Understanding Clustering
Supervising the unsupervised

Janu Verma

IBM T.J. Watson Research Center, New York

http://jverma.github.io/

jverma@us.ibm.com

@januverma

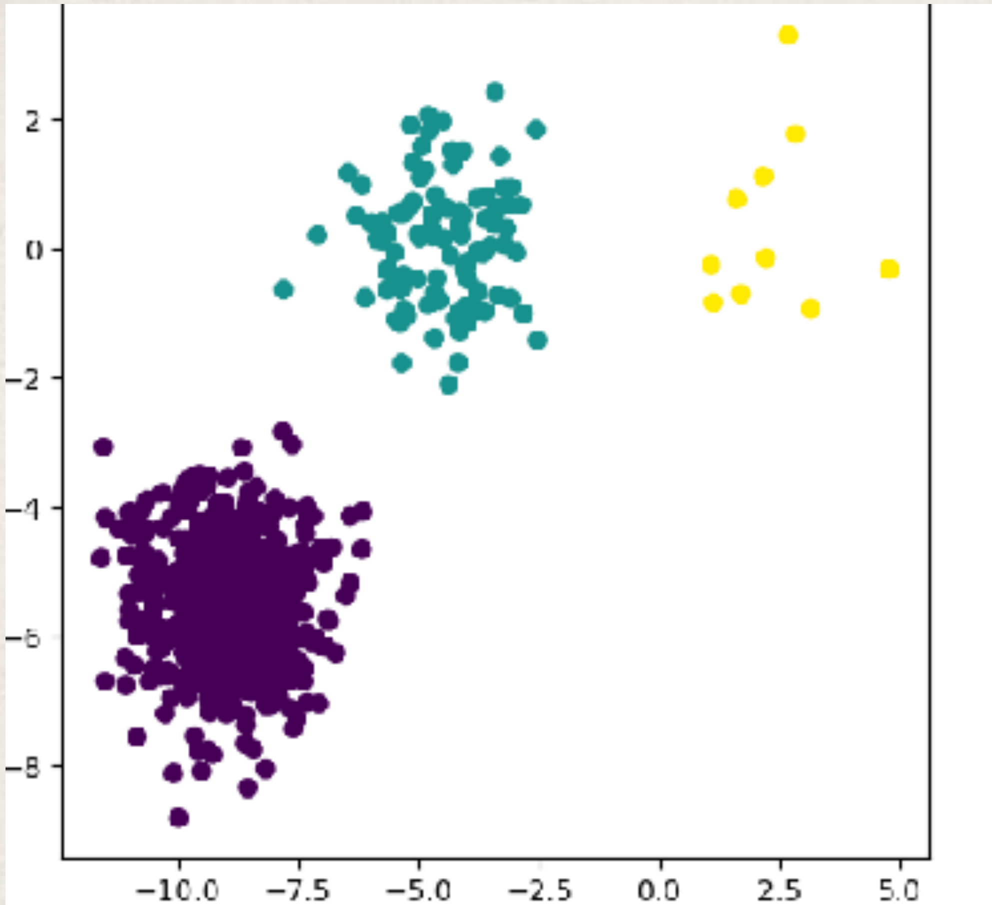
Clustering

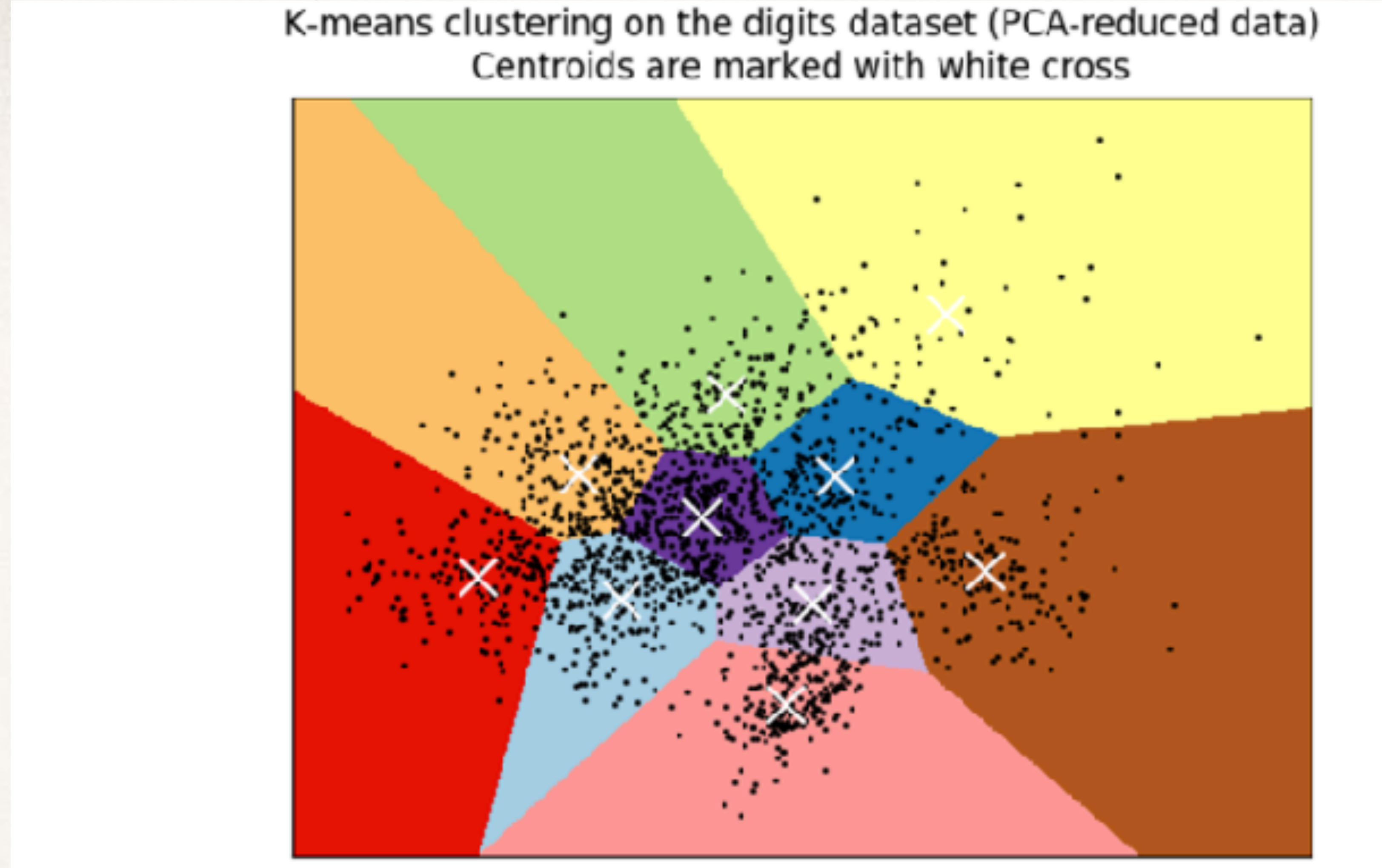
- Grouping together similar data points into distinct partitions. Items in same cluster are *more* similar to each other than to the items in other clusters.

