Comparison of recent clustering algorithms

Antoni Skubisz [1]

[1]Lancaster University

Motivation

Every clustering algorithm has its weaknesses and there is no one-size-fits-all algorithm. The choice of appropriate method should be made based on the specific characteristics of the data. Examples of differences between two most popular algorithms, k-means and DBSCAN, are shown in Figure 1

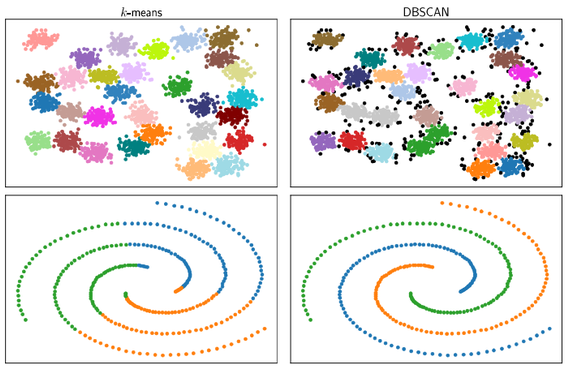

Figure 1. Cluster allocation obtained by k-means and DBSCAN on two datasets that exhibit different characteristics.

- k-means performs better with spherical clusters and struggles to detect "natural groupings" when clusters have non-standard shapes.
- DBSCAN is able to find arbitrarily shaped clusters, however, tends to merge together clusters that are overlapping.

Background

The differences emerge from the way algorithms conceptualise a notion of cluster. We can observe the following approaches to building clusters:

- **Centroid based models** — each cluster has a single vector known as the centroid, which serves as the most representative point within that cluster.
- **Density based models** — defines clusters as dense regions separated by sparser ones.
- **Hierarchical models** — provides clustering as a set of nested relationships, where two most similar groups are combined at each step.
- **Graph based models** — clusters are a set of connected components achieved by finding an optimal cut through graph representation of the data.

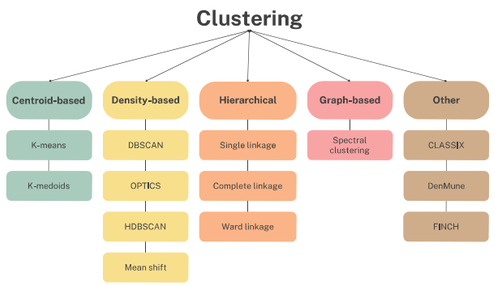

Figure 2. List of considered algorithms.

Benchmarking

Clustering algorithms will be evaluated based on the quality of clusters as well as the time needed to achieve them. Our approach to performing benchmarking is:

- 35 synthetic datasets that introduce different types of difficulties and 8 real-world classification datasets.
- 3 internal validation metrics (Calinski-Harabasz, Silhouette score, DBCV)
- 2 external validation metrics (Adjusted Rand Index, Adjusted Mutual Information)

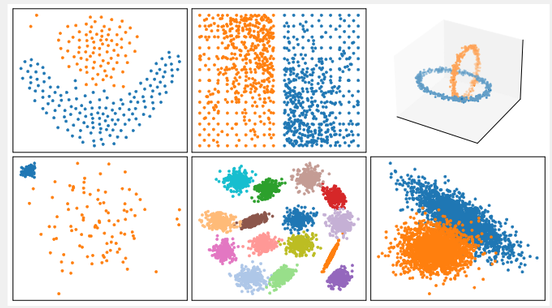

Figure 3. Sample of datasets used for benchmarking.

Results of quality comparison

Based on the results of our comparison we can form these key observations:

- k-means and DBSCAN complement each other. When one didn't perform well, the other proved to be effective.
- Algorithm like HDBSCAN, which offers easier parameter tuning than its predecessor, also restricts the possible range of solutions, which can affect the quality of clusters.
- Recent approaches such as spectral clustering, CLASSIX or DenMune showed greater flexibility in being able to solve a wide range of problems.

In Figure 4 we visualise the Adjusted Rand index achieved by algorithms on a sample of considered datasets.

