

UNSUPERVISED ML

OBJECTIVE

- Discover information from data without labeled examples
- Extract some hidden organisation, patterns, relation between elements
- Extract a (statistical ?) model of the data ?

OBJECTIVE

- Typical objectives:
 - ▶ Cluster discovery
 - ▶ Anomaly Detection
 - ▶ Latent variable discovery / Embedding / dimensionality reduction...

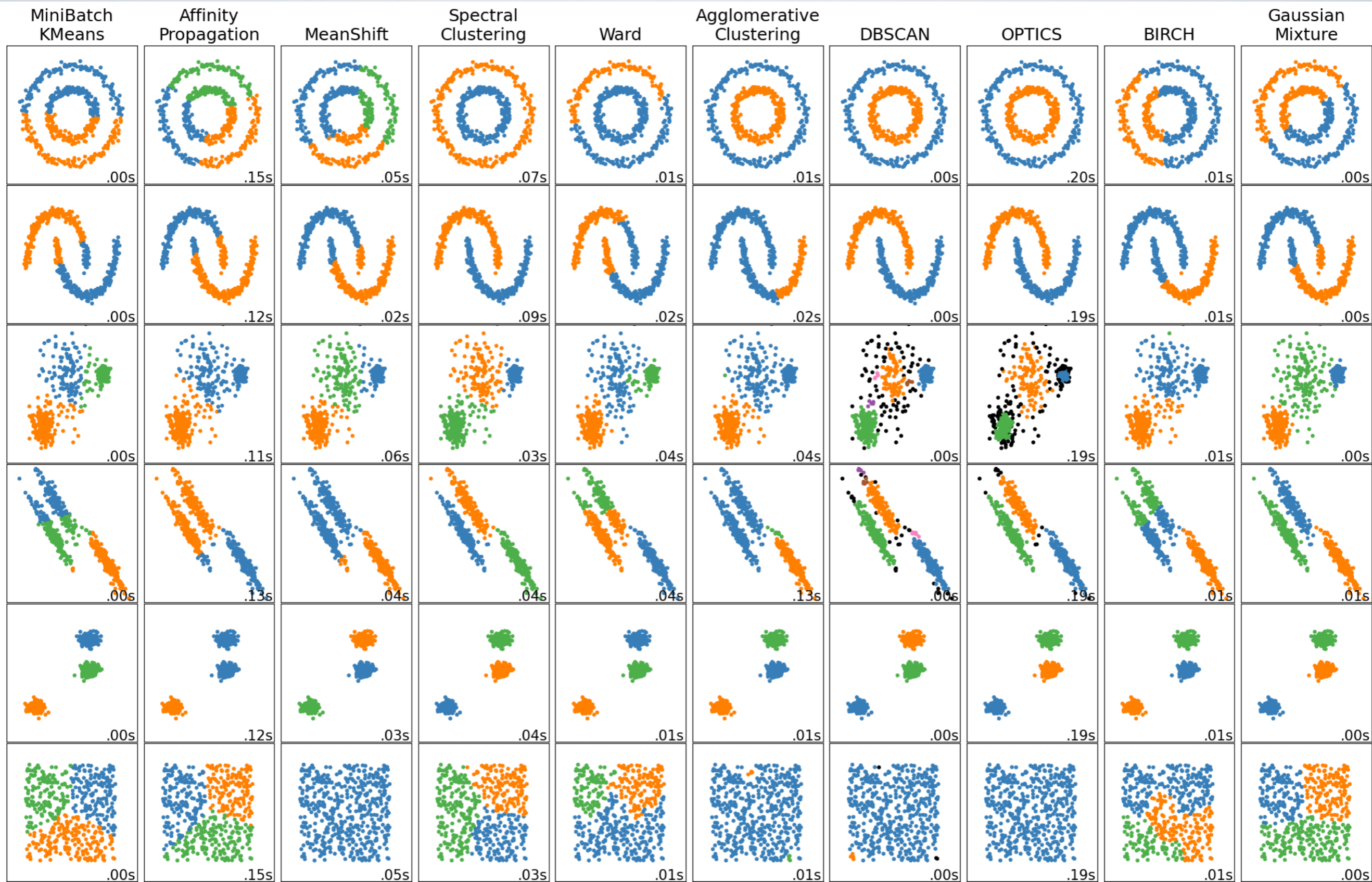
CLUSTERING

CLUSTERING

- The most famous unsupervised ML problem
- 100+ methods exist
 - Most people use “good old” methods: k-means (1967), DBSCAN (1996)
 - They are often “good enough”, well implemented, safe, ...
- Part of the problem: Clustering is not well defined
 - What is “a good cluster”?

CLUSTERING

- How would you define a good cluster ?
- A good partition in clusters ?



K-MEANS

- Definition:

- ▶ For a target number of clusters k
- ▶ Find the item assignment minimizing
 - The inter-cluster variance (weighted by cluster size)
 - Equivalently \Rightarrow The squared distance from points to their cluster center
 - Equivalently \Rightarrow The squared distance between cluster elements

K-MEANS

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg \min_{\mathbf{S}} \sum_{i=1}^k |S_i| \text{Var}(S_i)$$

with

\mathbf{S} a cluster assignment,

k a number of clusters

x a d dimensional item, and

$\boldsymbol{\mu}_i$ the centroid of items in the cluster \mathbf{S}_i .

K-MEANS

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg \min_{\mathbf{S}} \sum_{i=1}^k |S_i| \text{Var}(S_i)$$

This is only one possible objective for clustering!

For instance, why using the **squared distance**?