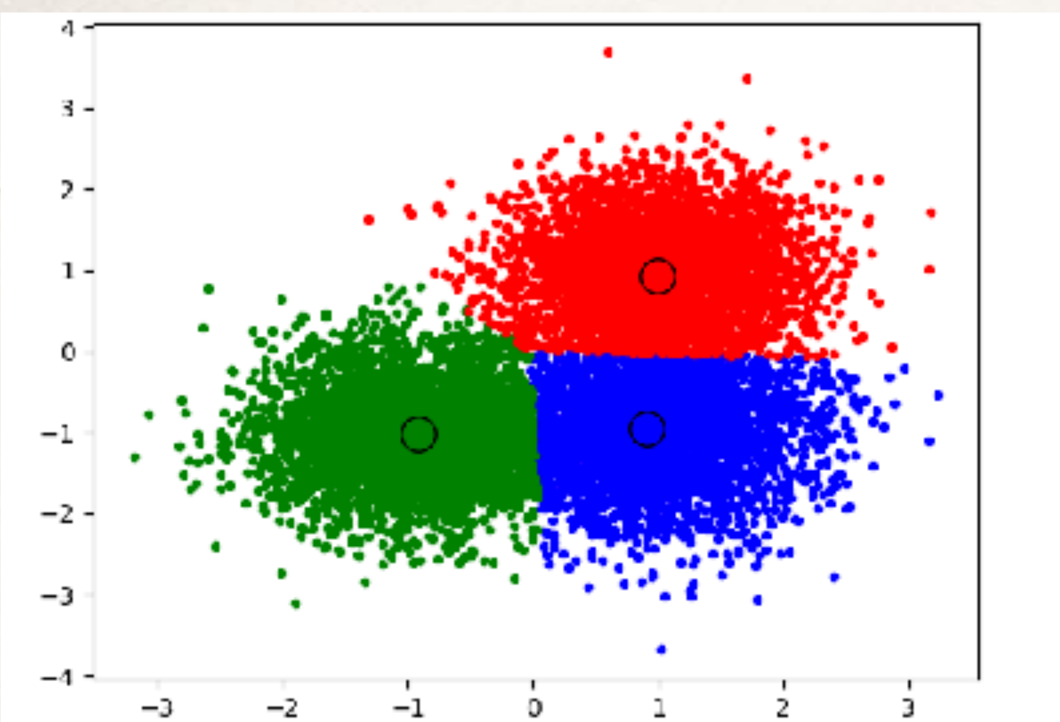
- Common unsupervised machine learning technique.

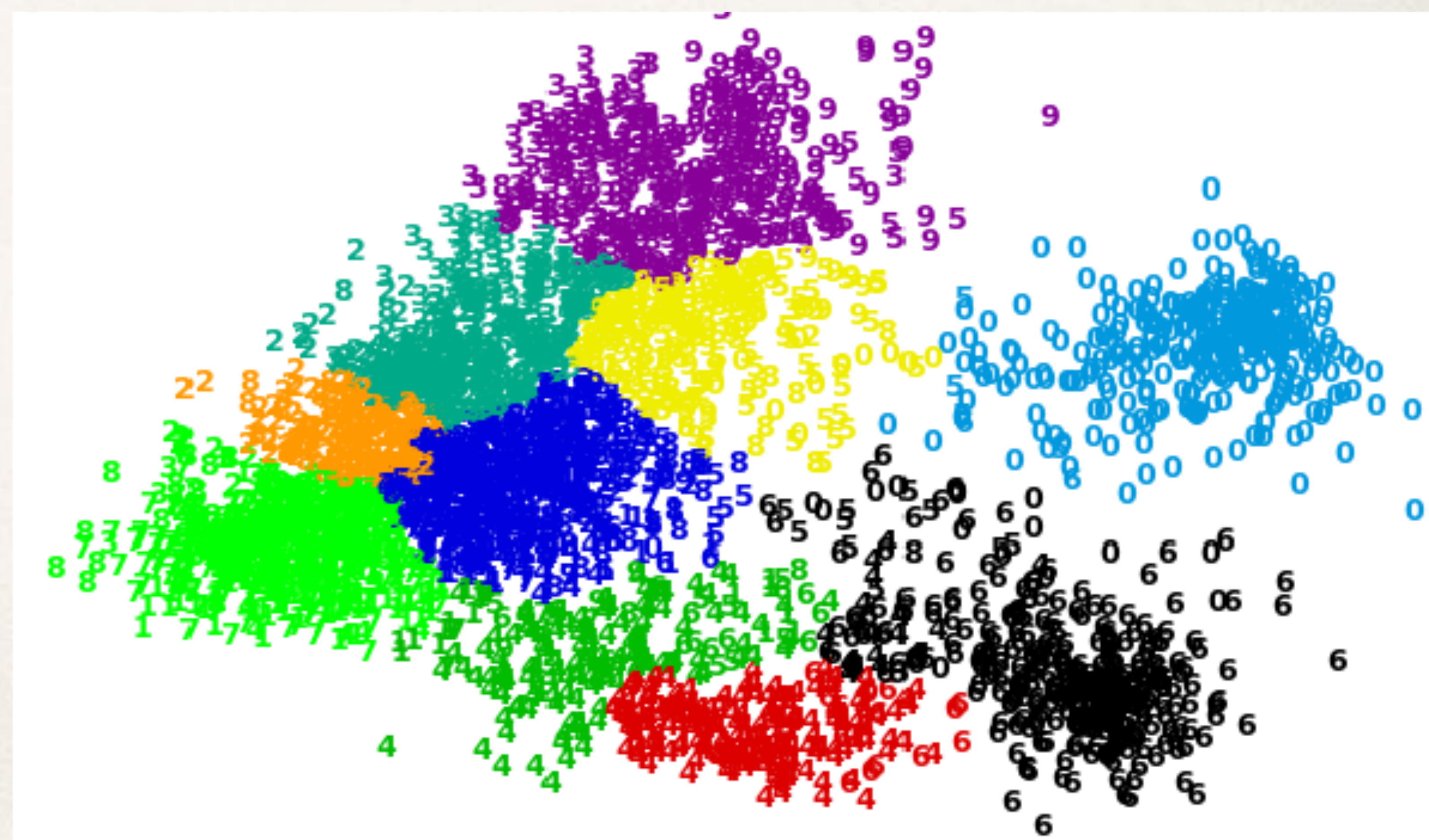
- Used for aggregating and summarizing complex multi-dimensional data. Very important exploratory step to understand data before any statistical analysis or data mining.

- From perception POV, a *scatterplot* is the best way to visualize clusters, often accompanied by a low-dimensional projection (PCA/MDS/tsne) onto 2 dimensions.