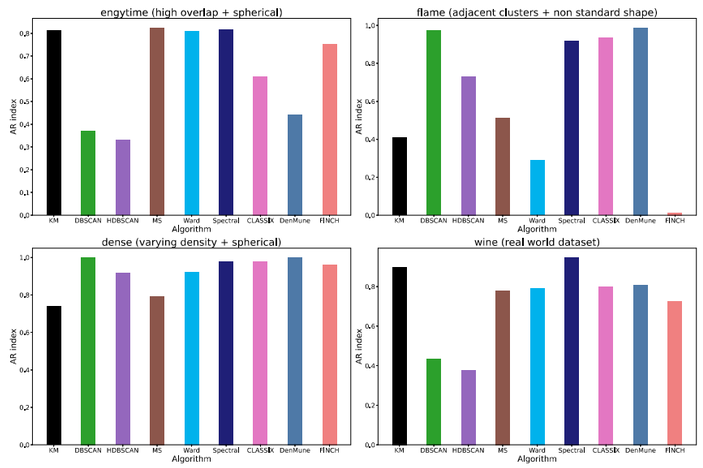

Figure 4. Adjusted Rand score achieved on datasets with different types of difficulties.

Results of scalability comparison

As for the comparison of scalability, we considered scalability with respect to sample size, dimensionality and number of clusters. We fixed two parameters and increased the third one to notice the trajectory of a fitted line. The results for sample size are shown in Figure 5

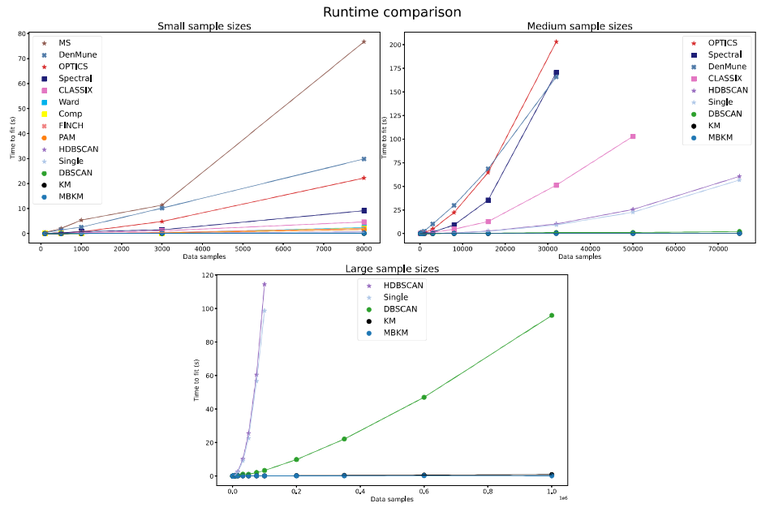

Figure 5. Scalability of algorithms with respect to sample size.

The main points with respect to scalability are:

- k-means is a clear winner and can scale to massive datasets.
- Many algorithms had to be removed early on due to memory constraints.
- As for dimension scalability, spectral clustering seemed to be insensitive, on the considered range of values. Meanwhile, most algorithms showed a linear growing trend.
- Increasing number of centroids led to a massive increase in runtime for spectral clustering and linear growth for k-means. The rest of the algorithms were rather unaffected.

Take-aways

Context	Algorithm
Large dataset	k-means
High dimensions	Spectral clustering, projection into lower dimensions + k-means
Non-convex clusters	DBSCAN (and variations), spectral clustering, single linkage, CLASSIX, DenMune
Noise in the data	DBSCAN (and variations), mean shift, CLASSIX, DenMune
Outliers in the data	DBSCAN (and variations), mean shift, CLASSIX, DenMune, PAM
Predefined number of clusters	k-means, hierarchical clustering, spectral clustering
Explainable clusters	k-means, hierarchical clustering, CLASSIX
Flexible distance metric	DBSCAN (and variations), hierarchical clustering (except Ward linkage), FINCH, PAM
Varying density clusters	OPTICS (with ξ method), HDBSCAN, spectral clustering, DenMune, CLASSIX
Highly overlapping clusters	k-means, mean shift, spectral clustering, Ward/complete linkage, FINCH

Figure 6. Examples of situations where certain clustering algorithms are applicable.