Section 8: Comparison of Algorithms

Section 8.1: Metrics for Measuring Quality of Clustering

Metrics for Measuring Quality of Clustering

Metrics:

- Silhouette Index
- Davies-Bouldin Index
- Dunn Index
- Examples of Internal Evaluation metrics (based on cluster results only and not any predefined labels)

Silhouette Index

- Silhouette Index $s(X_i)$ measures the similarity of point X_i to other points its own cluster compared to other clusters
- $s(X_i)$ ranges from -1 to 1
- The average of the silhouette values over all data points is silhouette index for entire dataset
- Here similar and different are quantified by a distance measure

Silhouette Index

- Silhouette index for a cluster of 1 point is 0
- For X_i in cluster C_i with more than 1 point, define (average distance to points within cluster):

$$a(X_i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i} dist(X_i, X_j)$$

where $|C_i|$ is the number of points in the cluster

- Define (min average distance to points in another cluster)

$$b(X_i) = min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} dist(X_i, X_j)$$

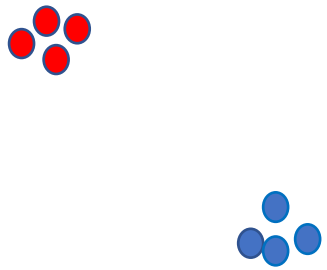
- $b(X_i)$ is min over other clusters of average distance between X_i and points in that cluster
- Define silhouette index:

$$s(X_i) = \frac{b(X_i) - a(X_i)}{max(a(X_i), b(X_i))}$$

Silhouette Index: Examples

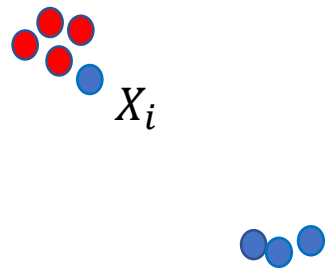
Example 1: Well separated clusters

- For each point:
  - $a(X_i) \approx 0 \quad b(X_i) \gg a(X_i)$
  - $s(X_i) = \frac{b(X_i) - a(X_i)}{max(a(X_i), b(X_i))} \approx 1$