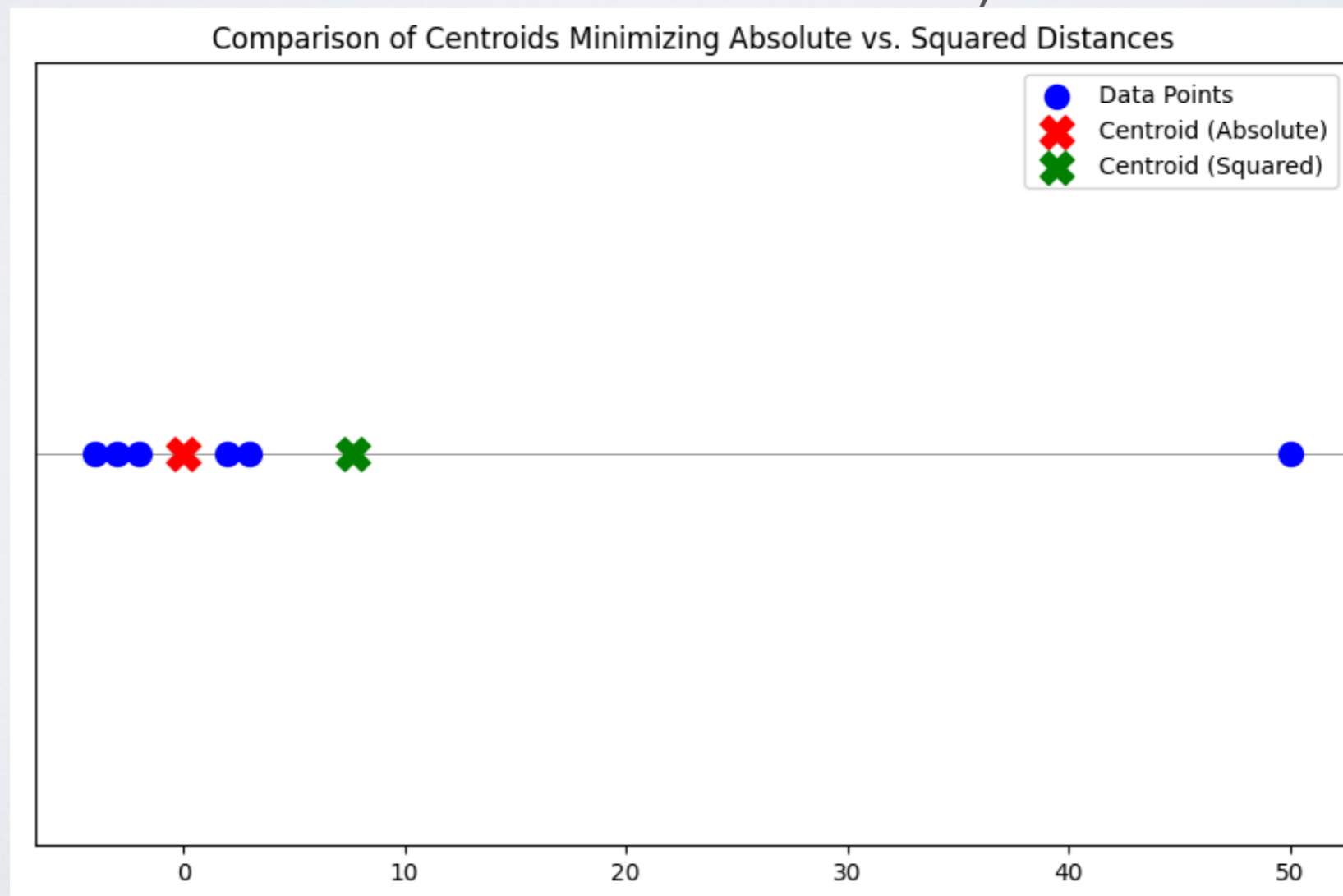
=> Good math properties (derivation), history

=> Consequence: outliers penalized more (pros and cons)

K-MEANS

=> Consequence: outliers penalized more (pros and cons)

Squared distance minimized by the **mean**.
Absolute distance minimized by the **median**.



K-MEDOIDS

Same method, replacing the squared distance by the absolute distance

K-MEANS

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg \min_{\mathbf{S}} \sum_{i=1}^k |S_i| \text{Var}(S_i)$$

Note that without fixing k , there is a trivial solution with each item alone in its own cluster.

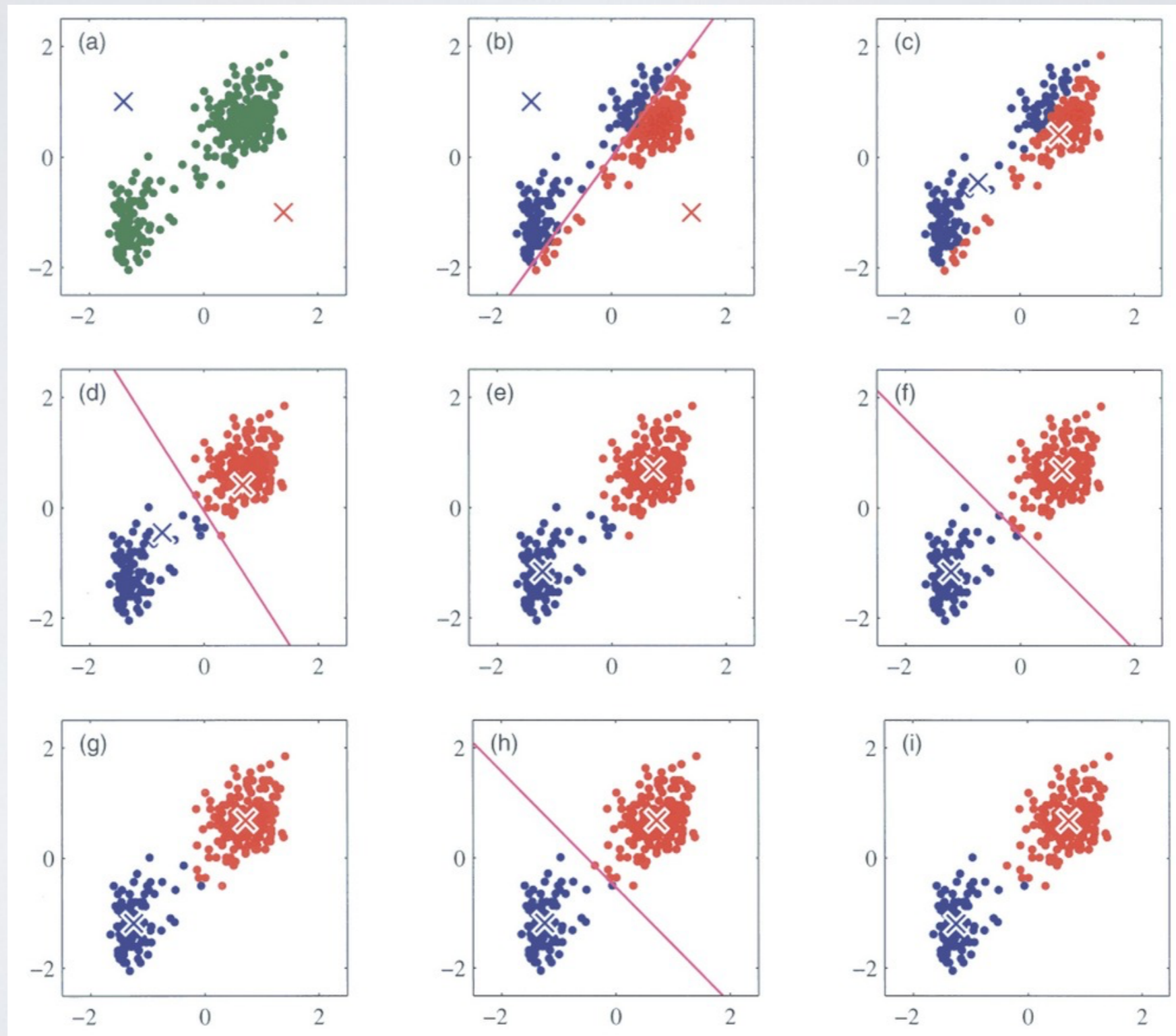
K-MEANS

- Discovering the global optimum is NP-hard
- How to find quickly a good solution ?
 - Naive k-means
 - K-means ++ (used in most current implementations)
 - Use optimized data structure (KDtrees...)

