**	**300**	**1.00E-04**	**lloyd**	**0.594**
63	500	300	1.00E-04	lloyd	0.565

The image representing these parameters is as follows:

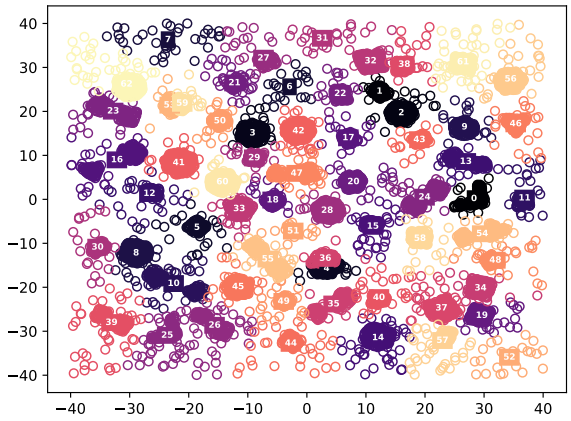

Fig. 6.

2) Maze: The table representing some of the trials is as follows:

clustersMaze.txt

n_clusters	n_init	max_iter	tol	algo	silhouette_score
50	**300**	**300**	**1.00E-04**	**lloyd**	**0.447**
50	300	300	1.00E-04	elkan	0.443
52	300	300	1.00E-04	elkan	0.44
52	300	300	1.00E-04	lloyd	0.444
48	300	300	1.00E-04	lloyd	0.43
49	300	300	1.00E-04	lloyd	0.431
60	300	300	1.00E-04	lloyd	0.437
50	300	300	1.00E-06	lloyd	0.437
50	300	300	1.00E-08	lloyd	0.446
50	300	300	1.00E-10	lloyd	0.439

The image representing these parameters is as follows:

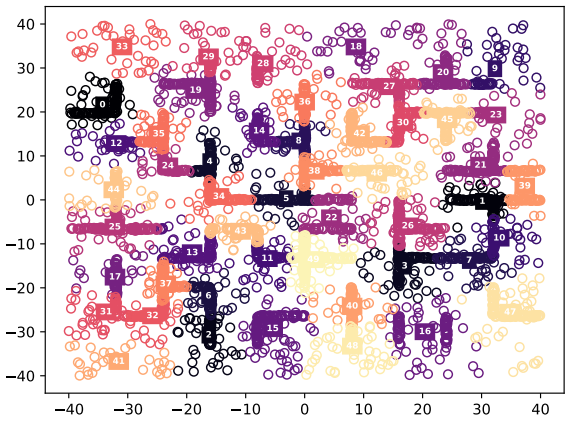

Fig. 7.

3) Objects: The table representing some of the trials is as follows:

clustersObjects.txt

n_clusters	n_init	max_iter	tol	algo	silhouette_sc
23	300	300	1.00E-04	lloyd	0.457
30	**300**	**300**	**1.00E-04**	**lloyd**	**0.481**
30	300	300	1.00E-04	elkan	0.478
35	300	300	1.00E-04	lloyd	0.466
32	300	300	1.00E-04	lloyd	0.473
31	300	300	1.00E-04	lloyd	0.475

The image representing these parameters is as follows:

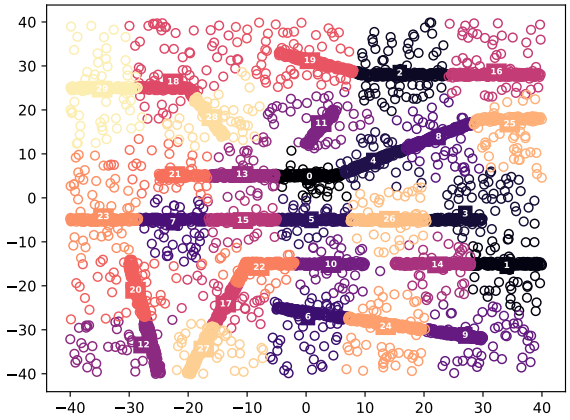

Fig. 8.

4) Rings: The table representing some of the trials is as follows:

clustersRings.txt

n_clusters	n_init	max_iter	tol	algo	silhouette_score
70	300	300	1.00E-04	lloyd	0.491
67	300	300	1.00E-04	lloyd	0.487
65	300	300	1.00E-04	lloyd	0.481
72	300	300	1.00E-04	lloyd	0.492
62	300	300	1.00E-04	lloyd	0.474
66	300	300	1.00E-04	lloyd	0.484
63	300	300	1.00E-04	lloyd	0.475
69	300	300	1.00E-04	lloyd	0.488
68	**300**	**300**	**1.00E-04**	**elkan**	**0.492**
68	300	300	1.00E-04	lloyd	0.482

The image representing these parameters is as follows:

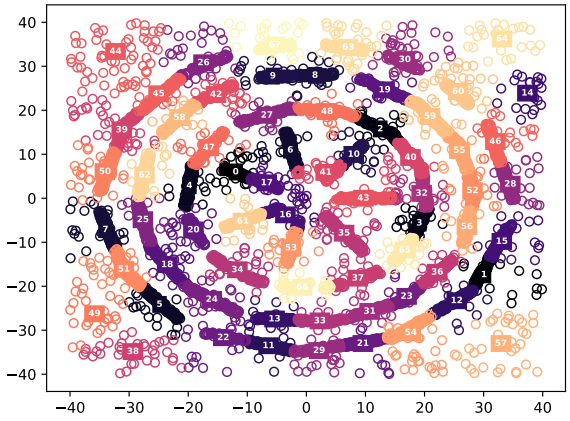

Fig. 9.