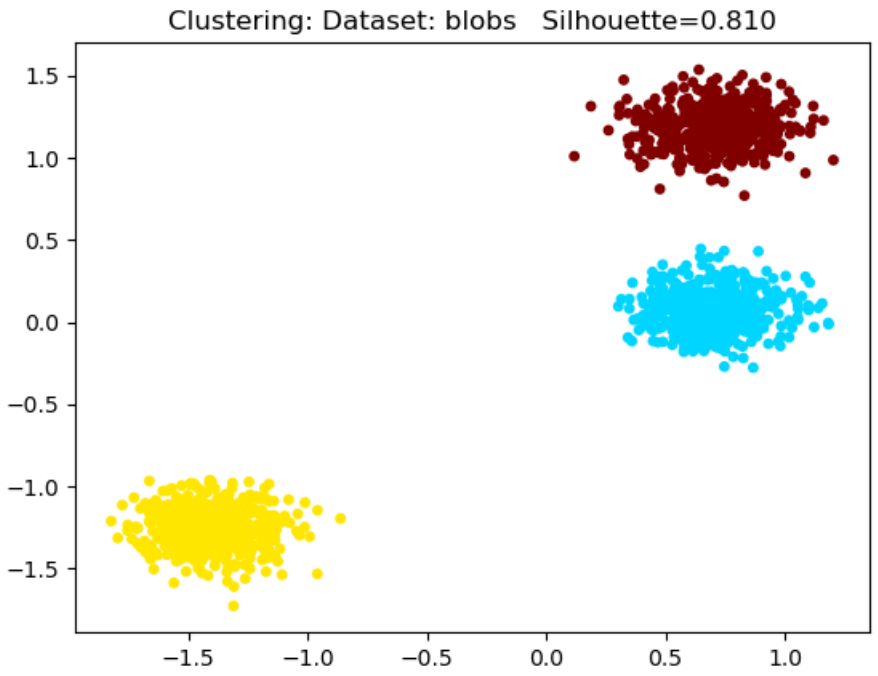
Example 2: Clusters close to each other

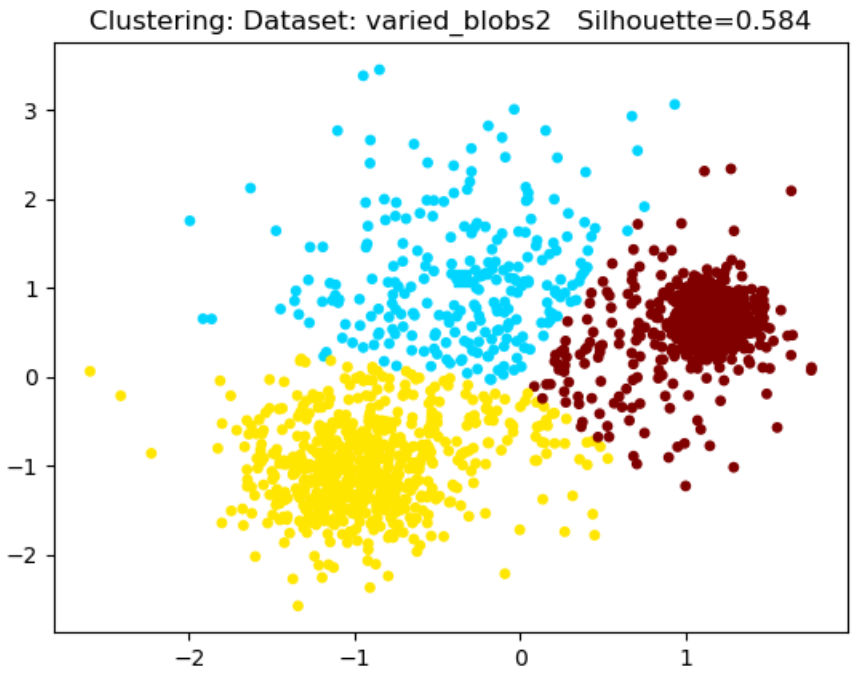
- For blue point X_i:
  - $a(X_i) > 0 \quad b(X_i) \ll a(X_i)$
  - $s(X_i) = \frac{b(X_i) - a(X_i)}{max(a(X_i), b(X_i))} \approx -1$



Silhouette Index: Example

- Dataset: 1500 points in “blobs” and “varied_blobs2” datasets
- K Means with 3 clusters