NAIVE K-MEANS

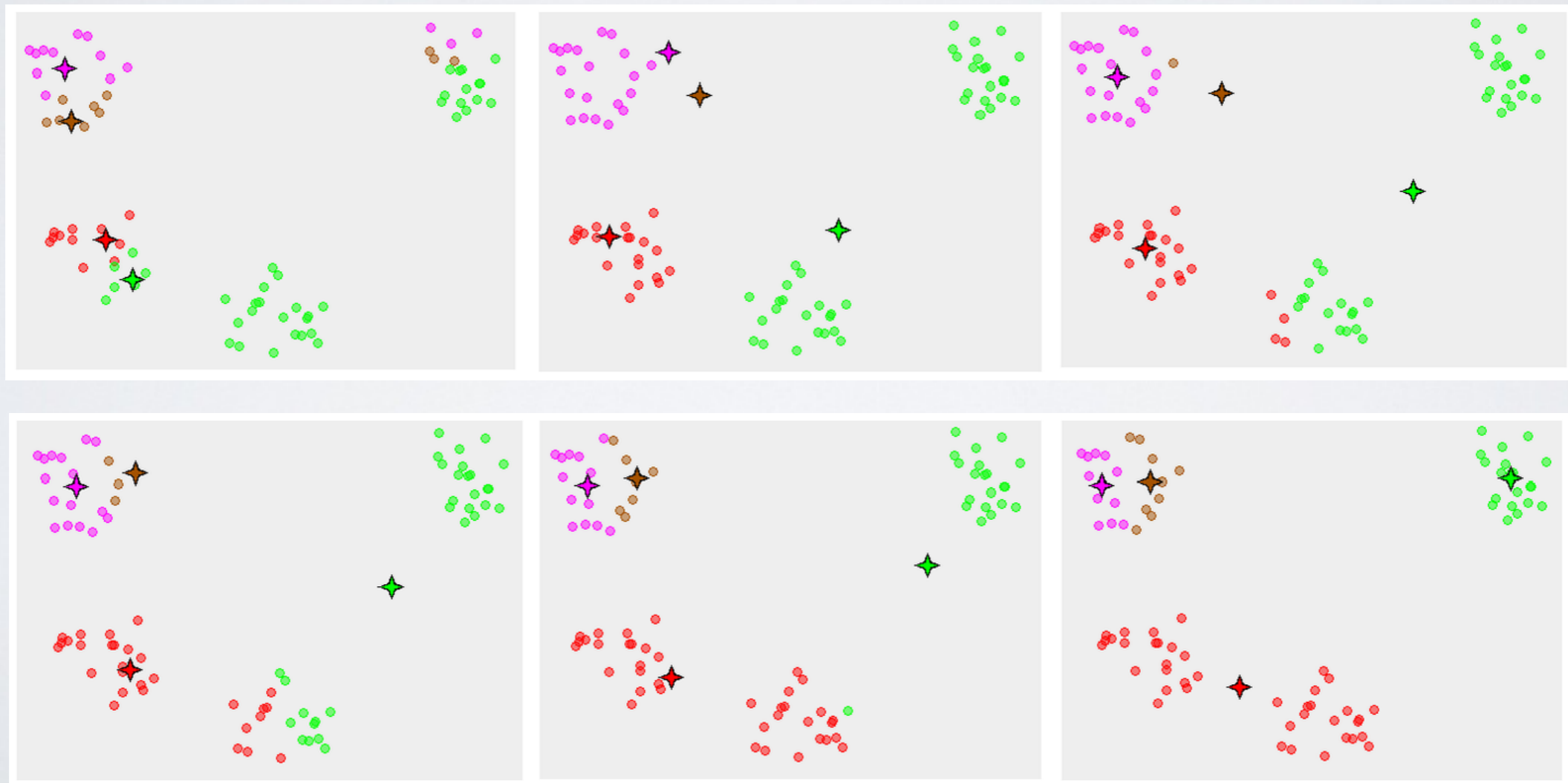
- 1) Assignment: Assign each item to its closest cluster center
- 2) Update: Recompute the center of each cluster as the mean (centroid) of items that compose that cluster
- Start with random centroids