5) Waves: The table representing some of the trials is as follows:

clustersWaves.txt

n_clusters	n_init	max_iter	tol	algo	silhouette_scor
50	300	300	1.00E-04	lloyd	0.414
80	300	300	1.00E-04	lloyd	0.473
85	300	300	1.00E-04	lloyd	0.479
87	300	300	1.00E-04	lloyd	0.474
93	**300**	**300**	**1.00E-04**	**lloyd**	**0.489**
94	300	300	1.00E-04	lloyd	0.487
93	300	300	1.00E-04	elkan	0.479

The image representing these parameters is as follows:

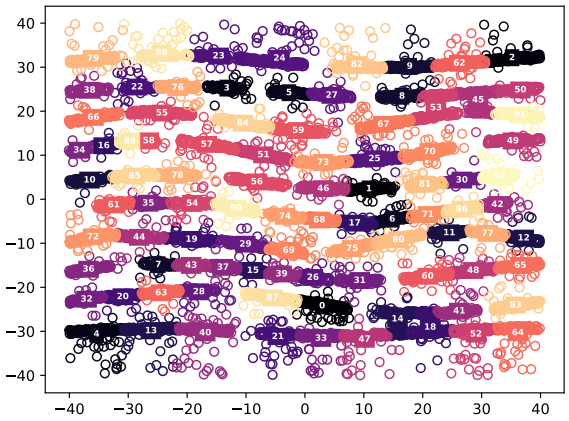

Fig. 10.

6) Analysis: As can be observed from the images, KMeans cannot filter out noise and is greatly affected by it. As a result of the noise, many of the cluster centers moved to towards the noise.

C. KMeans++

Th only difference between KMeans++ and KMeans is that KMeans++ selects initial clusters using sampling based on some probability distribution. As a result, the number of clusters parameter and the algorithm were slightly altered, but the other parameters remained the same: n_init, max_iter, and tol.

1) Dispersed: The table representing some of the trials is as follows:

clustersDispersed.txt

n_clusters	n_init	max_iter	tol	algo	silhouette_s
63	300	300	1.00E-04	lloyd	0.615
63	300	300	1.00E-04	elkan	0.613
59	300	300	1.00E-04	lloyd	0.604
62	300	300	1.00E-04	lloyd	0.611
64	300	300	1.00E-04	lloyd	0.616
65	300	300	1.00E-04	lloyd	0.618
66	300	300	1.00E-04	lloyd	0.623
67	300	300	1.00E-04	lloyd	0.622
68	300	300	1.00E-04	lloyd	0.627
69	**300**	**300**	**1.00E-04**	**lloyd**	**0.63**
70	300	300	1.00E-04	lloyd	0.624
69	300	300	1.00E-04	elkan	0.629
70	300	300	1.00E-04	elkan	0.63
68	300	300	1.00E-04	elkan	0.629

The image representing these parameters is as follows:

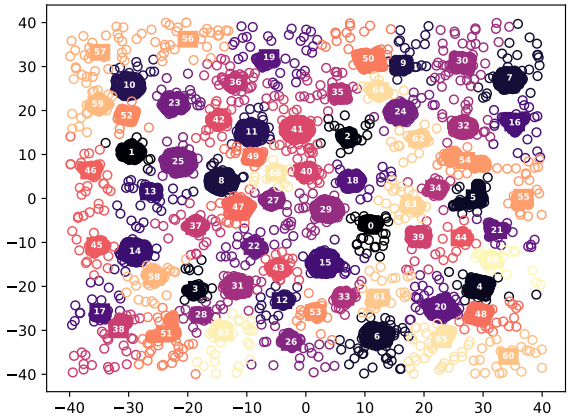

Fig. 11.

2) Maze: The table representing some of the trials is as follows:

clustersMaze.txt

n_clusters	n_init	max_iter	tol	algo	silhouette_score
50	300	300	1.00E-04	lloyd	0.437
52	300	300	1.00E-04	lloyd	0.437
48	300	300	1.00E-04	lloyd	0.436
49	300	300	1.00E-04	lloyd	0.436
51	300	300	1.00E-04	lloyd	0.437
50	300	300	1.00E-04	elkan	0.434
52	300	300	1.00E-04	elkan	0.438
51	**300**	**300**	**1.00E-04**	**elkan**	**0.44**

The image representing these parameters is as follows:

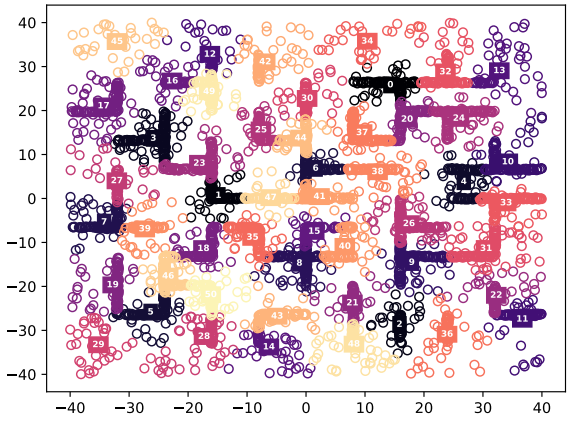

Fig. 12.

