Skyler Gray

Clustering Report

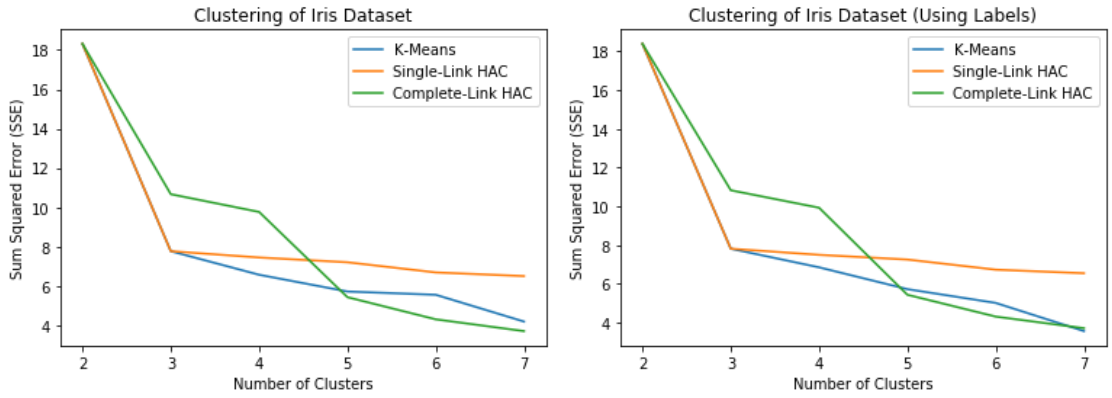
The following report summarizes my implementations and research regarding the k-means and both single- and complete-link HAC algorithms.

1. Implement the Algorithms

I fit several models (k-means and single- and complete-link HAC) on the dataset in file "abalone.arff" using the debug settings; their clustering performance is comparable to the scikit-learn module's algorithm. I then fit the data from "seismic-bumps_train.arff", including the clustering results of the three algorithms in this directory.

2. Iris Dataset

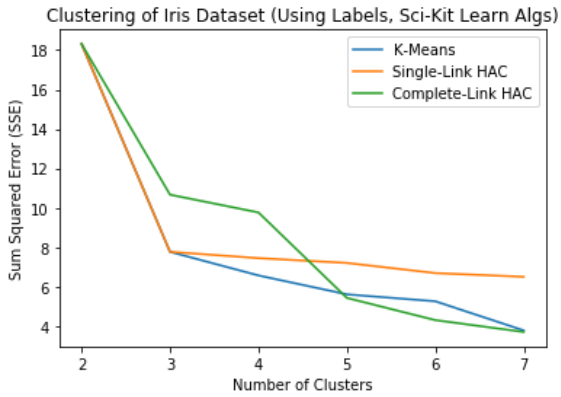
After debugging these several algorithms, I tested their performance on clustering the iris dataset across numbers of clusters. In the plots below, the data was normalized before fitting. I fit the models to the data first excluding and lastly using labels. Clustering SSE performance is similar regardless of whether labels are part of the fit data.



The last analysis I ran with the iris dataset involved running the k-means algorithm on the dataset five times. Each time, I used four clusters and initialized the centroids randomly. I meant to investigate the variation in clustering performance. Consistently, the centroids settled such that two among them contained fifty observations each. Between any two k-means fittings, they shared at least one of those major centroids. Overall, though, the collective SSE was different at each run by between 0.001 and 0.5.

3. Sci-Kit Learn

To compare the performance of Python's sci-kit learn module algorithms with my own implementation, I fit its k-means as well as single- and complete-link HAC models to the iris dataset. All three algorithms look practically the same in SSE performance compared to my own algorithms.

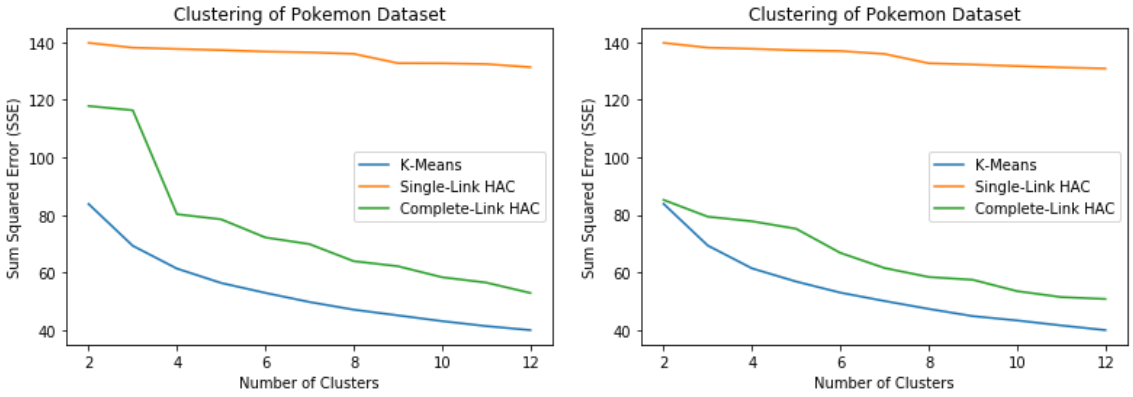


4. Custom Dataset: Pokémon

Below is an attempt at clustering Pokémon based on their basic stats. The Pokémon types were removed to focus the clustering on the Pokémon fighting stats.

With the default values, it is interesting to note that K-means and single-link HAC perform significantly better in reducing SSE than complete-link HAC. As expected, increasing the number of clusters leads the SSE of every method to decrease.

I expected to see a change in K-means' performance when I tweaked several hyper-parameters (10 iterations per value of k; initializing centroids at completely random--not necessarily efficient--spots; and only stopping when the centroids do not change between iterations). Unfortunately, none of these changes significantly adjusted the shape and values of this curve across values of k. When changing the distance measurement for complete-link, though, its SSE performance more closely matched that of the K-means curve. The same change hardly altered the performance of single-link HAC.



In my opinion, none of the clusterings stand out as the dominant clustering when measuring their SSE performance, because SSE biased toward cluster compactness and does not account for cluster separability. Silhouette scoring is preferable in this instance of choosing the best clustering method as well as the best number of clusters for the data because it accounts for both separability as well as compactness.

I reran my Pokémon code, this time recording and plotting the silhouette score of each algorithm across the number of clusters. It is more obvious now that clustering into two groups using single-link HAC with 2 clusters results in a better combination of separability and compactness of clusters.