NAIVE K-MEANS



NAIVE K-MEANS

- Known limit: convergence to poor local minimum if poor initial centroids



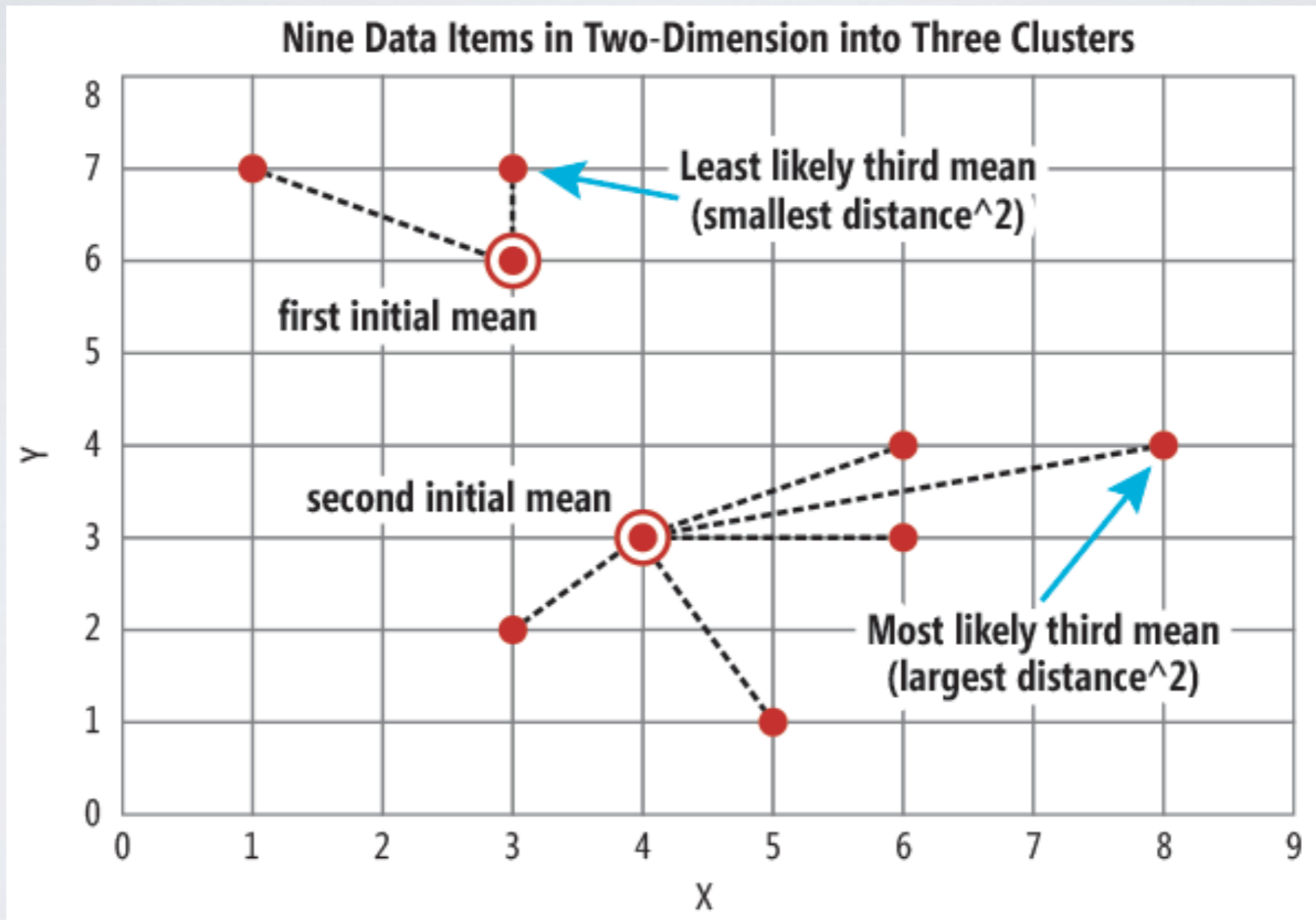
K-MEANS++

- Several variants to choose wisely the initial centroids
- K-means++ is proven to improve the results, statistically
 - Not always, but improves more often than deteriorate the results.

K-MEANS++

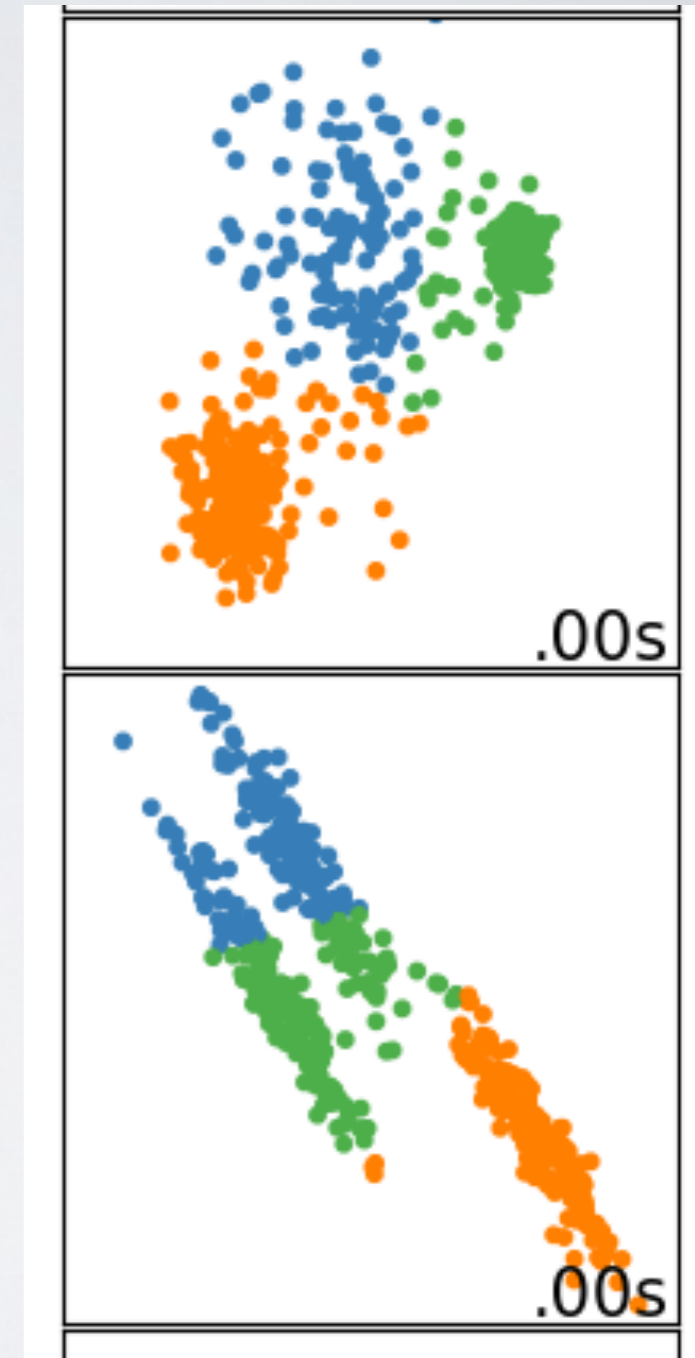
1. Choose one center uniformly at random among the data points.
2. For each data point x not chosen yet, compute $D(x)$, the distance between x and the nearest center that has already been chosen.
3. Choose one new data point at random as a new center, using a weighted probability distribution where a point x is chosen with probability proportional to $D(x)^2$.
4. Repeat Steps 2 and 3 until k centers have been chosen.

K-MEANS++



WEAKNESSES

- We can identify some clear weaknesses:
 - ▶ K-means has a tendency to search for clusters of equal sizes (minimize **overall** cluster variance)
 - ▶ Clusters tend to be **circular**, since all directions are worth the same.



NORMALIZATION

- Important point: k-means is based on **Euclidean distance**.
 - ▶ We minimize the inter-cluster Euclidean distance between points
 - ▶ We could adapt the method to other distances
- Data needs to be **normalized/standardized**
 - ▶ Clustering based on age in years and revenue in \$. The “distance” in \$ will dominate
 - ▶ Remember: normalization/standardization are not fixing magically problems (outliers..)
 - You need to **think**: Is 1 unit in one dimension *worth* 1 unit in other dimensions?