3) Objects: The table representing some of the trials is as follows:

clustersObjects.txt

n_clusters	n_init	max_iter	tol	algo	silhouette_score
30	**300**	**300**	**1-e4**	**lloyd**	**0.481**
31	300	300	1-e4	lloyd	0.48
29	300	300	1-e4	lloyd	0.489
30	300	300	1-e4	elkan	0.481

The image representing these parameters is as follows:

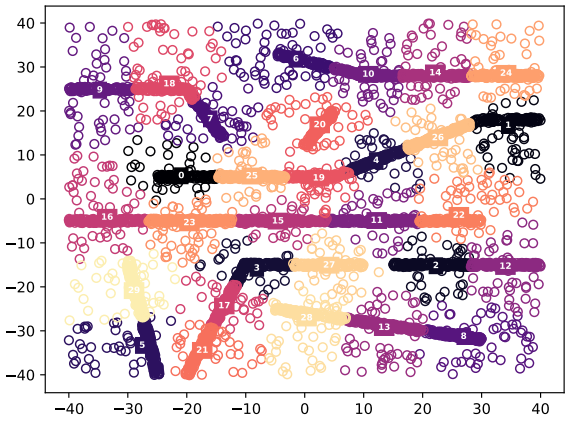

Fig. 13.

4) Rings: The table representing some of the trials is as follows:

clustersRings.txt

n_clusters	n_init	max_iter	tol	algo	silhouette_score
68	300	300	1.00E-04	lloyd	0.489
66	300	300	1.00E-04	lloyd	0.49
67	300	300	1.00E-04	lloyd	0.492
70	**300**	**300**	**1.00E-04**	**lloyd**	**0.495**
70	300	300	1.00E-04	elkan	0.491

The image representing these parameters is as follows:

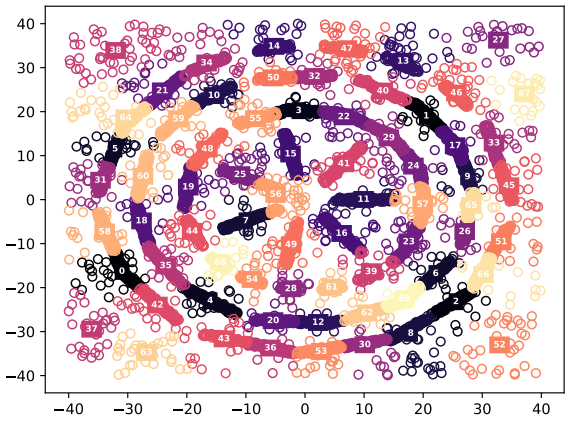

Fig. 14.

5) Waves: The table representing some of the trials is as follows:

clustersWaves.txt

n_clusters	n_init	max_iter	tol	algo	silhouette_score
93	300	300	1.00E-04	lloyd	0.493
95	300	300	1.00E-04	lloyd	0.494
97	300	300	1.00E-04	lloyd	0.494
93	**300**	**300**	**1.00E-04**	**elkan**	**0.495**
95	300	300	1.00E-04	elkan	0.491

The image representing these parameters is as follows:

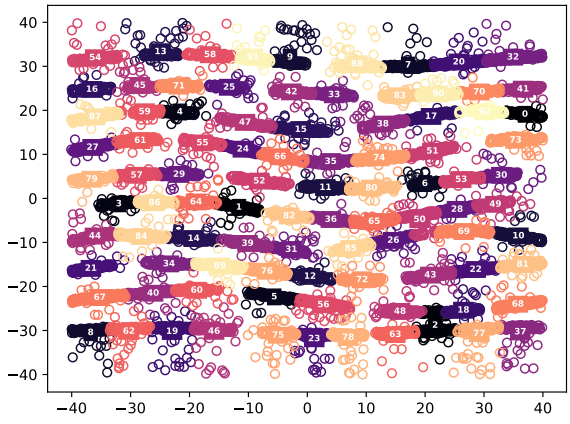

Fig. 15.

6) Analysis: As seen from the images, there is still a lot of noise, and the affects of it are still present. But the cluster centers are generally where we would expect them to be.

D. Bisecting KMeans

The parameters for Bisecting KMeans are very similar to KMeans, so what worked well previously was kept the same. However, there is a new parameter to determine the bisecting strategy that is used. The trials below consider the combination of the KMeans algorithm and bisecting strategy. Note that KMeans++ is being used to improve the choices of the initial clusters.

