K-Means Algorithm For Clustering

Graham Burgsma

***Abstract*—The purpose of this paper is to analyze the k-means algorithm for clustering problems. The analysis will focus on testing the k-means algorithm on different data sets, k values and distance measures. As a benchmark for these tests, the clusters generated will be evaluated with the Dunn index and visually by graphing the clusters.**

Keywords*—*K-Means, Clustering, Dunn-index, Distance.

I. Introduction

K-means clustering partitions a set of points into k clusters. Clusters have a centroid which represents the center of the cluster. The k-means clustering algorithm is most efficient when using the optimal k value and distance measure for the data being tested. K-means clustering has practical applications like grouping similar colours or points that have common traits.

A. Data Sets

The data sets being used for this clustering are the s-sets. They are four data sets that increase in overlapping and difficulty. Each set has 5000 synthetic 2D vectors with 15 Gaussian clusters. Using the k-means clustering algorithm on these data sets, the optimal k value will be 15.

II. K-Means Algorithm

The k-means clustering algorithm works to partition a set of points into k clusters. This algorithm works by first randomly assigning k points from the data set to be the cluster centroids. Once assigned the algorithm assigns all points in the data set to a cluster. It does this by using some distance measure to find the closest cluster centroid to each point, the point is then assigned to the cluster of the nearest centroid. After all points are assigned to a cluster, the cluster centroid's position is recalculated. This is done by taking the average X and Y position of all points in the cluster. The action of assigning points to the nearest cluster and recalculating the centroids of each cluster is repeated until no points change clusters. The pseudo code for this algorithm is shown in Figure 1.

```
function kMeans(kValue)
    randomly generate K cluster centroids
    loop until no points change clusters
        call AssignPointsToNearestCluster
        recalculate centroids
    return dunnIndex
end

function AssignPointsToNearestCluster
    loop through points
        loop through centroids
            calculate distance
        assign point to closest centroid
end
```

Fig. 1: K-means Algorithm Pseudocode

A. Distance Measures

Three different distance measures will be used to test the k-means clustering algorithm. The three distance measures are Euclidean, Chebyshev and Manhattan. It will be tested how each of these distance measures impact the performance of the clustering.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Fig. 2: Euclidean Distance

$$max(|x_1 - x_2|, |y_1 - y_2|)$$

Fig. 3: Chebyshev Distance

$$|x_1 - x_2| + |y_1 - y_2|$$

Fig. 4: Manhattan Distance

B. Dunn Index

To get a benchmark for the performance of each run the Dunn index will be used. The Dunn index gives a numerical value for how well each run of the algorithm clusters the points. The Dunn index as shown in Figure 5 is a simple but expensive equation. The numerator of the equation is the smallest distance between two points of different clusters. The denominator is the largest distance between two object from the same cluster, also known as the diameter of a cluster. The Dunn index is the ratio between these two values and should give a numerical value for how well the clustering algorithm performed. The Dunn index should be maximized and is a relative value, so there is no set value of a good Dunn index.

$$Dunn = \frac{Intra_{min}}{Inter_{max}}$$

Fig. 5: Dunn Index

III. Results

To give an accurate comparison of the different parameters for the k-means clustering algorithm, it is important to have consistent results. For the purposes of comparison a static seed value has been used for all tests, this way the random values generated are the same for each run and so it is easy to compare the differences between k values or distance measures. For parameters tested only graphs are shown for data sets S1 and S4 to reduce the number of graphs, however all four data sets are included in analysis.

A. *K Values*

With k-means clustering, the k value represents the number of clusters that will be created. For the S data sets, the optimal k value is 15. To test how the k value affects the performance of the algorithm, a k value of 10, 15 and 20 will be tested for each data set. Graphs are shown of the runs for the S1 and S4 data sets and conclusions will be made from analyzing the results from all data sets. For this test, the Euclidean distance measure was used for all runs.