Section 8.2: Comparison of Algorithms

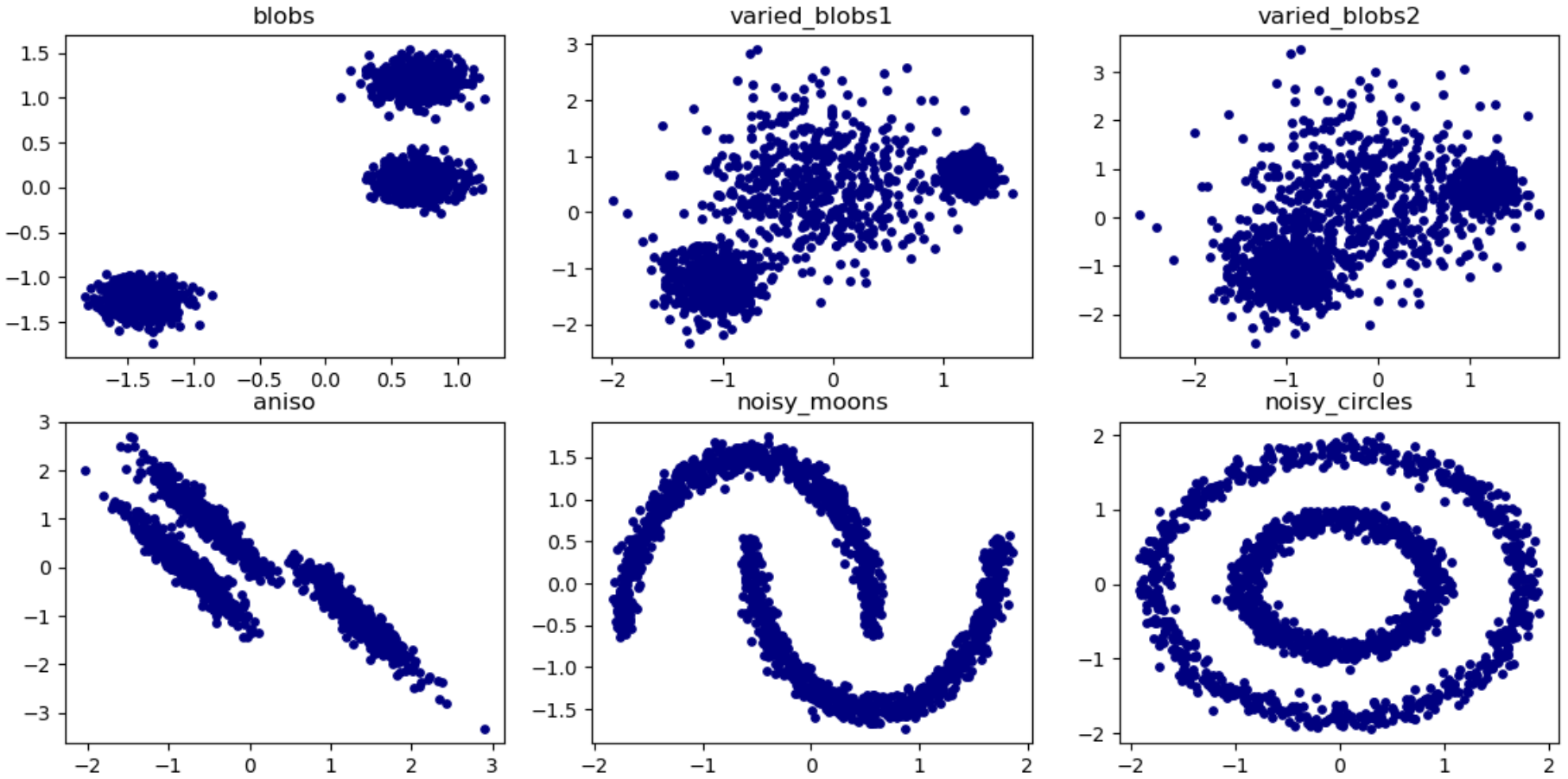
Comparison of Clustering Algorithms

- Compare clustering using K Means, Gaussian Mixture Model and DBSCAN for various datasets
- Will not use Hierarchical Clustering since it is a impractical choice if there are a large number of data points
- Similar to what is done in sklearn