1) Dispersed: The table representing some of the trials is as follows:

clustersDispersed.txt

n_clusters	algo	bisect_algo	silhouette_score
69	lloyd	intertia	0.455
69	**elkan**	**intertia**	**0.552**
69	elkan	cluster	0.455
69	lloyd	cluster	0.455

The image representing these parameters is as follows:

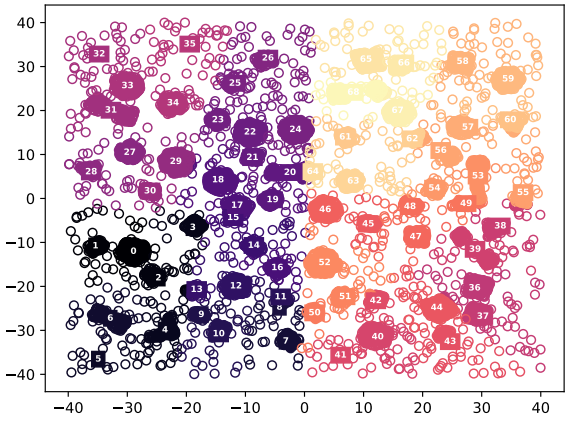

Fig. 16.

2) Maze: The table representing some of the trials is as follows:

clustersMaze.txt			
n_clusters	algo	bisect_algo	silhouette_score
51	lloyd	inertia	0.3642
51	**elkan**	**inertia**	**0.3643**
51	elkan	cluster	0.358
51	lloyd	cluster	0.358

The image representing these parameters is as follows:

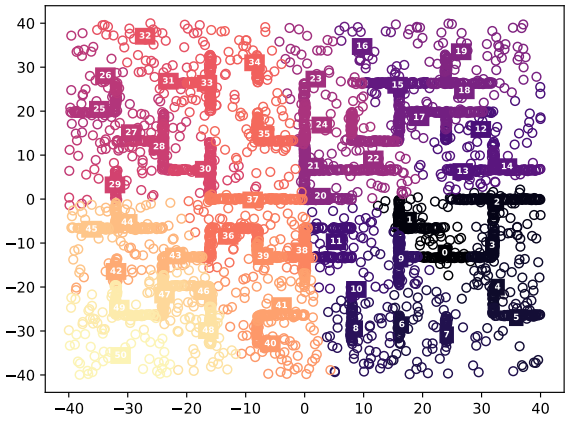

Fig. 17.

3) Objects: The table representing some of the trials is as follows:

clustersObjects.txt			
n_clusters	algo	bisect_algo	silhouette_score
30	lloyd	inertia	0.405
30	elkan	inertia	0.405
30	elkan	cluster	0.408
30	lloyd	cluster	0.411
40	lloyd	inertia	0.413
40	lloyd	cluster	0.41
40	**elkan**	**inertia**	**0.421**

The image representing these parameters is as follows:

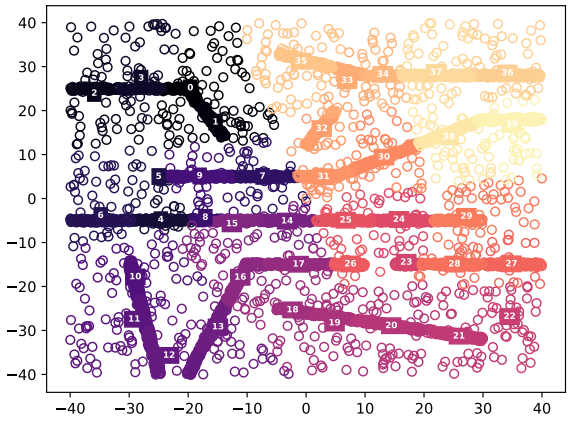

Fig. 18.

4) Rings: The table representing some of the trials is as follows:

clustersRings.txt			
n_clusters	algo	bisect_algo	silhouette_score
70	lloyd	inertia	0.425
70	elkan	inertia	0.425
70	elkan	cluster	0.422
70	**lloyd**	**cluster**	**0.423**

The image representing these parameters is as follows:

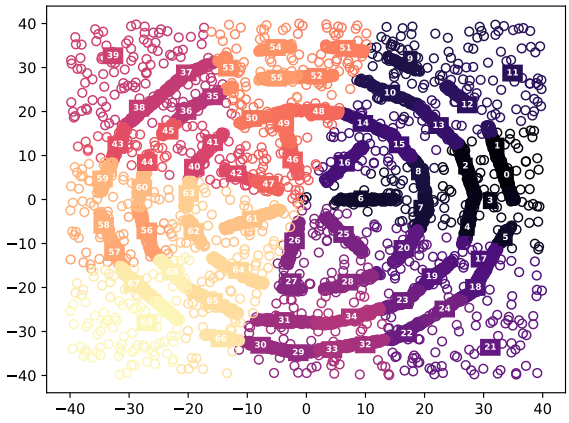

Fig. 19.

