Analysis of Various Clustering Algorithms

sduri2, kasivis2, jtrichm2, bommine2

Introduction

When dealing with unstructured data sets, we often seek to find patterns of similarities within the dataset. While these similarities may not always be significant or meaningful, such patterns may sometimes reveal common groupings which are in fact significant; for example, we might notice that certain gene sequences correspond to a high chance of being diagnosed with certain medical conditions.

There are various methods by which we can try to find these patterns in unstructured data. Clustering is one common approach, where we seek simply to group together similar instances of data, for various definitions of “similar”. As we vary what we mean by “similar” – for example, we could define cluster similarity as the Euclidean or squared Euclidean distance between clusters’ center points – we are left with various algorithms and methods to cluster or group data points with each other. There are many algorithms for clustering including partitional methods such as k-means, minibatch k-means and Gaussian mixture models; and hierarchical methods such as agglomerative clustering, and so on. Each algorithm performs better or worse on certain types of data sets, and with certain types of data (for example, low vs. higher dimensional data).

As we become more capable of collecting large data sets of gene sequences and other biological data, the task of finding common groupings becomes harder and harder. One of the goals of research is identifying various new means of defining “similarity” as well as identifying the types of data sets a particular algorithm performs well (or poorly on), and vice-versa.

Our project seeks to compare various algorithms for DNA clustering on micro-arrays. We will compare the performances of the algorithms listed below on the same datasets and seek to determine the implications of these algorithms' performances.

The algorithms we compared were the following:

- K-means
- Ward Agglomerative Clustering
- Average Agglomerative Clustering
- Complete Agglomerative Clustering
- Spectral Clustering
- HDBScan

Methods

We used two data sets for our evaluations: "The human whole miRNOme project version 1 (blood)" dataset, obtained here, and the "Mice Protein Expression Data Set" obtained from here. The former is a dataset of the blood expression profiles of 863 miRNA for 454 individuals; it originates from a study which sought to analyze this data, comprised of individuals diagnosed with different human diseases to test for disease-specific alterations. During preprocessing, we removed data columns which displayed low variance in order to increase the relevance of the data in the dataset for our clustering algorithms. The latter is a dataset of 77 proteins measured in the cerebral cortex of 8 classes of Down syndrome mice. Overall, we used these datasets as common datasets with which to compare clustering outputs between various algorithms. For each of the algorithms listed above we ran the clustering methods on the data sets and compared the outputs, both visually and quantitatively. Quantitatively, we measured the silhouette and Calinski-Harabasz scores as well as the runtime performance of these algorithms as measured by runtime. We used the former as a measure of the effectiveness of each algorithm/method, and the latter as a measure of the cost of each algorithm.

Results & Discussion

Mice Protein Data

Algorithm/Method	Silhouette Score	Calinski-Harabasz	Runtime (sec.)
K-means	0.1990	248.6167	0.1378 s
Ward Agglomerative Clustering	0.1321	211.7696	0.0594 s
Average Agglomerative Clustering	0.1638	102.9129	0.0556 s
Complete Agglomerative Clustering	0.1624	118.5988	0.0553 s
Spectral Clustering	0.1306	123.6942	0.1816 s

The Mice Protein data is a dataset of 77 proteins measured in the cerebral cortex of 8 classes of Down syndrome mice. The eight classes of mice come about based on varying features such as genotype, behavior and treatment applied. We see the results of applying the clustering algorithms to this dataset above. For each algorithm, the Silhouette Score and Calinski-Harabasz Scores and the average runtime in seconds over 100 repetitions of each algorithm were calculated. Silhouette Scores near 0 indicate the clusters overlapped each other, with values closer to 1 representing better scores. We see here that none of the algorithms had particularly high Silhouette scores, which suggests that the dataset did not lend itself well to being clustered. We see that K-means performs the best on this dataset, with the highest Silhouette Score of the lot.

Calinski-Harabasz scores represent the ratio between within-cluster and between-cluster dispersions, with higher values being better. Interestingly, we see that Ward Agglomerative Clustering and K-Means have similar Calinski-Harabasz scores, better than the other algorithms, but the former runs about 2.5 times faster than the latter. Between the agglomerative algorithms, we see that Ward clustering performs better than the others in terms of the Calinski-Harabasz scores, with similar runtimes for all of these agglomerative algorithms. Overall, however, all of the algorithms performed relatively poorly on the Mice Protein dataset, with low Silhouette scores.

We see visual representations of these algorithms’ clusters in the plots below. Note how the Average Agglomerative Clustering, which had the worst Calinski-Harabasz score, has poor clustering overall.

Agglomerative Clustering Methods

K-Means

Spectral Clustering

Blood Dataset

Algorithm/Method	**Silhouette Score**	**Calinski-Harabasz**	**Runtime (sec.)**
K-means	0.2605	174.7596	0.0541
Ward Agglomerative Clustering			
Average Agglomerative Clustering			
Complete Agglomerative Clustering			
Spectral Clustering	0.5517	383.5449	0.0511
HDBSCAN	0.1369	21.5085	0.0160

Agglomerative Clustering Methods

K-Means

Spectral Clustering

Conclusion