Figures 6, 7 and 8 show the results of using data set S1. Figures 9, 10 and 11 show the results of using data set S4. From figures 7 and 10 the k value is set to the optimal 15, from visual analysis of these graphs the cluster centroids are very close to each visible cluster of points. In these graphs, each centroid corresponds to each visible cluster with a 1:1 ratio. Looking at figures 6 and 9 where the k value is 10, this ratio is not preset. From visual analysis these graphs do not have a centroid for each visible cluster. There are few clusters in which the centroid corresponds with the most dense part of each cluster. For some other clusters the centroid is suspended between two clusters and is in a region with sparse points. This is as to be expected because there are not enough centroids to map 1:1 to the clusters in the data sets. Graphs 8 and 11 show the opposite and have a k value of 20, 5 higher than the optimal. With these graphs some centroids are placed in the middle of a dense set of points, but some visual clusters have two centroids because there are more k clusters than the optimal amount for these data sets.

The graphs are a clear visual representation of the performance of each clustering given different k values. Another measure of the performance is the Dunn index of each clustering. Table I shows the respective Dunn index values for each of the runs. All tests with $k = 15$ have the best Dunn index compared to the $k = 10$ and $k = 20$ runs for each data set. The exception to this result is the S3 data set in which the run with $k = 20$ was very slightly higher than the result from the $k = 15$ run. The Dunn index results confirm the visual analysis of the graphs: $k = 15$ provide the best clustering and any value higher or lower does not perform as well.

K	Dunn Index
S1 Data Set	
10	0.0016321123
15	0.0085988041
20	0.0007104440
S2 Data Set	
10	0.0019683217
15	0.0038161196
20	0.0023067369
S3 Data Set	
10	0.0017271050
15	0.0020163298
20	0.0021777280
S4 Data Set	
10	0.0023854194
15	0.0026832598
20	0.0023571549

TABLE I: Dunn values for Variable K

B. *Distance Measures*

With k-means clustering, the distance measure is an important factor of how well the algorithm performs. For testing of the distance measure the following three distance measures were used: Euclidean, Chebyshev and Manhattan. All the runs used k as 15 but the distance algorithm was changed each run and tested on all four of the S data sets.

Figures 12 through 17 show the results as a graph for S1 and S4 data sets. Visually analyzing the graphs of the S1 data (Figures 12 - 14) shows Euclidean and Manhattan as nearly identical placement of centroids in the optimal locations. Chebyshev distance resulted in a few centroids being misplaced. Analysis of S4, which is a more difficult data set to correctly cluster, shows little difference between all three distance measures. Visual analysis is difficult to determine the performance of each distance measure.

Table II shows the results from each data while testing each distance measure. The Dunn index shows Euclidean distance performed the best for all four data. Chebyshev distance performed the second best for data sets S2 and S3. Manhattan distance had the second best Dunn index for S1 and S4. From analysis of the Dunn index, Euclidean is clearly the best however Chebyshev and Manhattan show mixed results for each of the data sets.

Distance	Dunn Index
S1 Data Set	
Euclidean	0.0085988041
Chebyshev	0.0010041226
Manhattan	0.0040854322
S2 Data Set	
Euclidean	0.0038161196
Chebyshev	0.0027384861
Manhattan	0.0012718491
S3 Data Set	
Euclidean	0.0020163298
Chebyshev	0.0017688764
Manhattan	0.0012455691
S4 Data Set	
Euclidean	0.0026832598
Chebyshev	0.0010861724
Manhattan	0.0022849825