5) Waves: The table representing some of the trials is as follows:

clustersWaves.txt			
n_clusters	algo	bisect_algo	silhouette_score
110	**lloyd**	**inertia**	**0.443**
110	elkan	inertia	0.443
110	elkan	cluster	0.439
110	lloyd	cluster	0.44

The image representing these parameters is as follows:

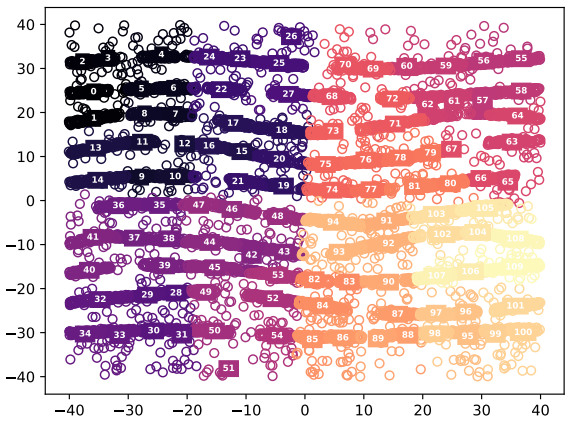

Fig. 20.

6) Analysis: As can be observed from the images, noise is still an issue. Ignoring the noise, the cluster centers seem to be in the correct, general area.

E. Agglomerative Clustering

This method works by recursively merging pairs of clusters. From the previous algorithms, a good number of clusters have already been determined for each of the data files. The main parameter to analyze for this method was the linkage criterion used: ward, complete, average, or single. Note that euclidean distance was used as the metric – this is because the data represents points in two dimensions.

1) Dispersed: The table representing some of the trials is as follows:

clustersDispersed.txt		
n_clusters	link_algo	silhouette_score
72	**ward**	**0.614**
72	complete	0.5
72	average	0.546
72	single	-0.56

The image representing these parameters is as follows:

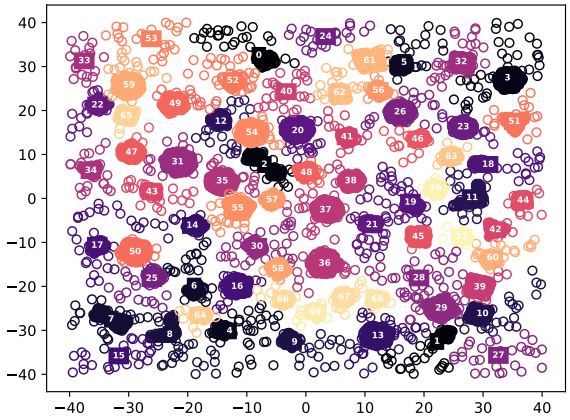

Fig. 21.

2) Maze: The table representing some of the trials is as follows:

clustersMaze.txt		
n_clusters	link_algo	silhouette_score
51	**ward**	**0.402**
51	complete	0.346
51	average	0.398
51	single	-0.71

The image representing these parameters is as follows:

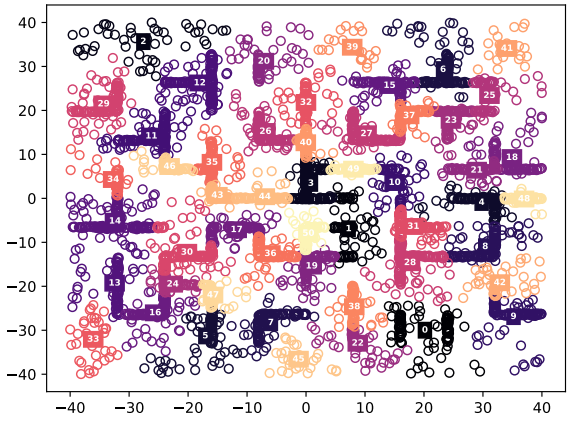

Fig. 22.

3) Objects: The table representing some of the trials is as follows:

clustersObjects.txt		
n_clusters	link_algo	silhouette_score
30	**ward**	**0.448**
30	complete	0.333
30	average	0.443
30	single	-0.632

The image representing these parameters is as follows:

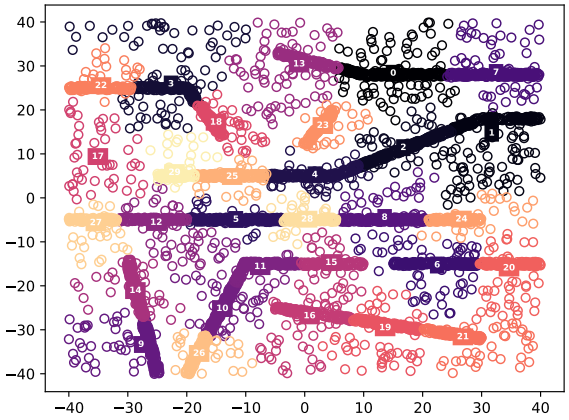

Fig. 23.