GAUSSIAN MIXTURES

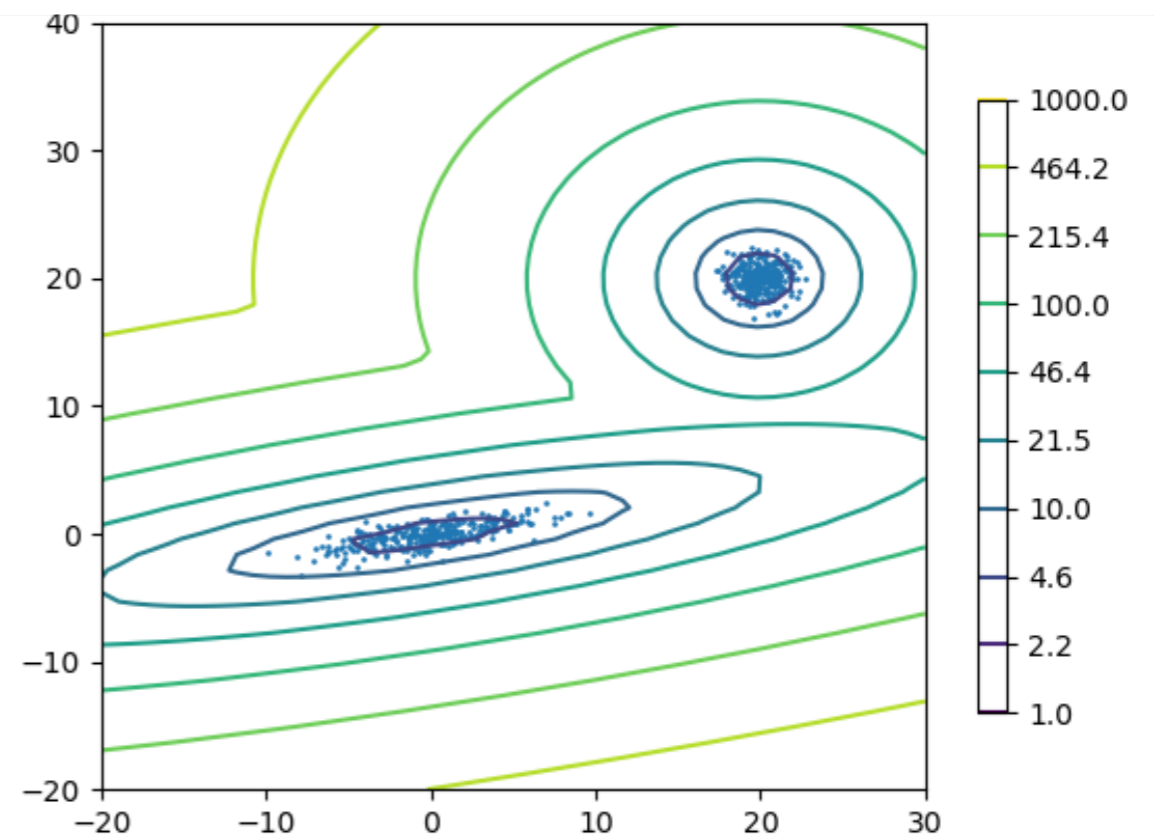
GAUSSIAN MIXTURES

- Generalize k-means concept:
 - Clusters are sets of points that are close in euclidean space
 - Different clusters tend to be far appart
- Translate it statistically:
 - Each cluster can be described using a normal distribution centered on its centroid, with the probability of observing points decreasing with the distance to the centroid.

GAUSSIAN MIXTURES

Train accuracy: 88.4

Test accuracy: 92.1

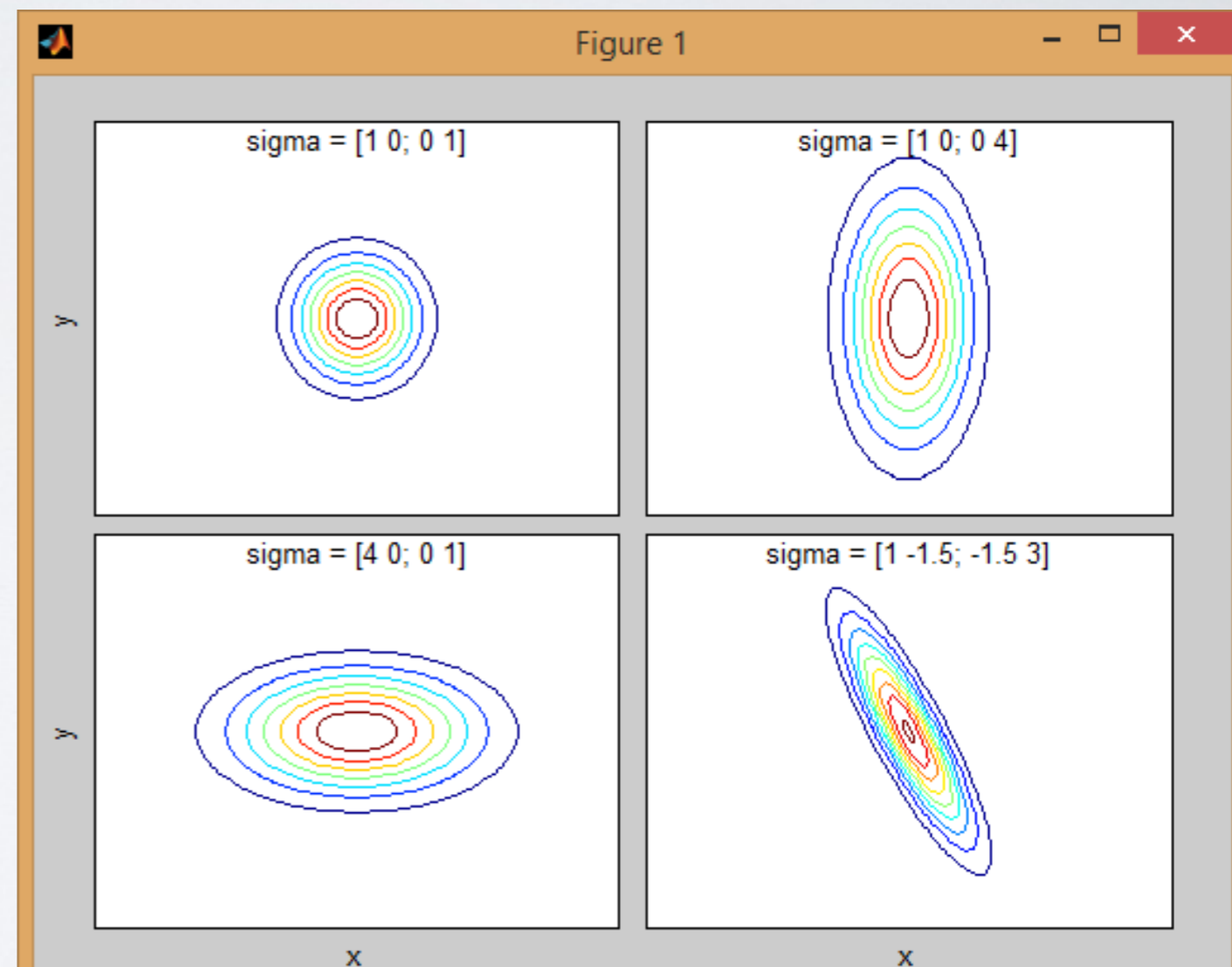


GAUSSIAN MIXTURES

- We define a **generative model** for k clusters
 - Each cluster corresponds to a gaussian distribution, defined by a center and a *variance, or covariance matrix*
 - The problem to solve is to find the parameters Θ (centers, variances) that maximize the likelihood of the corresponding model to generate the observed items X
 - More formally, we are searching for: $\arg \max_{\Theta} p(X | \Theta)$