2
0
−2
−4
−6
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0

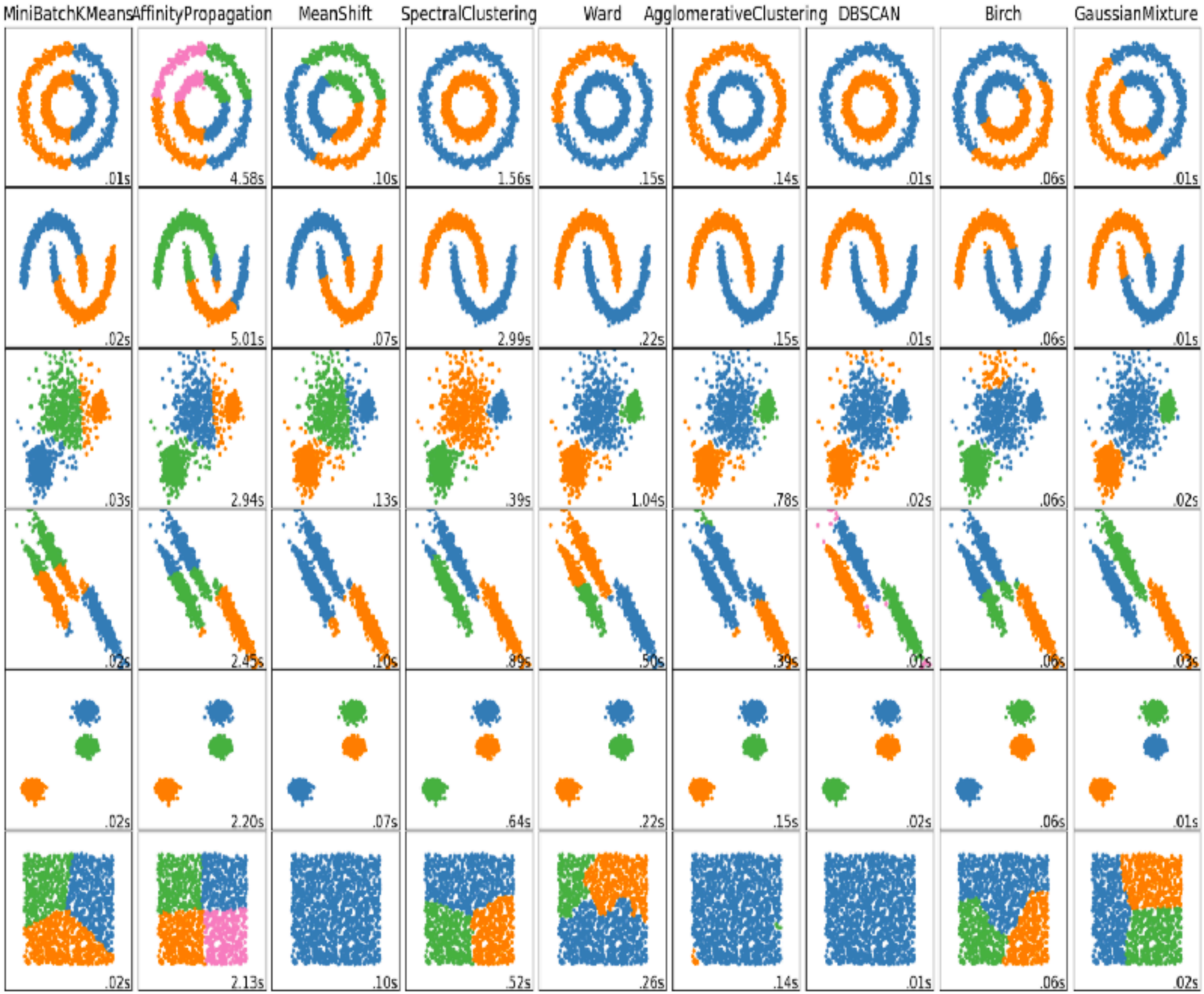

K-means clustering on the digits dataset (PCA-reduced data)
Centroids are marked with white cross


4
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3



Clustering algorithms

- There are a plethora of clustering algorithms that differ in their set of assumptions on data & clusters, and the methods to effectively find clusters.
- These algorithms have their strengths and weaknesses.
- The choice of appropriate clustering algorithm and its parameters depend on the individual data set and intended use of the results.
- However, given a particular dataset and analytical task, there are NO systematic procedures for knowing which algorithm will provide the best clustering.



A comparison of the clustering algorithms in scikit-learn

Centroid-based Clustering algorithms

- Find cluster representatives, **centroids**, and assign a data point to the cluster whose centroid is the nearest to the data point.
- Incredibly hard problem: Infinitely many possibilities, NP hard! **Slightly easier version:** k-means, assumes there are k clusters in the data. Still NP hard! Can obtain approximate solution (local minima).
- **Lloyd's algorithm:** Start with k random centroids, assign points to the nearest centroid, choose new centroid as the mean of the points in the clusters and repeat until a stopping criterion.
- Partitions data into a *Voronoi diagram*, also related to *Expectation-Maximization*.
- **Use case:** General-purpose, even cluster size, flat geometry, not too many clusters.
- **Requires:** choice of metric, apriori knowledge of k.
- **Drawbacks:** Prefers convex and isotropic clusters, not robust to randomness.
- **Modifications:** k-mediods, k-medians, k-means++, fuzzy c-means.

Connectivity-based clustering algorithms

- AKA **Hierarchical clustering,** provides a nested partitioning of the data by successively merging (*agglomerative*) or splitting (*divisive*) them, thereby producing a hierarchy which can be shown visually as a *dendrogram*.

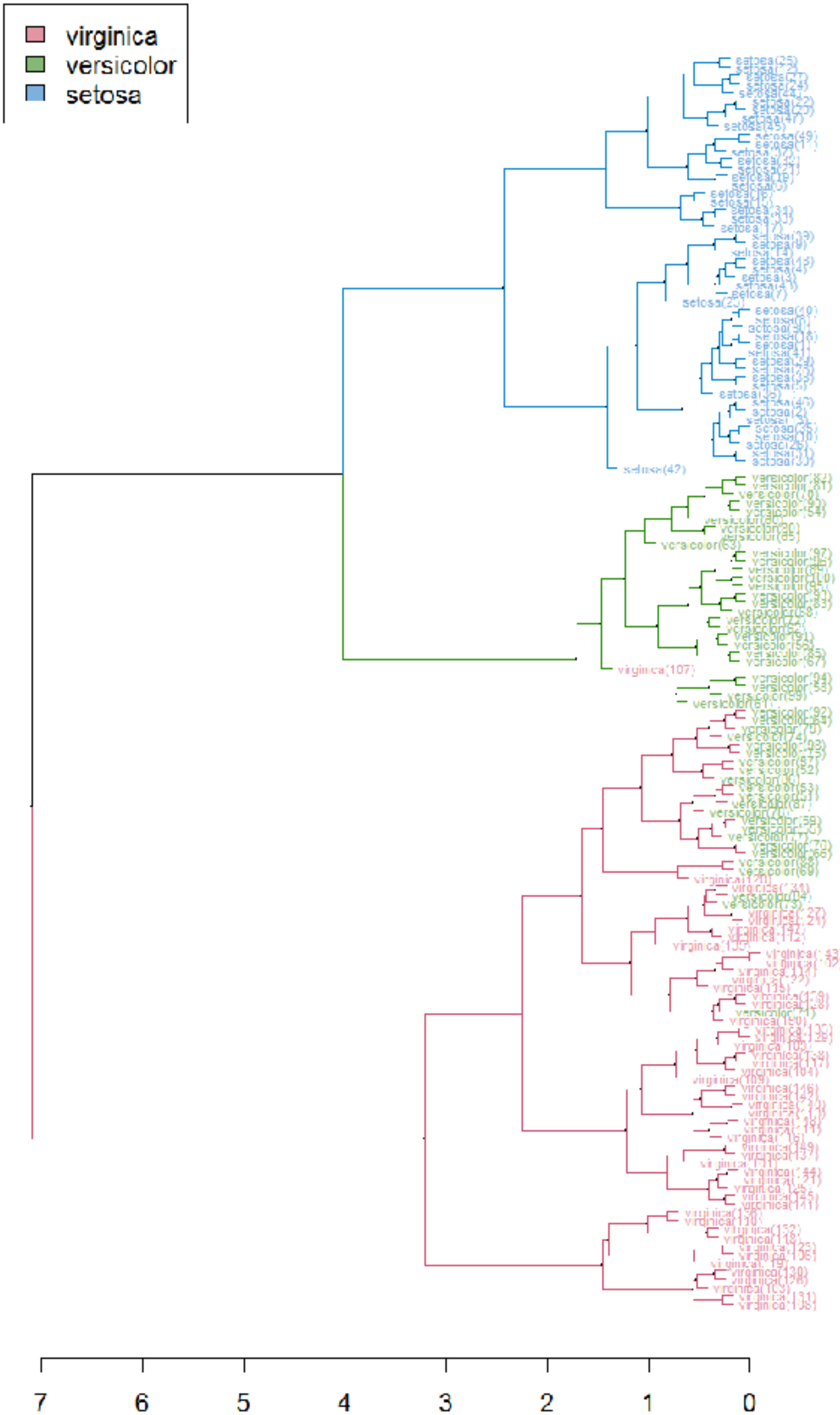
- The root of the dendrogram is the unique cluster containing all the data points, and the leaves are clusters each containing exactly one data point.