4) Rings: The table representing some of the trials is as follows:

clustersRings.txt		
n_clusters	link_algo	silhouette_score
70	**ward**	**0.464**
70	complete	0.378
70	average	0.444
70	single	-0.738

The image representing these parameters is as follows:

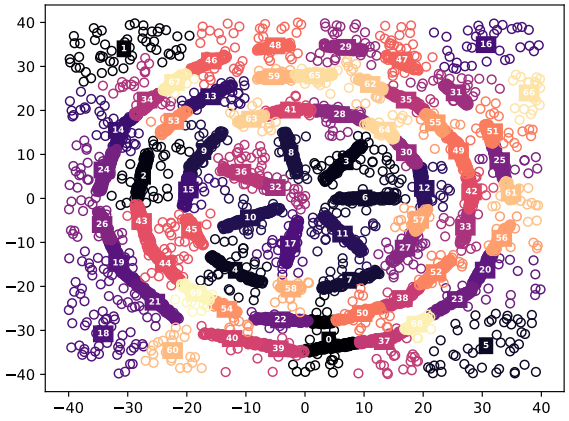

Fig. 24.

5) Waves: The table representing some of the trials is as follows:

clustersWaves.txt		
n_clusters	link_algo	silhouette_score
96	**ward**	**0.474**
96	complete	0.41
96	average	0.458
96	single	-0.71

The image representing these parameters is as follows:

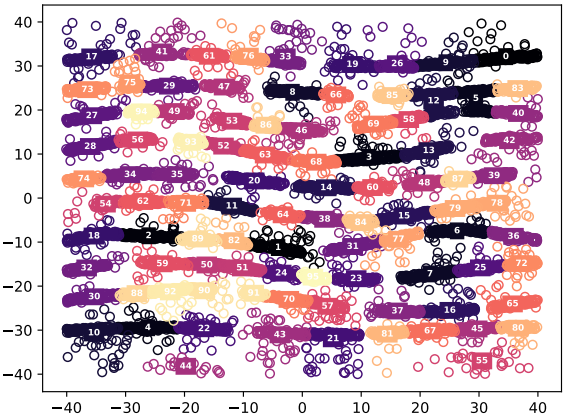

Fig. 25.

6) Analysis: From the tables, it can be observed that the ward linkage criterion performed the best for all of the clusters. Like the KMeans algorithms, this method is also affected heavily by noise, but the cluster centers are generally in the correct areas.

F. DBSCAN

The DBSCAN algorithm works by classifying points as core, border, or noise points. The main parameters to consider were epsilon and minimum sampling. Epsilon determines the maximum distance between two points to be considered within the neighborhood and minimum sampling determines the number of points to be considered a core point. Once again, euclidean distance was used because the data represents two dimensional points.

1) Dispersed: The table representing some of the trials is as follows:

clustersDispersed.txt		
eps	min_samp	silhouette_score
0.5	5	0.44
0.5	10	0.06
0.25	5	-0.5
0.4	5	0.06
0.5	6	0.424
0.5	**4**	**0.47**
0.5	3	0.438

The image representing these parameters is as follows:

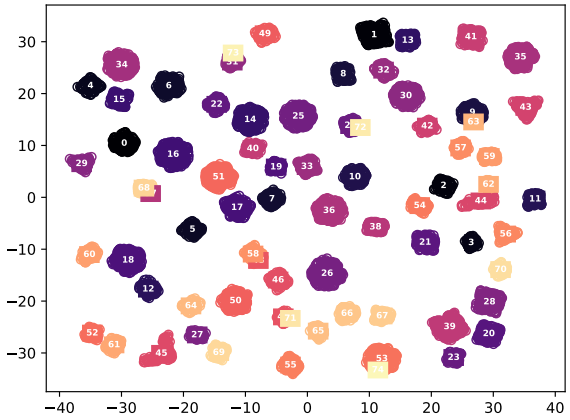

Fig. 26.

2) Maze: The table representing some of the trials is as follows:

clustersMaze.txt		
eps	min_samp	silhouette_score
0.5	5	0.112
0.5	3	0.097
0.5	4	0.109
0.3	5	-0.326
0.45	5	0.093
0.6	5	0.11
0.7	**5**	**0.06**
0.8	10	0.096

The image representing these parameters is as follows:

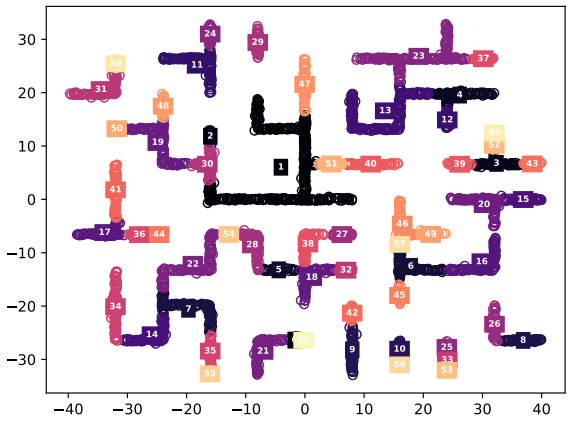

Fig. 27.