https://scikit-learn.org/stable/modules/clustering.html

Comparisons of Algorithms: Datasets

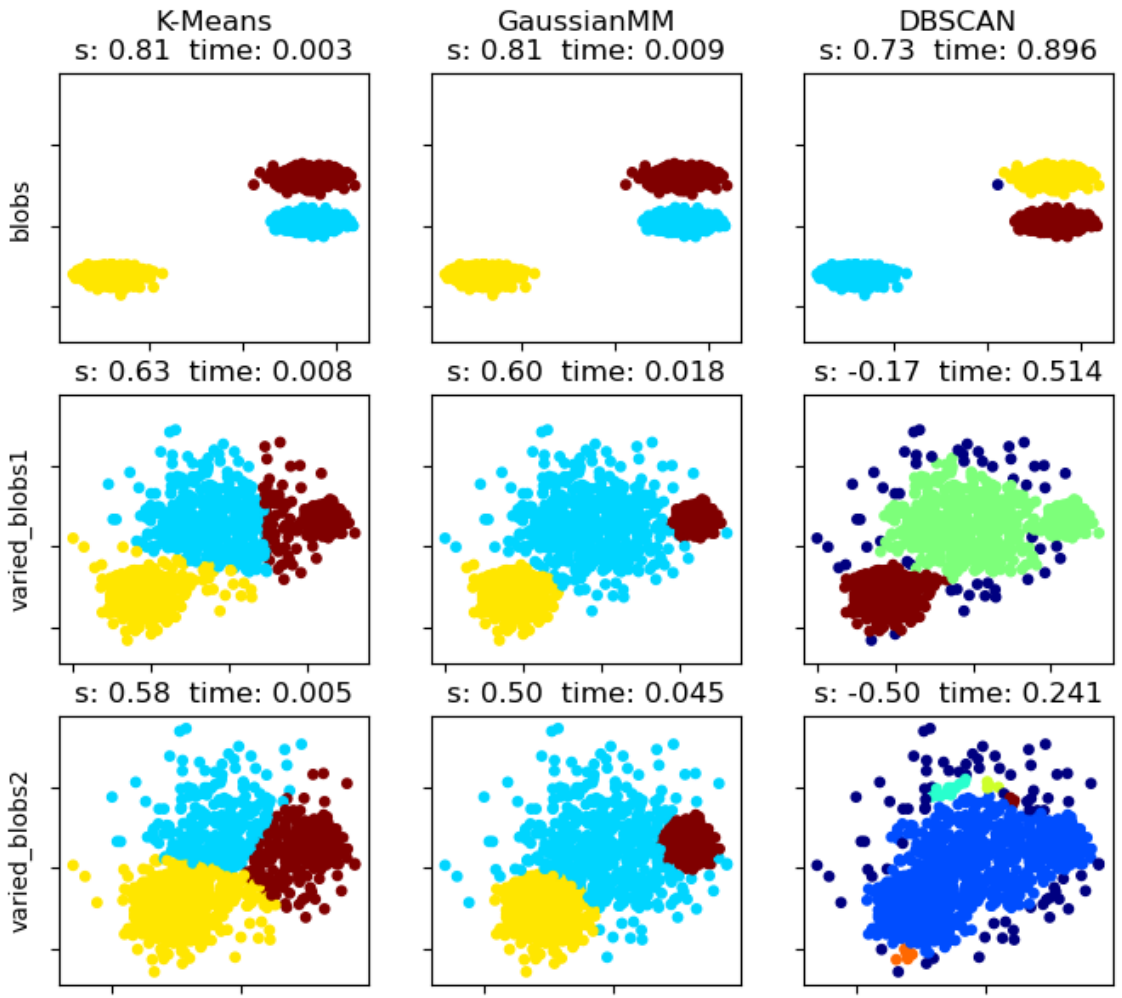
- sklearn datasets using 1500 data points



Comparison of Algorithms: Settings

Dataset/Algorithm	DBSCAN	K Means	Gaussian Mixture Model
blobs	minpts = 5, epsilon = 0.18	3 clusters, kmeans++	3 clusters, kmeans++
varied_blobs1	minpts = 5, epsilon = 0.18	3 clusters, kmeans++	3 clusters, kmeans++
varied_blobs2	minpts = 5, epsilon = 0.18	3 clusters, kmeans++	3 clusters, kmeans++
aniso	minpts = 5, epsilon = 0.18	3 clusters, kmeans++	3 clusters, kmeans++
noisy_moons	minpts = 5, epsilon = 0.18	2 clusters, kmeans++	2 clusters, kmeans++
noisy_circles	minpts = 5, epsilon = 0.18	2 clusters, kmeans++	2 clusters, kmeans++