MULTIVARIATE GAUSSIAN

- A gaussian is defined by
 - a mean
 - a variance
- A multivariate gaussian is defined by a
 - A center
 - a covariance matrix



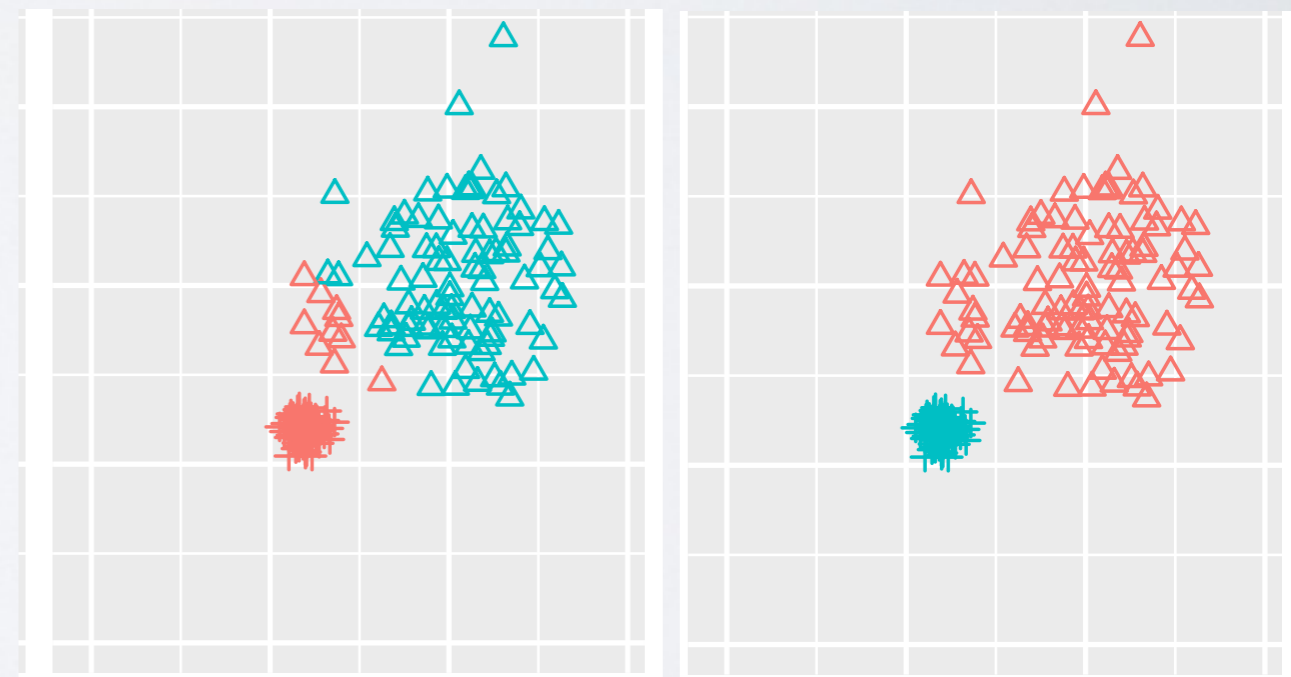
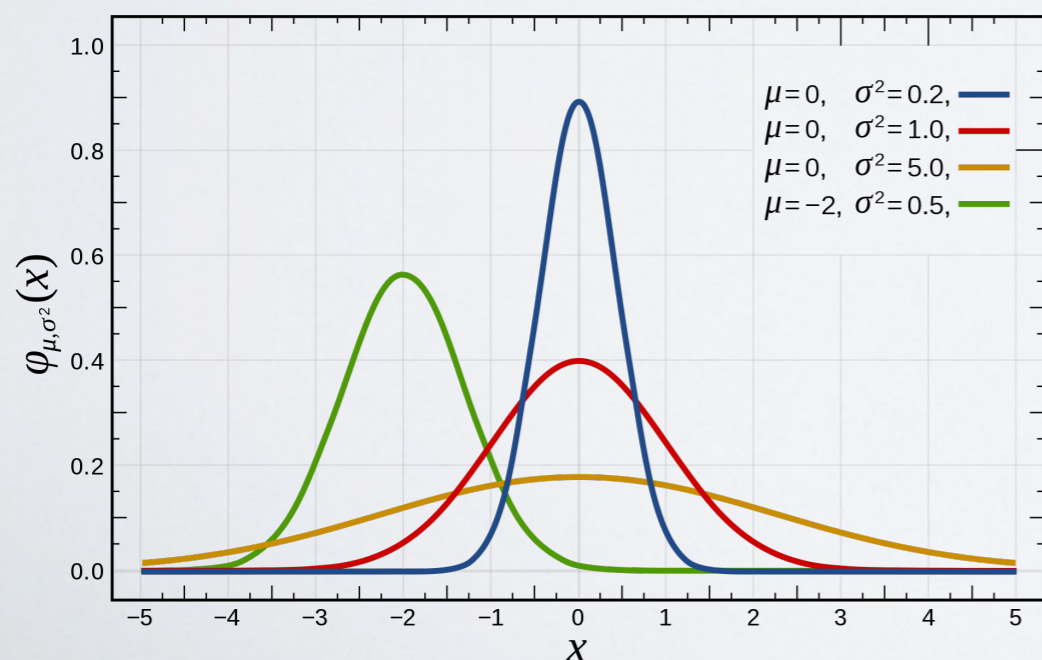
K-MEANS EQUIVALENCE

$$\begin{bmatrix} \text{Var}(x_1) & \dots & \text{Cov}(x_n, x_1) \\ \vdots & \cdot & \vdots \\ \text{Cov}(x_n, x_1) & \dots & \text{Var}(x_n) \end{bmatrix}$$

- If we assume that:
 - ▶ The gaussian distributions are defined only by their variance, not by complete covariance matrices
 - Similar in all directions, “spherical”
 - ▶ The variance value is the same for all gaussian distributions
 - Spheres of the same “size”
 - ▶ The probability for each item to be generated by each of the gaussian distribution is identical
- Then it can be shown that the objective is equivalent to the k-means objective !
 - ▶ We can relax some of those constraints to get richer results

DENSITY HETEROGENEITY

- Allowing denser/sparser clusters
 - ▶ Consider the case in which Gaussians are defined by a single value of variance (covariance=0)
 - ▶ If they differ for each cluster, some can be denser than others