3) Objects: The table representing some of the trials is as follows:

clustersObjects.txt		
eps	min_samp	silhouette_score
0.5	5	0.096
0.6	5	0.108
0.7	**5**	**0.062**

The image representing these parameters is as follows:

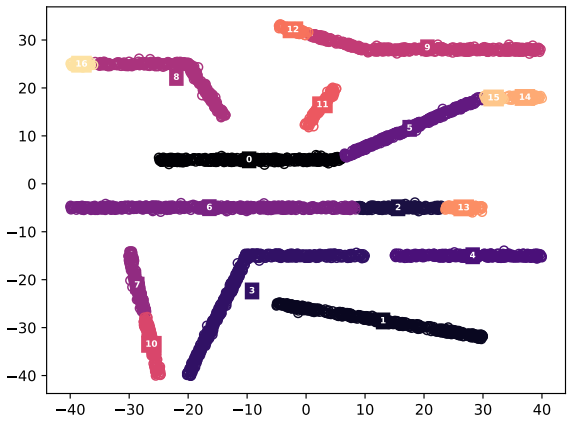

Fig. 28.

4) Rings: The table representing some of the trials is as follows:

clustersRings.txt		
eps	min_samp	silhouette_score
0.5	5	0.208
0.8	**5**	**-0.021**
0.8	4	-0.033

The image representing these parameters is as follows:

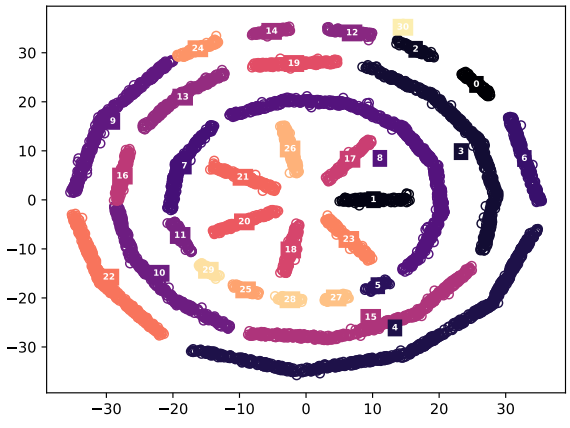

Fig. 29.

5) Waves: The table representing some of the trials is as follows:

clustersWaves.txt		
eps	min_samp	silhouette_score
0.5	5	0.251
0.6	5	0.125
0.7	**5**	**0.117**
0.75	5	0.119

The image representing these parameters is as follows:

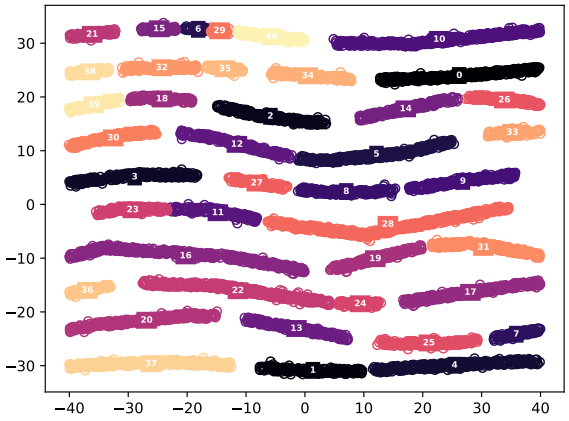

Fig. 30.

6) Analysis: As can be observed from some of the tables, the highest Silhouette Score was not necessarily chosen. When the plots were visually inspected, a higher Silhouette score usually had extra, unneeded clusters than a lower one. Instead, the best parameters were chosen using visual inspection. One thing to note is that DBSCAN handles noise a lot better than the previous algorithms. Additionally, it can identify the original trends using a lot less clusters.

G. Mini-Batch KMeans++

This is another approach that follows similarly to the KMeans algorithm, it uses smaller batch sizes of the data set to update clusters. The overlapping parameters that worked well for KMeans++ were chosen. The main focus for this algorithm was determining a good reassignment ratio.

1) Dispersed: The table representing some of the trials is as follows:

clustersDispersed.txt

n_clusters	ratio	silhouette_score
69	1.00E-02	0.604
69	1.00E-03	0.61
69	**1.00E-04**	**0.627**
69	1.00E-05	0.608
69	1.00E-01	0.58

The image representing these parameters is as follows:

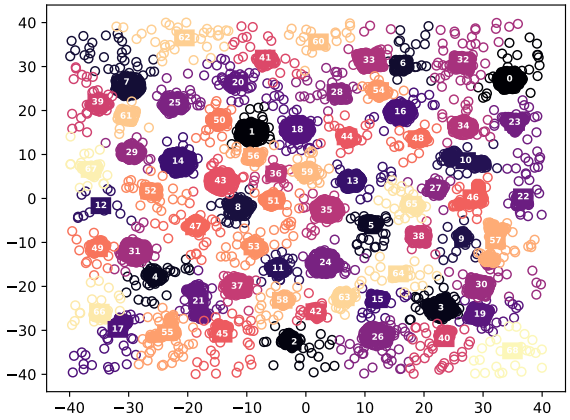

Fig. 31.