- e.g. in *Agglomerative clustering*, each data point starts as an individual cluster and are then merged in successive steps.

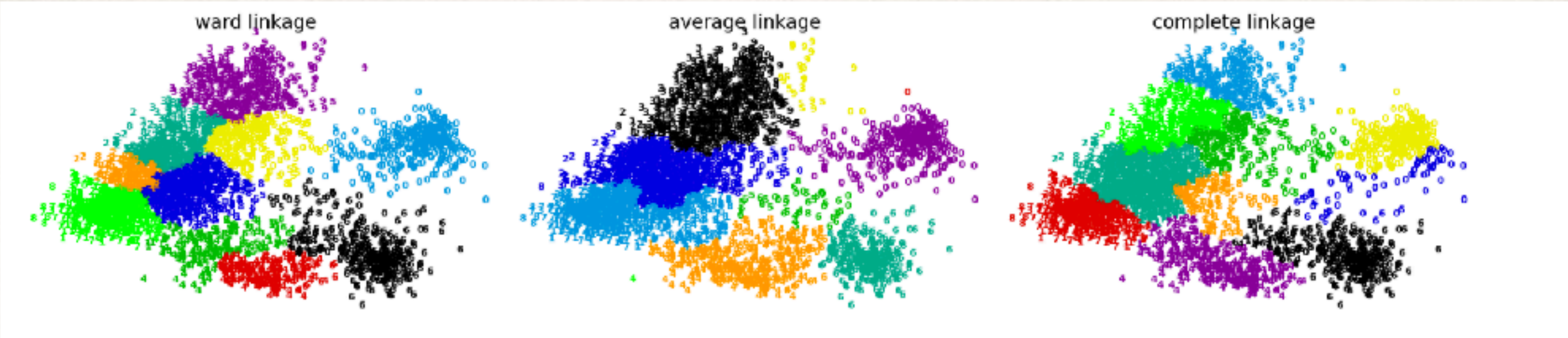
- **Use case:** Many clusters, possibly connectivity constraints, non Euclidean distances.

- **Requires:** Choice of a metric (distance between two data points), Linkage criterion (distance between two clusters).

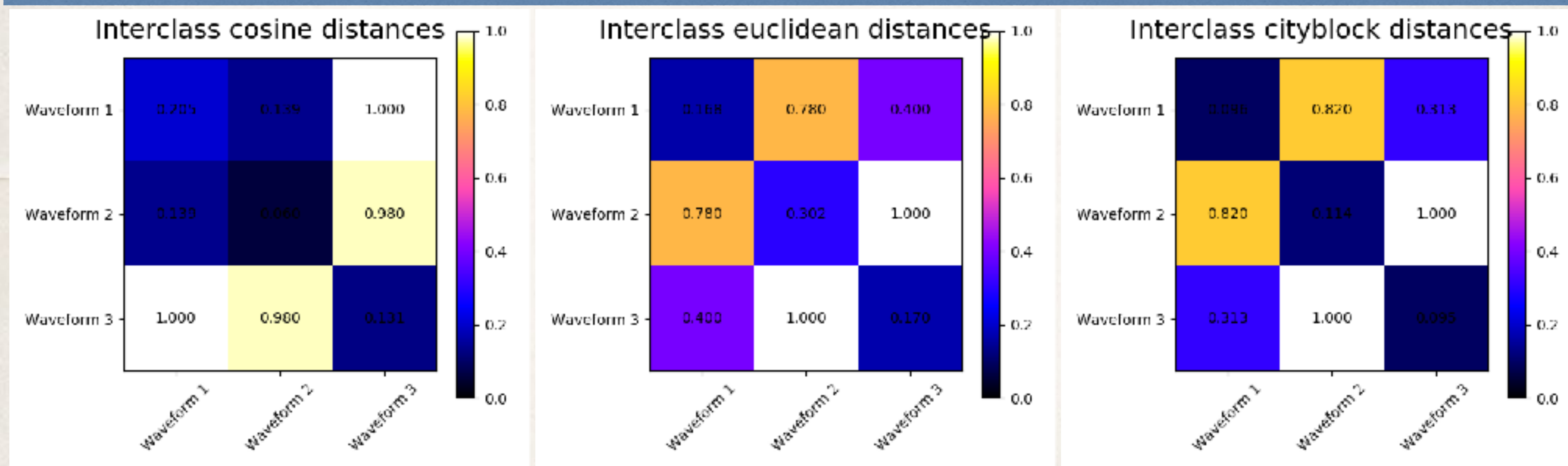
Clustered Iris data set
(the labels give the true flower species)