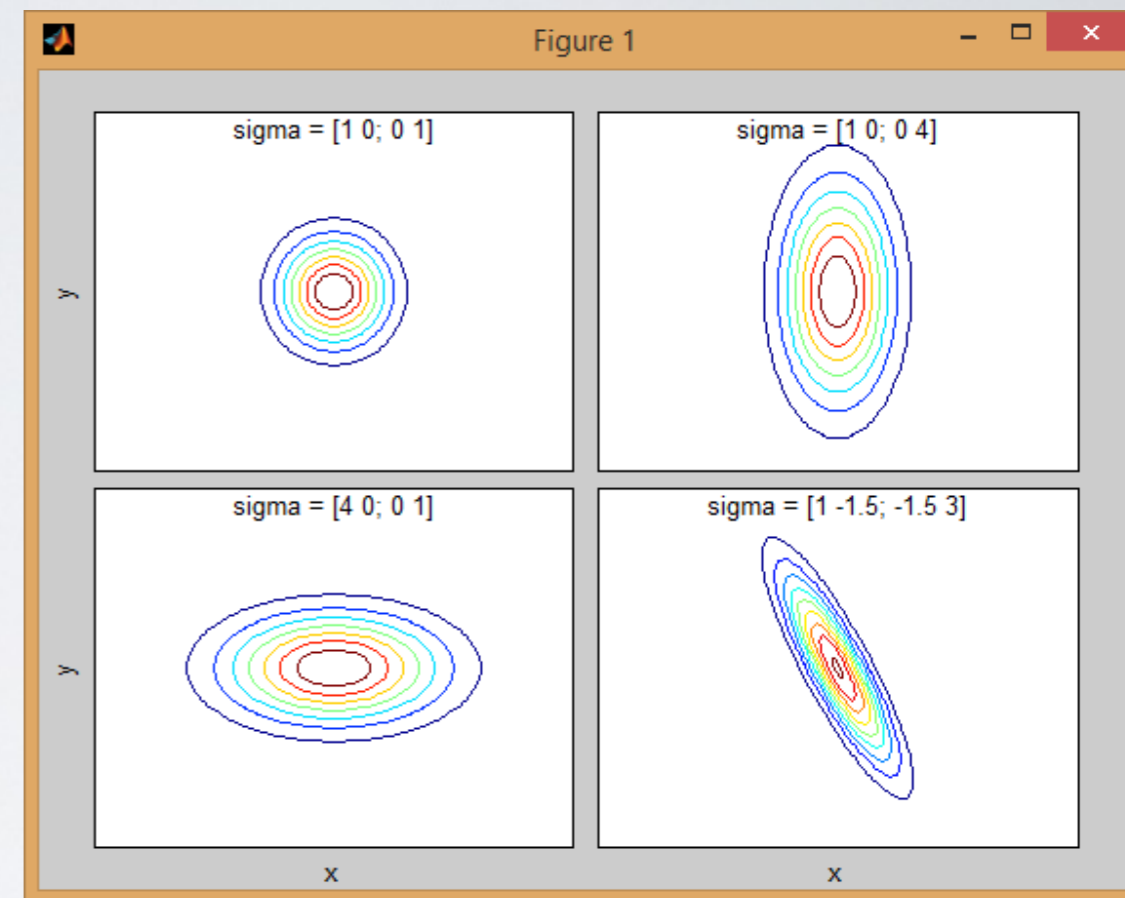


K-means

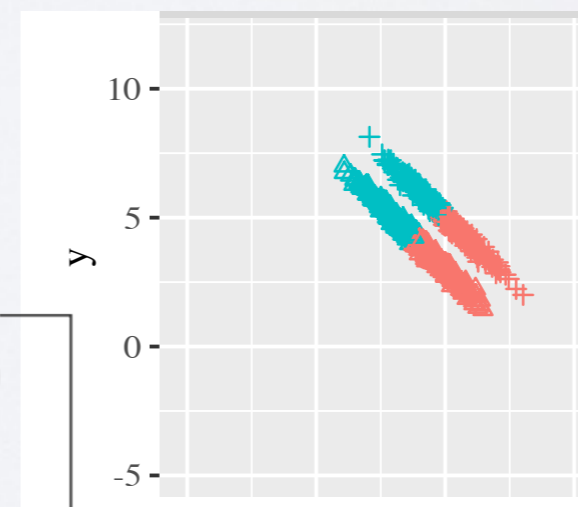
GM,
free variance

SHAPE VARIATIONS

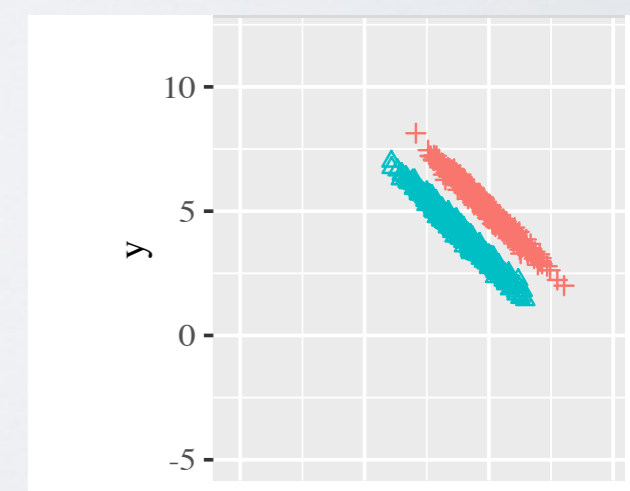
- Allowing non-circular shaped clusters
 - ▶ If values on the diagonal of the covariance matrix differs, the matrix can have ellipsoidal shape, in the direction of the axes
 - ▶ If the full covariance matrix is inferred, any ellipsoidal shape can be obtained



$$\begin{bmatrix} \text{Var}(x_1) & \dots & \text{Cov}(x_n, x_1) \\ \vdots & \ddots & \vdots \\ \text{Cov}(x_n, x_1) & \dots & \text{Var}(x_n) \end{bmatrix}$$