2) Maze: The table representing some of the trials is as follows:

clustersMaze.txt

n_clusters	ratio	silhouette_score
51	1.00E-01	0.428
51	1.00E-02	0.426
51	1.00E-03	0.427
51	1.00E-04	0.43
51	**1.00E-05**	**0.442**
51	1.00E-06	0.42

The image representing these parameters is as follows:

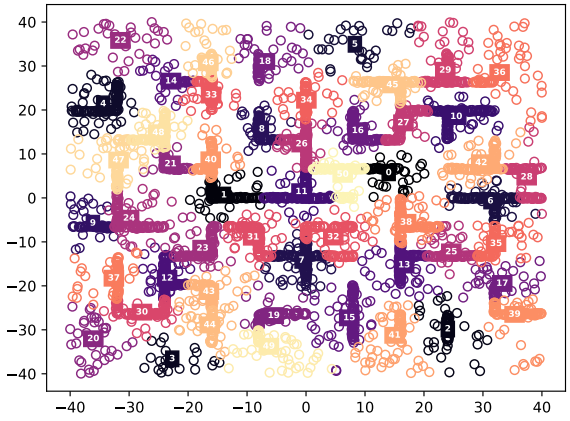

Fig. 32.

3) Objects: The table representing some of the trials is as follows:

clustersObjects.txt

n_clusters	ratio	silhouette_score
40	1.00E-01	0.46
40	**1.00E-02**	**0.474**
40	1.00E-03	0.462
40	1.00E-04	0.472
40	1.00E-05	0.456

The image representing these parameters is as follows:

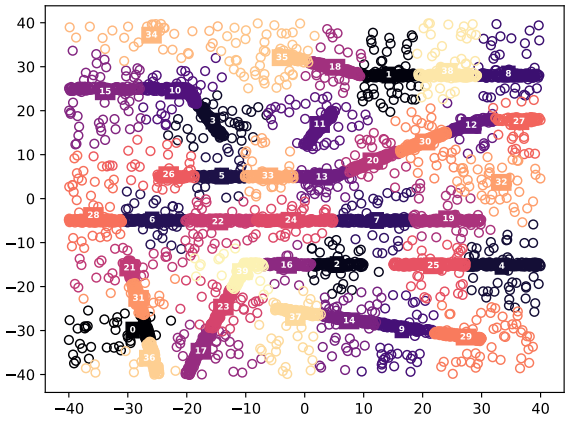

Fig. 33.

4) Rings: The table representing some of the trials is as follows:

clustersRings.txt

n_clusters	ratio	silhouette_score
70	1.00E-01	0.467
70	1.00E-02	0.474
70	1.00E-03	0.486
70	1.00E-04	0.482
70	**1.00E-05**	**0.483**
70	1.00E-06	0.481

The image representing these parameters is as follows:

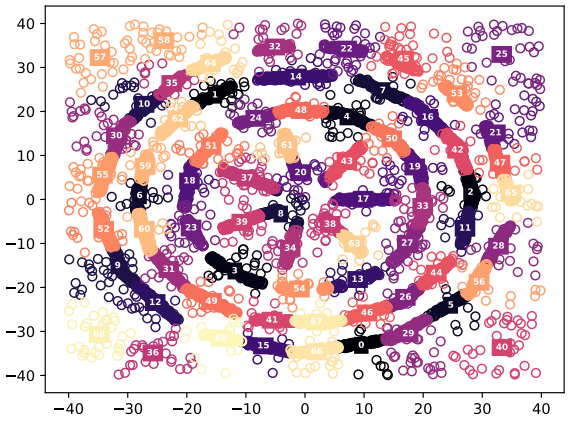

Fig. 34.

5) Waves: The table representing some of the trials is as follows:

clustersWaves.txt

n_clusters	ratio	silhouette_score
110	1.00E-01	0.49
110	**1.00E-02**	**0.496**
110	1.00E-03	0.491
110	1.00E-04	0.495
110	1.00E-05	0.492

The image representing these parameters is as follows:

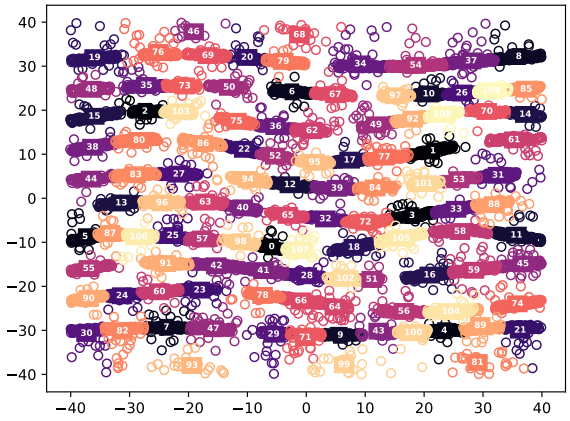

Fig. 35.

6) Analysis: Since the algorithm is similar to KMeans, it is also affected by noise. Disregarding the noise, the cluster centers moved generally well to where they were expected to.

III. Conclusion

After testing all of the algorithms over these data sets, it can be observed that DBSCAN performed the best – it handled the noise extremely well and did not require a large number of clusters to pick up on the shapes of the data. In the future, when using the other algorithms, noise should be filtered out first. With the noise gone, KMeans, KMeans++, Bisecting KMeans++, Agglomerative Clustering, and Mini-Batch KMeans++ would all drastically improve. Even with noise, the cluster centers were in the correct, general areas; without the noise, the cluster centers may converge faster and more accurately.