TABLE II: Dunn values for Distance Measures

C. Dunn Index

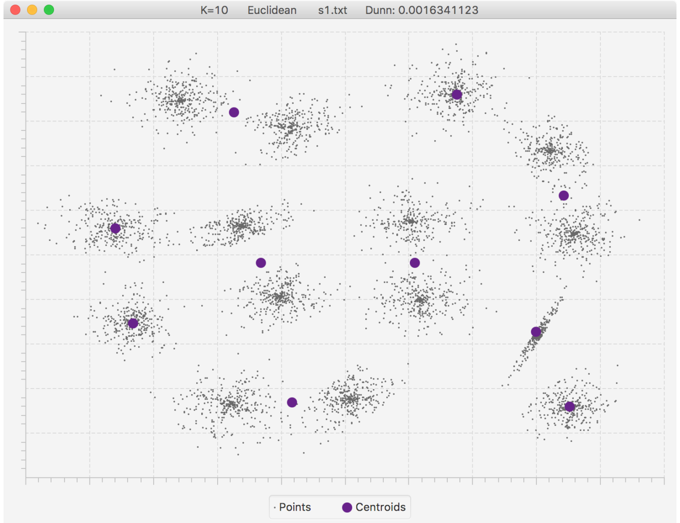
The Dunn index results as found in Tables I and II are found to be good measures of the performance for the k-means clustering algorithm. When testing different k values, the Dunn index verified the visual analysis of the graphs. With testing the different distance measures, visual analysis was difficult to measure the performance of the clusters. In this case the Dunn index provided a numerical value that could evaluate the performance of the clustering and was easy to analyze.

IV. Conclusion

In conclusion it is evident k values are crucial in the performance of k-mean clustering algorithms. Since choosing the k value is done by user input, the k-means clustering algorithm is limited. This makes the algorithm less suitable for various data sets unless the optimal k value is given. Too small of a k value and clusters will be too broad and too large of a k value will result in too specific of clustering.

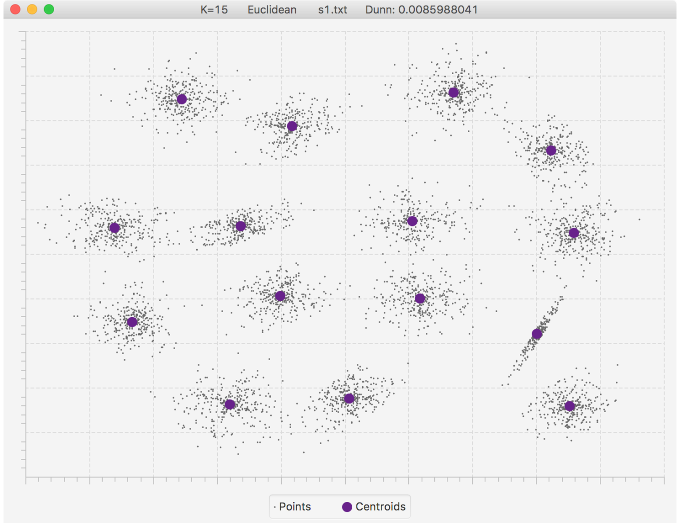
There are many different distance measures available. Distance measures play a large role in the performance of the k-means clustering algorithm and differ depends on the data. For the S data sets, Euclidean distance performed the best. Chebyshev and Manhattan gave inconsistent results.

For the S data sets, $k = 15$ using Euclidean distance performed the best. Visual analysis of the points and clusters works well for evaluating the performance when there are large differences. Dunn index is a good performance measure and can evaluate clusters that look similar in visual analysis.



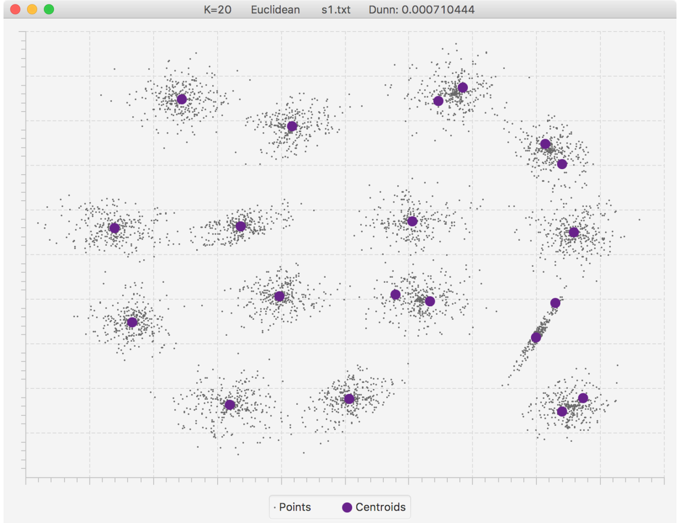
K=10 Euclidean s1.txt Dunn: 0.0016341123