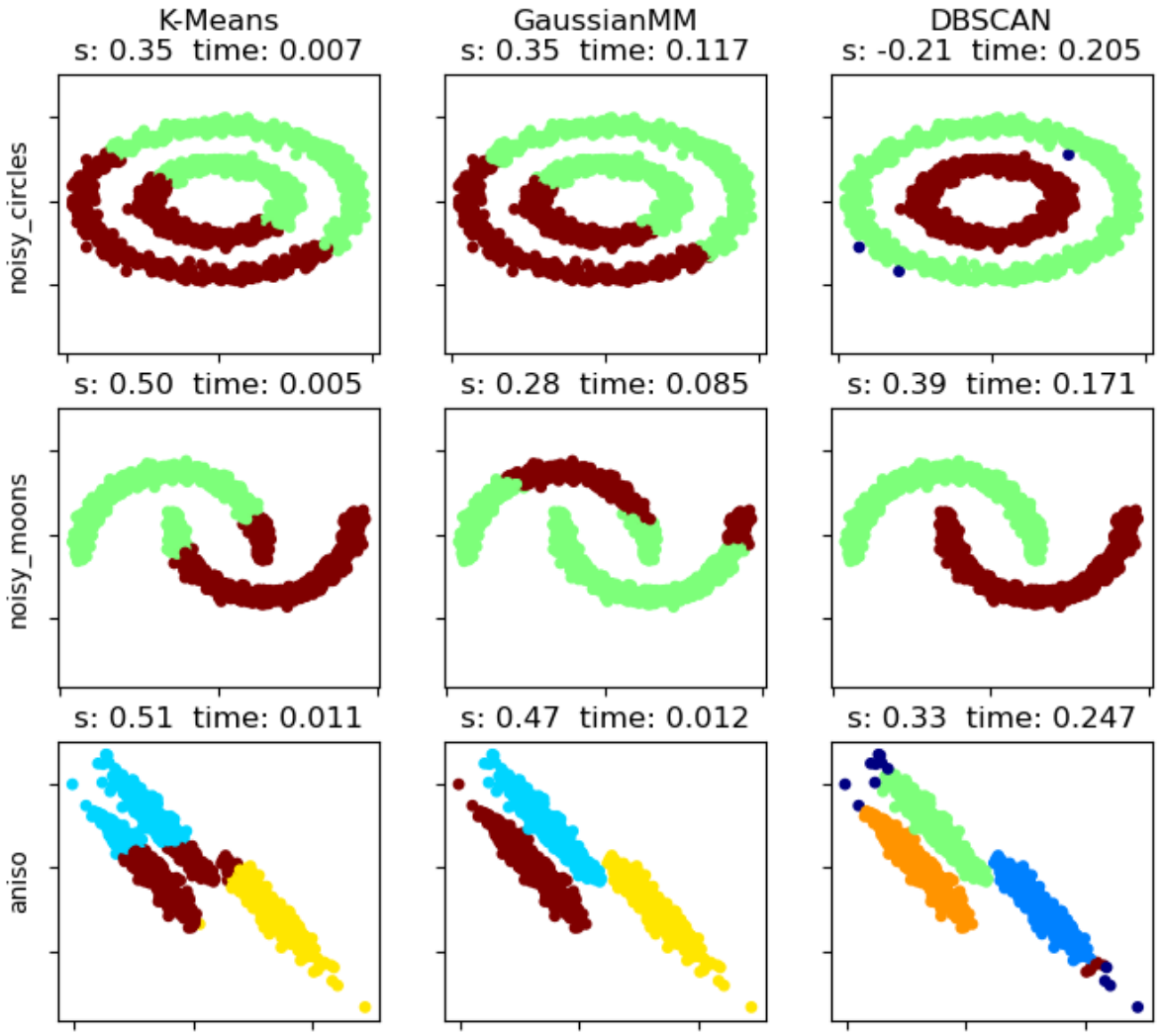
Comparisons of Algorithms: Results1



Notes:

- K Means and GaussianMM:
  - Perform similarly
  - K Means faster than GMM
- DBSCAN: impacted by minpts and epsilon:
  - If density too low: then all points belong to a single cluster
  - If density too high: then lots of clusters with single points
  - Does not do well with clusters of varying density
  - DB Scan much slower than K Means and GMM

Comparisons of Algorithms: Results2



- K Means:
  - Does not work well for non-convex regions (circles or moons)
  - Does not work well for elongated regions (aniso)
- GMM:
  - Does not work well for non-convex regions (circles or moons)
  - Can handle elongated regions
- DBSCAN:
  - Can handle non-convex regions