Different linkages in sklearn

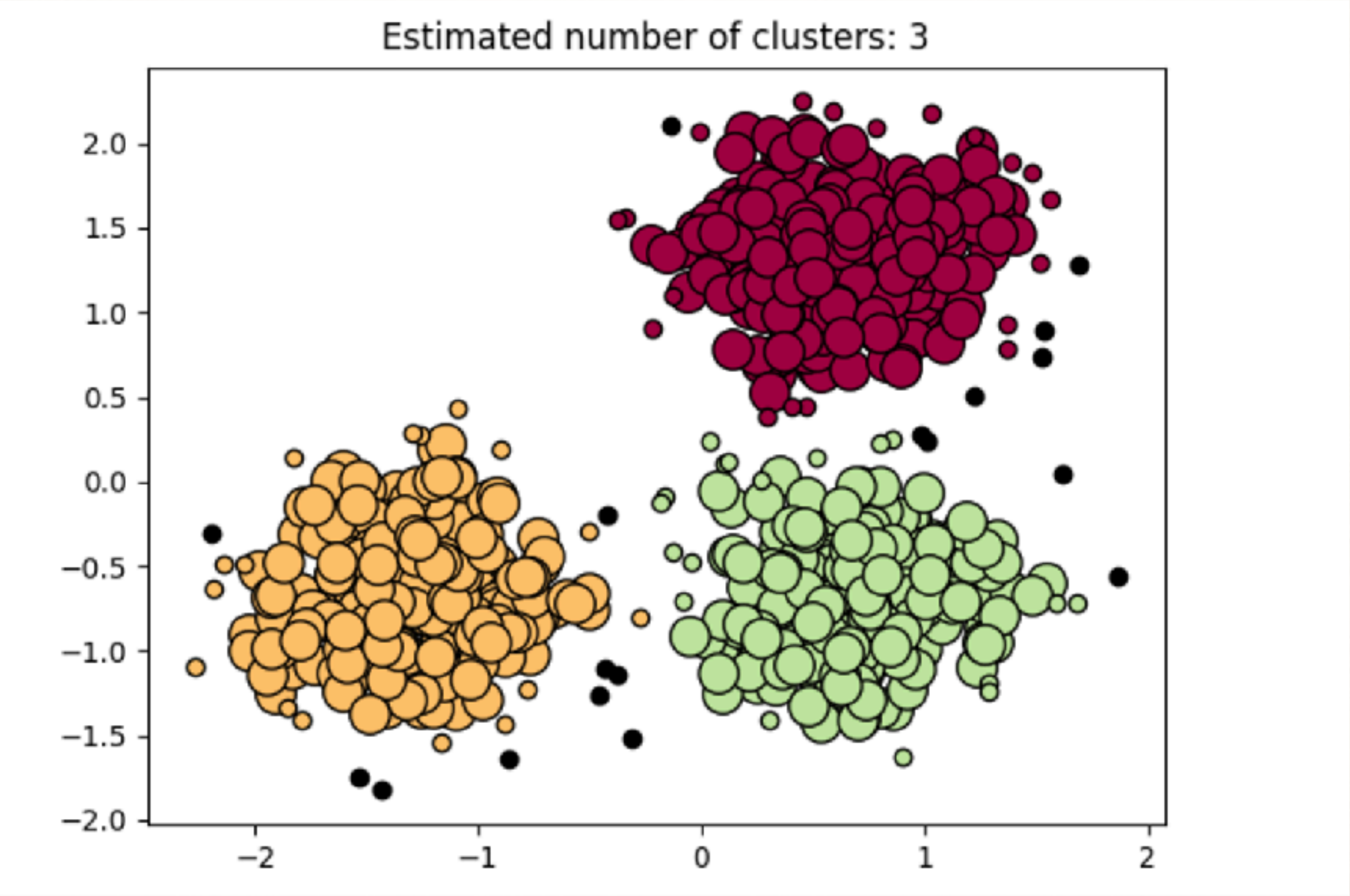


different metric choices



Density-based clustering algorithms

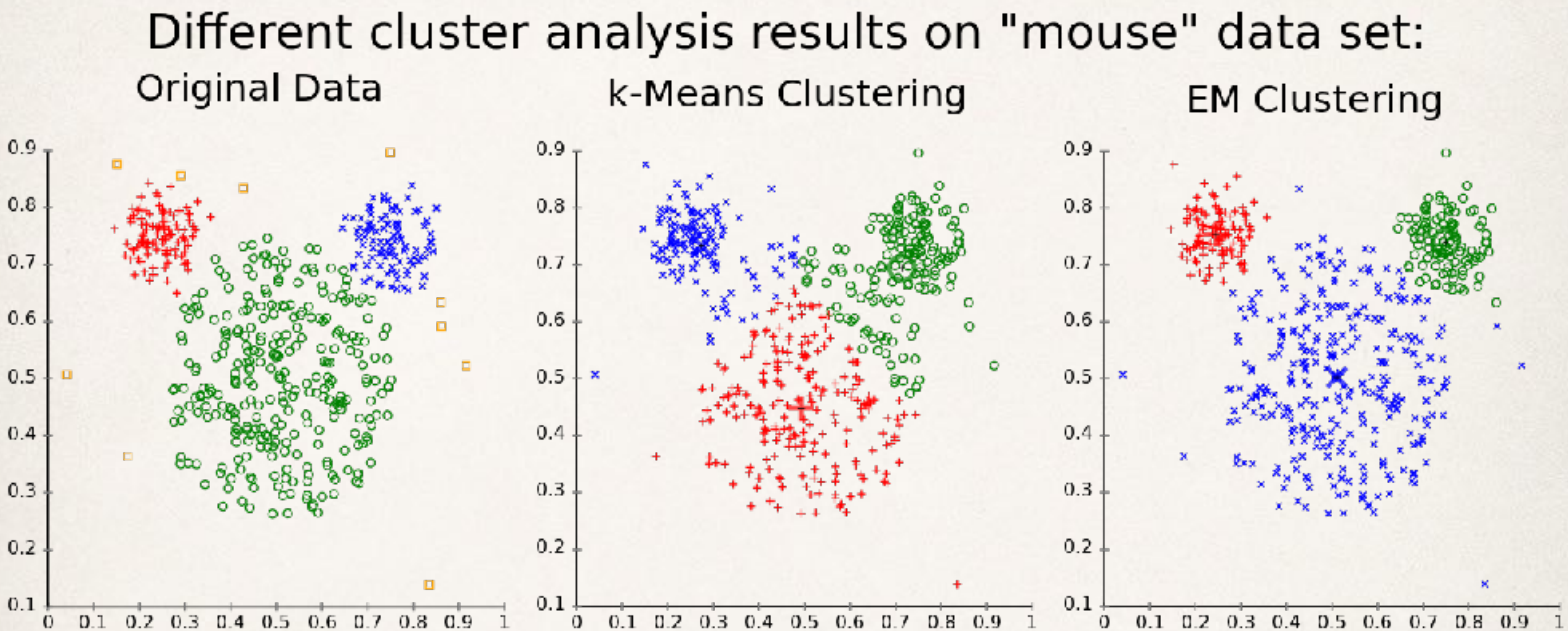
- Attempts to find regions of high density (clusters) in the data which are separated by regions of low density (boundaries/noise).
- Can detect clusters of any shape, not just convex.
- e.g. **DBSCAN:** Find highly dense regions as clusters and assign points in the low-density regions to the cluster they are closer to. Unassigned points are *outliers*.
- **Use case:** Non-flat geometry, uneven cluster sizes, outliers
- **Requires:** Quantify density (e.g. set of points for each of which there exists m number of points at a distance less then d, in *sklearn*, *min_samples* and *eps*).
- **Drawbacks:** Need sharp density gradient to detect clusters, not effective where the gradient is continuous e.g. a mixture of Gaussians.
- **Variants:** OPTICS, Mean Shift



Estimated number of clusters: 3
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2

Probability-based clustering algorithms

- Distinct clusters are samples from distinct probability distributions. Assume a distribution model, and try to separate based on the parameter estimates.
- **Gaussian Mixture model:** The data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters which are estimated using *Expectation-Maximization* algorithm.
- Usually overfits unless constrained model is used, e.g. fixing the number of Gaussians (number of clusters), in fact GMM is a generalisation of k-means to include covariance.
- Can also constrain covariance.
- **Use case:** Flat geometry, good for density estimation.
- **Requires:** k (usually), covariance constraint, convergence threshold for EM algorithm, initialization for parameters.



Cluster analysis performed on an artificial dataset ("Mouse", similar to a well-known comic figure) comparing k means and EM clustering results.

Issues in cluster analysis

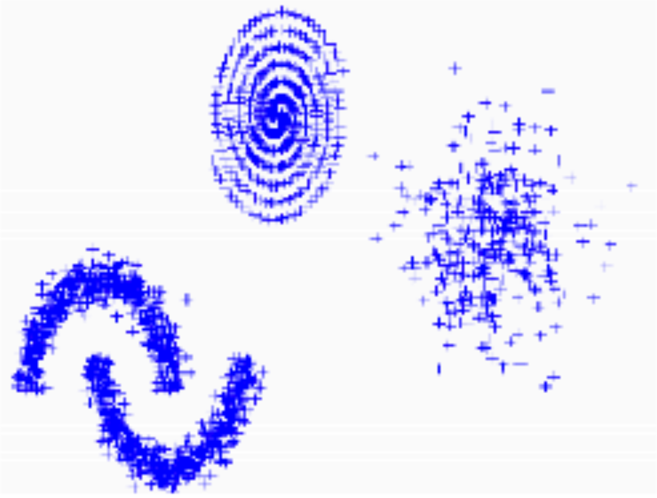
- So many algorithms to choose from, NO systematic-mathematical way to decide. Optimal parameters for the chosen algorithm depends on the dataset and the analytical task at hand.