Fig. 6: Euclidean K=10 S1 Data



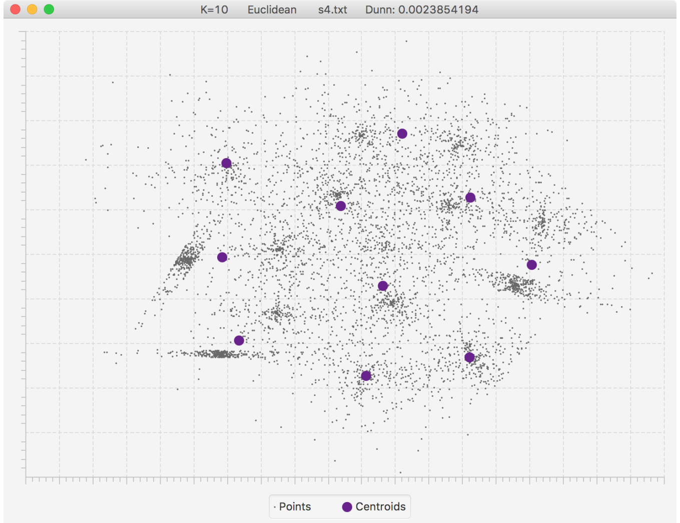
K=15 Euclidean s1.txt Dunn: 0.0085988041

Fig. 7: Euclidean K=15 S1 Data



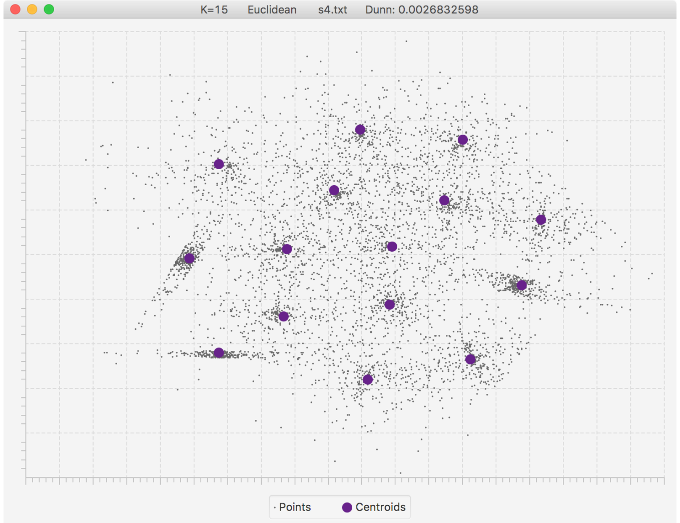
K=20 Euclidean s1.txt Dunn: 0.000710444

Fig. 8: Euclidean K=20 S1 Data



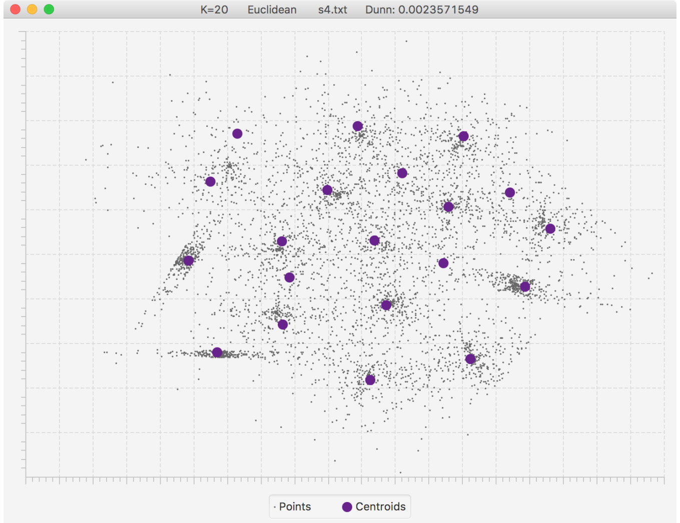
K=10 Euclidean s4.txt Dunn: 0.0023854194