- *"The notion of cluster can't be precisely defined. Clustering is in the eyes of the beholder."* - **Why so many clustering algorithms,** *Vladimir Estivill-Castro*

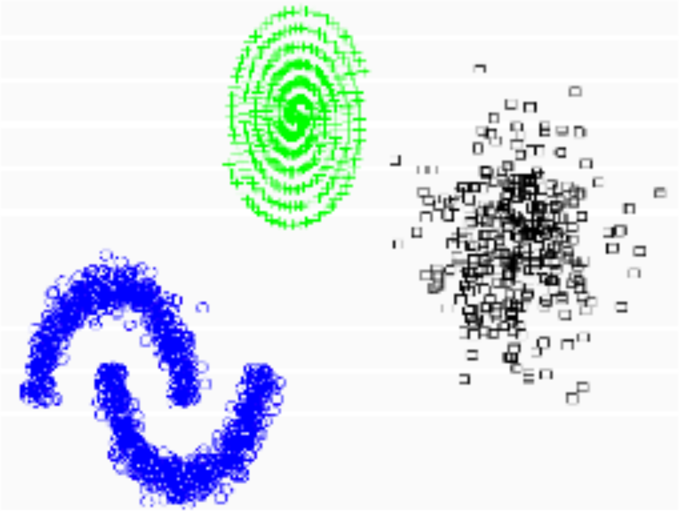
- Ability to compare various clustering results and estimate quality of a clustering result.

- Unsupervised method - lack of ground truth. Evaluation is difficult.

Clustering



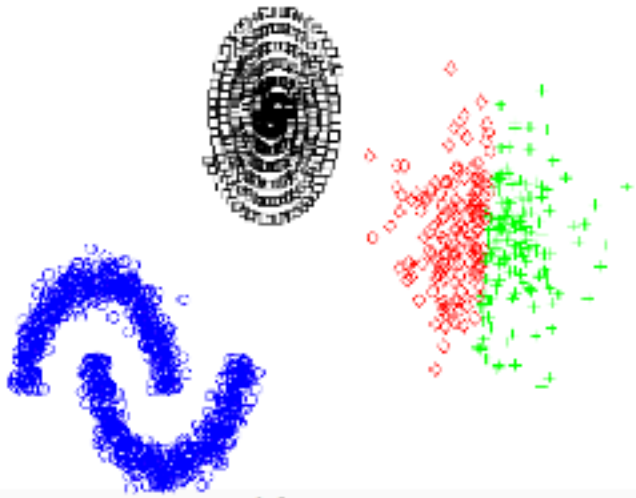
Data



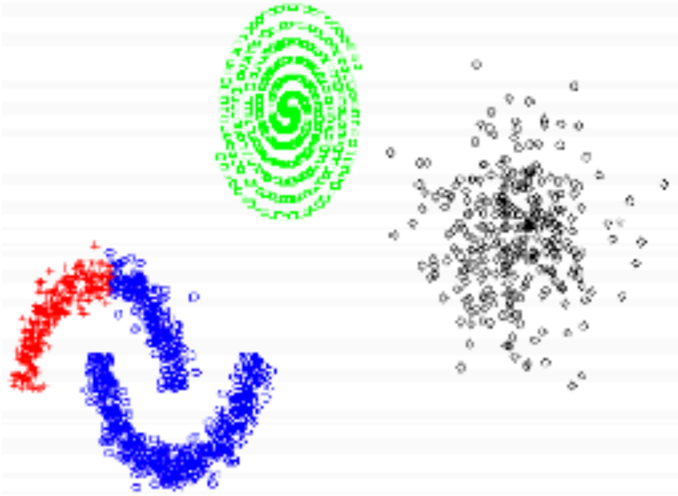
DBSCAN

Different clustering methods output different partitions!

Which method do I pick?



Kmeans



Complete Linkage HAC

Clustering evaluation metrics

- Difficult task, at least as difficult as clustering (*Pfitzner et al*).
- **External evaluation:** results are compared with ground truth. Not practical.
- **Internal evaluation:** results are aggregated into a single statistic. Without ground truth.
- **Manual evaluation:** human expert makes decision, not practical in this big data paradigm.
- Internal evaluation with human-in-the-loop ?

Silhouette coefficient

- The *Silhouette Coefficient* is a measure of how similar a point is to its own cluster compared to other clusters, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.

- For a data point x proposed to be in cluster C, the *Silhouette Coefficient* is defined as

$$s(x) = \frac{D_x - C_x}{max(C_x, D_x)}$$

- where C_x is the average distance between x and all other points in C, and D_x is the distance between x and all the points in the cluster nearest to x.

- The *Silhouette coefficient* for the whole clustering is defined to be mean of the values for all the data points. The value lies in (–1, 1) where the higher the value of the coefficient, the better the clustering .

- **Reference:** P. J. Rousseeuw *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis*. Journal of Computational and Applied Mathematics, 20:53–65, 1987 (sklearn.metrics.silhouette_score)

Calinski-Harabaz index

- The *Calinski-Harabaz index* of a clustering is defined as the ratio of the between-cluster variance and the within-cluster variance.
- Well-defined clusters have higher between-cluster variance and lower within-cluster variance, thus higher value of the index.
- For k clusters, CZ(k) = Total inter-cluster variance / Total intra-cluster variance, where

$$\text{Inter-cluster variance} = \sum_{i=1}^{k} n_i ||m_i - m||^2$$

$$\text{Intra-cluster variance} = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - m_i||^2$$

- Here i-th cluster, C_i has mean m_i and contains n_i elements, and m is the overall mean of the data.
- **Reference:** M. Kozak. *"A dendrite method for cluster analysis" by Calinski and Harabaz: A classical work that is far too often incorrectly cited.* Communications in Statistics - Theory and Methods, 41(12):2279–2280, 2012 (sklearn.metrics.calinski_harabaz_score)

Davies-Bouldin coefficient

- Similar to *Calinski-Harabaz index*, is defined as the average over all clusters the ratio of within-cluster dispersion and the pairwise between-cluster dispersion.

$$DB(k) = \frac{1}{k}\sum_{i=1}^{k} max_{i \neq j} DB_{ij}$$

$$DB_{ij} = \frac{\bar{D}_i + \bar{D}_j}{\bar{D}_{ij}}$$

- Here $\bar{D}_i$ is the average distance between each point in the i-th cluster and its centroid, and $\bar{D}_{ij}$ is the average distance between the centroids of the i-th and the j-th cluster.

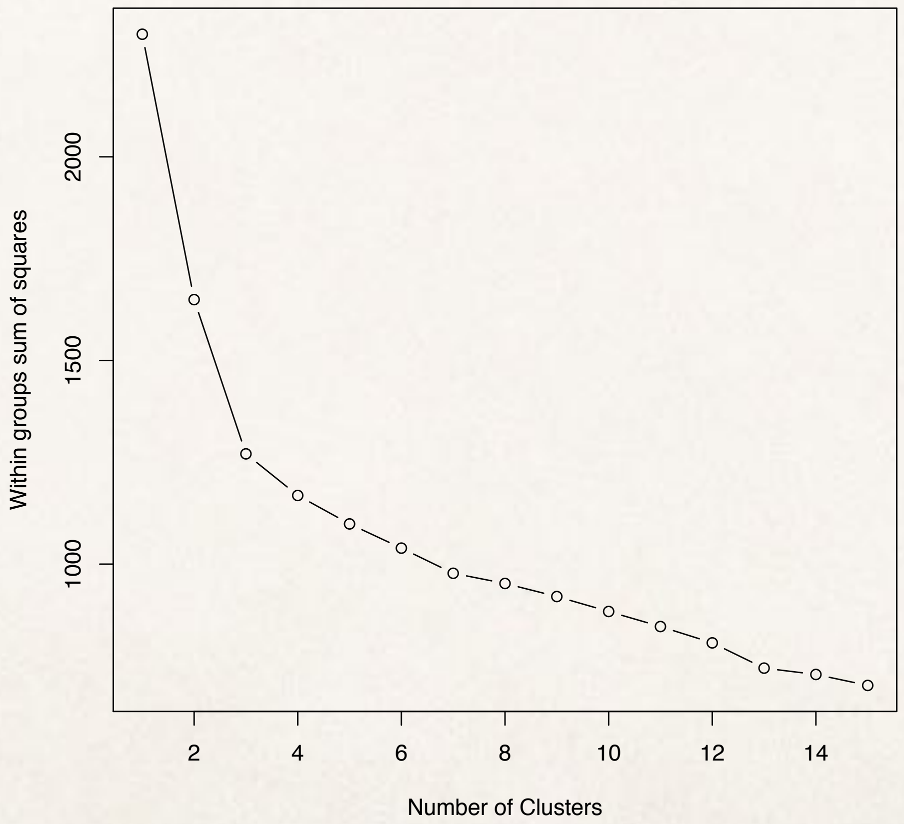
- Smaller the *Davies-Bouldin index*, better the clustering results are.