Fig. 9: Euclidean K=10 S4 Data



K=15 Euclidean s4.txt Dunn: 0.0026832598

Fig. 10: Euclidean K=15 S4 Data



K=20 Euclidean s4.txt Dunn: 0.0023571549

Fig. 11: Euclidean K=20 S4 Data

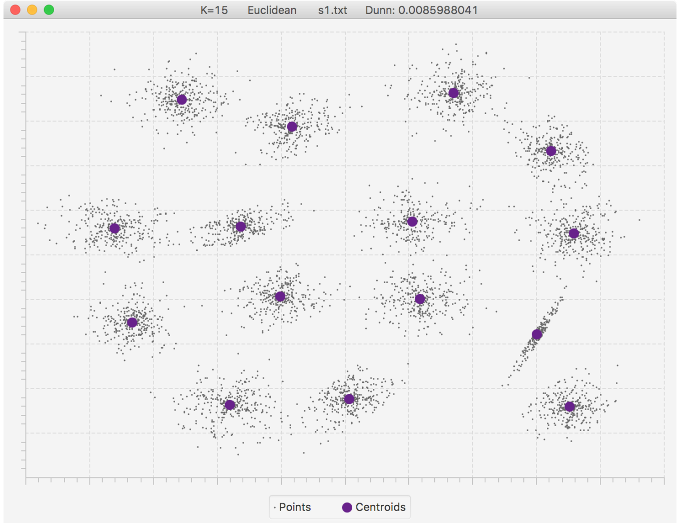

K=15 Euclidean s1.txt Dunn: 0.0085988041

Fig. 12: Euclidean K=15 S1 Data

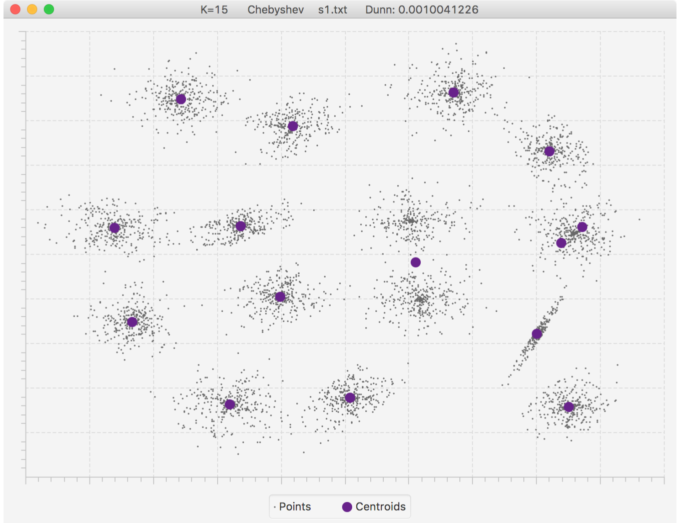

K=15 Chebyshev s1.txt Dunn: 0.0010041226