- **Reference:** D. L. Davies and D. W. Bouldin. *A cluster separation measure.* IEEE Trans. Pattern Anal. Mach. Intell., 1(2):224–227, Feb. 1979.

Gap statistic

- The *elbow method*, where a metric(usually within to between-cluster distance ratio) is plotting against an internal parameter, is a popular and intuitive method to find the optimal value of the parameter. *Tibshirani et al* provided a statistical formulation of this technique and defined *gap statistic*.

- The idea is to consider clustering of random permutations of the data to observe how they compare with a null reference distribution of data with no clustering structure.



- **Reference:** R. Tibshirani, G. Walther, and T. Hastie. *Estimating the number of clusters in a dataset via the gap statistic*. 63:411–423, 2000.

Sdb_w

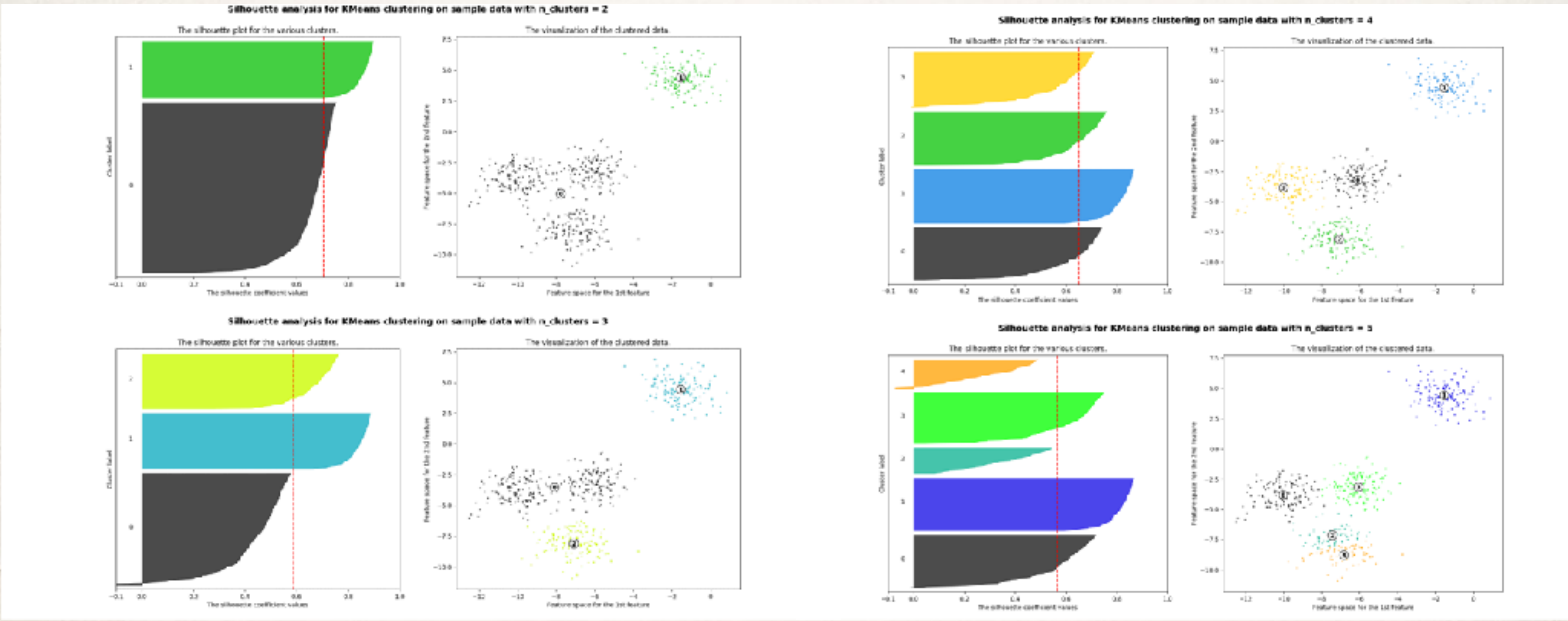
- *Sdb_w* attempts to measure quality by taking into consideration the *compactness*, *separation*, and the *density* of the clusters .

- Relies on the notion of the *density of a point x relative to a pair of clusters* which is equal to the number of points in these clusters which are inside a ball centered at *x*.

- Defined under the assumption that for each pair of clusters, the density of at least one of the centroids must be greater than the density of their midpoint to have a good clustering

- Reference: M. Halkidi and M. Vazirgiannis. *Clustering validity assessment: Finding the optimal partitioning of a data set*. In Proceedings of the 2001 IEEE International Conference on Data Mining, ICDM '01, pp. 187–194. IEEE Computer Society, Washington, DC, USA, 2001

Selecting number of clusters using Silhouette analysis



k=3,5 are not good options!!

Which is the best (most optimal) value of k?

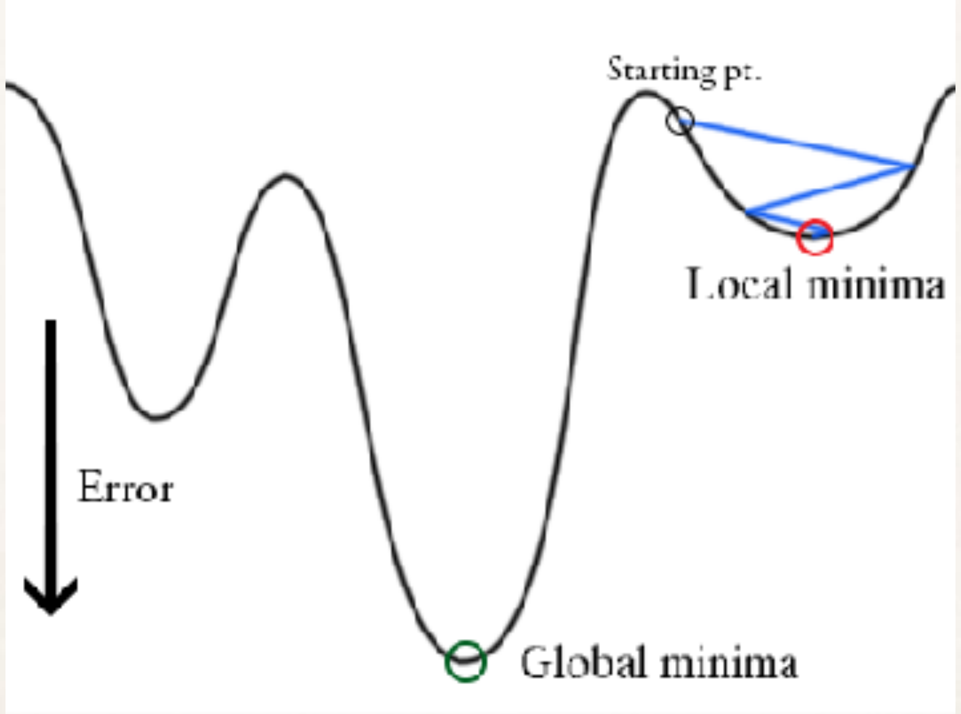
Clustervision

- *Clustervision* is a visual analytical tool that helps ensure data scientists find the right clustering among the large amount of techniques and parameters available.
- Developed by researchers at IBM Watson Research Center.
- The system clusters data using a variety of clustering techniques and parameters and then ranks clustering results utilizing five quality scoring metrics.
- The visual user interface allows users to find high quality clustering results, explore the clusters using several coordinated visualization techniques, and select the cluster result that best suits their task.
- To appear at IEEE Vis October 2017.

Clustervision
Current Dataset: bobross
Dimensionality Reduction Technique: tsne
Number of Clusters:
2 - 20
Rerun
SuperPoint Scale:
Clustering: #5
Cluster 1
Members
Cohesion
Separation
Silhouette
Inliers
Outliers
#126: 1.042
#136: 1.042
#109: 1.174
#197: 1.174
#121: 1.226
Data Point 372
Name
Score
Tree
Ocean
Mountain
Waves
Structure
Snowy Mountain
Trees
Beach
Cabin
Mountains
Conifer
Deciduous
River
Palm Trees
Clouds
Fence

Randomness in clustering

- Many clustering algorithms like k-means, GMM use a random initialization.
- Since these algorithms are approximate solutions to the optimization problem, they attempt to find a local minima.
- Which local minima is found, depends on the initial state, and thus we can get different results for multiple runs on same data, using same algorithm and its parameters.



- Also makes difficult to compare the clustering results from different models.

Consensus clustering

- One way to get better hold of the sensitivity to random initialization is to do multiple runs of the clustering algorithm, and observe the differences between the runs.

- Ideally, define a statistic which quantifies the differences among the results of different runs in the ensemble.

- First, define the *consensus matrix* whose entries reflect the probability that two different data items belong to the same cluster. Perform clustering *m* times, then the *ij*-th entry in the consensus matrix C_ij = *#(i and j are in same cluster)/m*

- Define *dispersion* of the clustering as

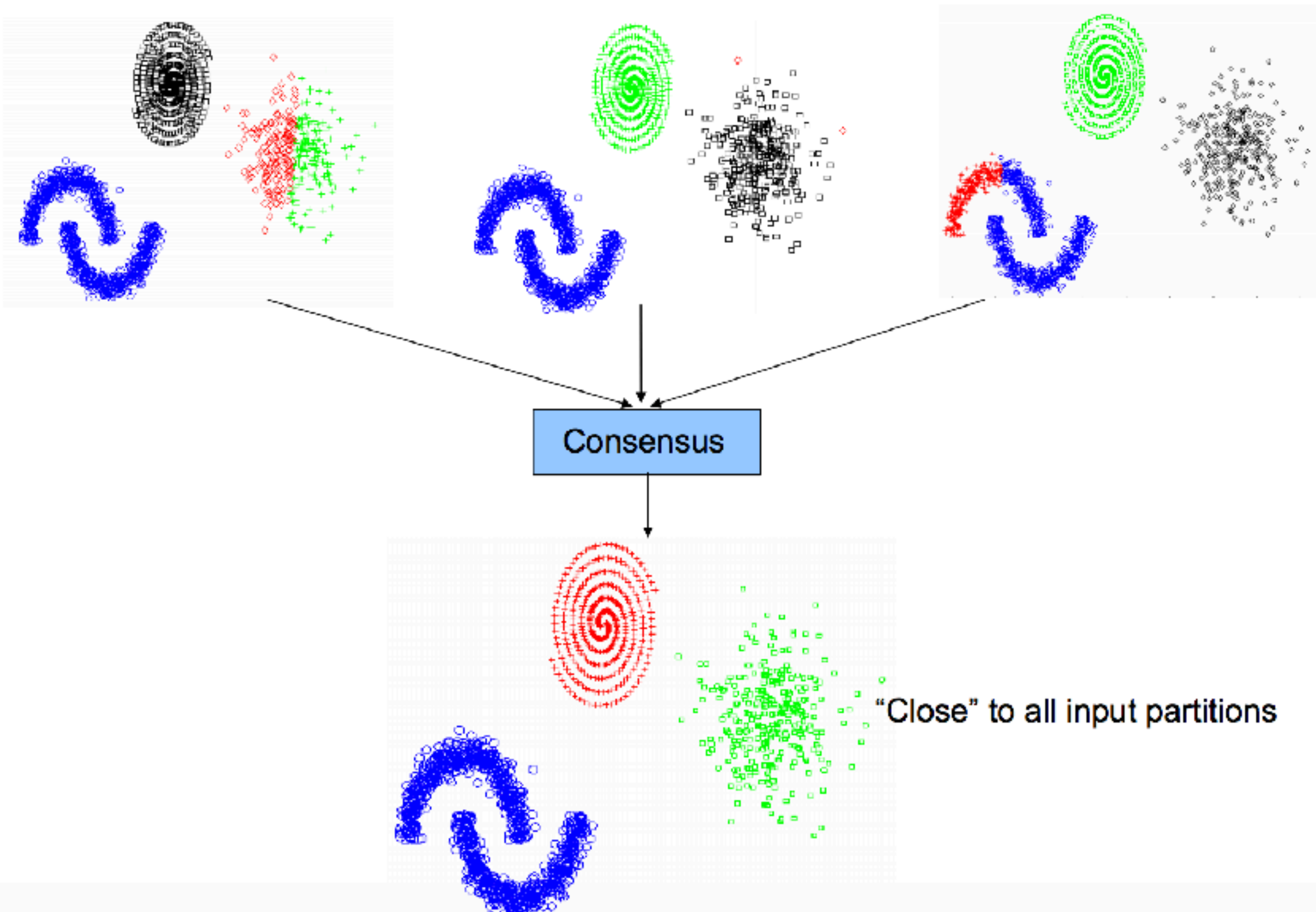
$$\rho = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} 4 * \left(C_{ij} - \frac{1}{2}\right)^2.$$

- The value of the coefficient is 1 for a perfect consensus matrix (all entries 0 or 1). Ideally we want this value to the as close to 1 as possible. This indicates that the different clusterings in the ensemble are statistically similar and are thus robust of the random initializations.

Consensus clustering

- This can also be used as a clustering evaluation metric and thus can be used to compute the optimal number of clusters. Compute dispersion for different values of k and then choose k with maximal value of dispersion.
- Another such metric is *cophenetic correlation*.
- If multiple clusterings have been obtained for a given dataset e.g. for different algorithms, same algorithm different parameters, different initialization etc. it's desirable to obtain a single clustering which is an aggregate of all the runs in the ensemble. Such a clustering, called *consensus clustering*, provides a reconciliation of clusterings from different sources.
- Many different ways to compute the consensus clustering.

Reconcile!



Consensus clustering computation

- The rows of the consensus matrix provides a vector representation for the data points in terms of how they were clustered across multiple runs.
- Compute the point-wise similarity.
- Various metrics like cosine, Euclidean, KL-divergence etc. can be used to compute the similarity.
- Now we perform another clustering on the similarity matrix to obtain the consensus.
-