Fig. 13: Chebyshev K=15 S1 Data

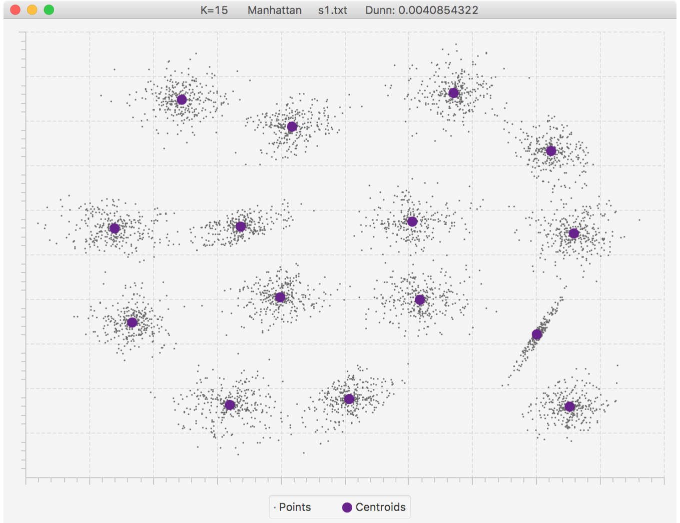

K=15 Manhattan s1.txt Dunn: 0.0040854322

Fig. 14: Manhattan K=15 S1 Data

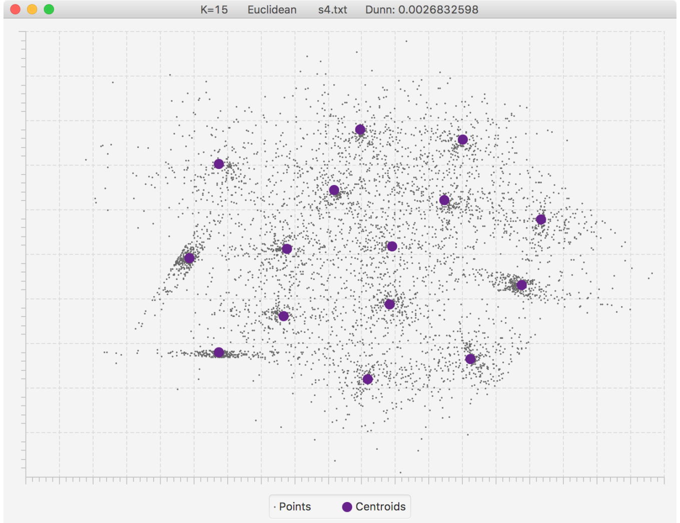

K=15 Euclidean s4.txt Dunn: 0.0026832598

Fig. 15: Euclidean K=15 S4 Data

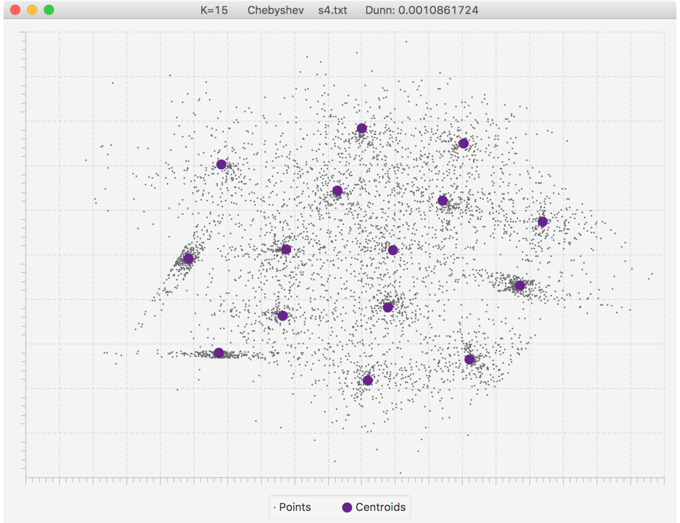


Fig. 16: Chebyshev K=15 S4 Data

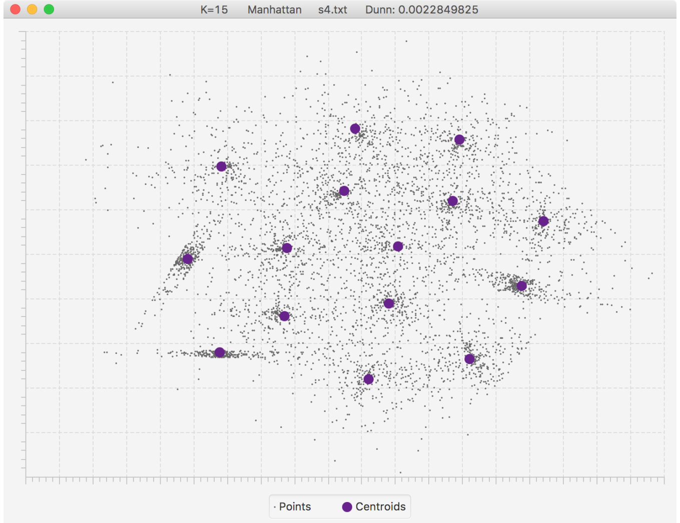


Fig. 17: Manhattan K=15 S4 Data