K-means

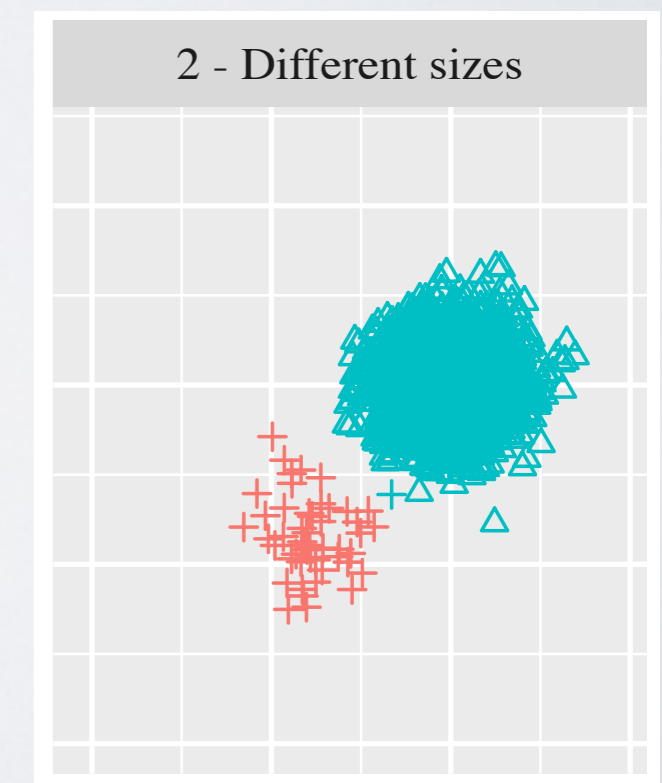
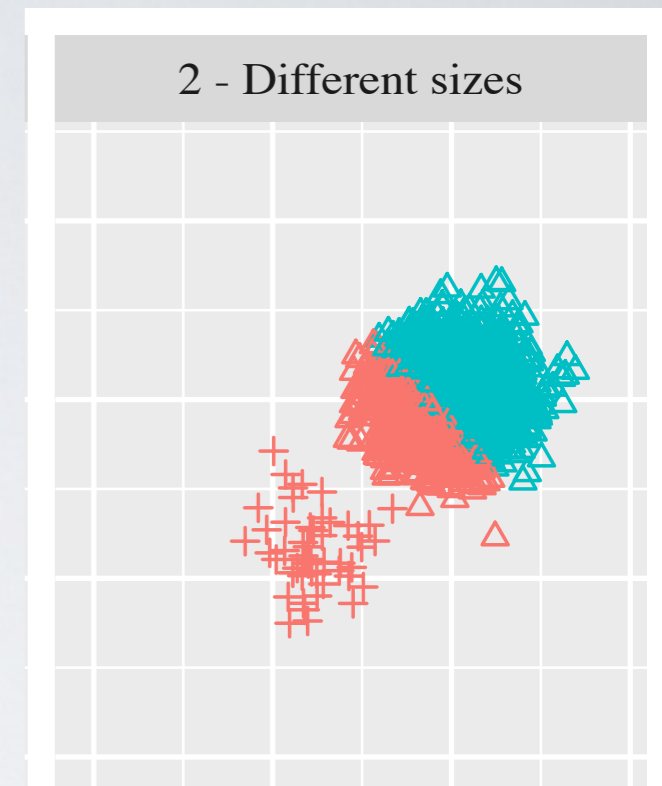


Full gaussian

SIZE HETEROGENEITY

- The fraction of all items generated by each generative gaussian (e.g., cluster) is the same.
- We usually add a *strength* parameter π to weight the fraction of items generated by each cluster

$$p(X) = \sum_{k=1}^K \pi_k G(X | \mu_k, \sigma_k)$$



ALL TOGETHER

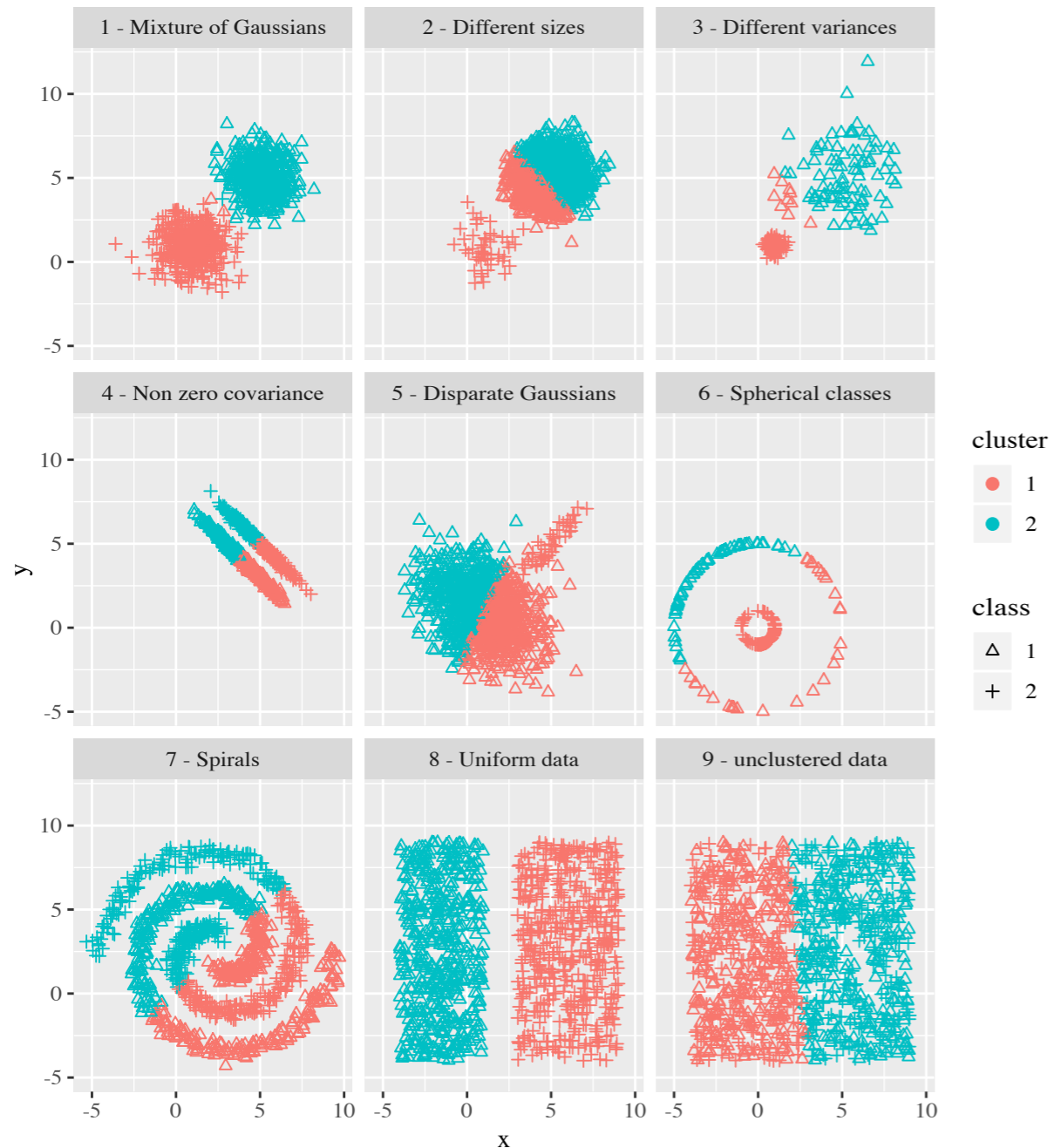
$$p(X) = \sum_{k=1}^K \pi_k G(X | \mu_k, \sigma_k)$$

$$\arg \max_{\Theta} p(X | \Theta)$$

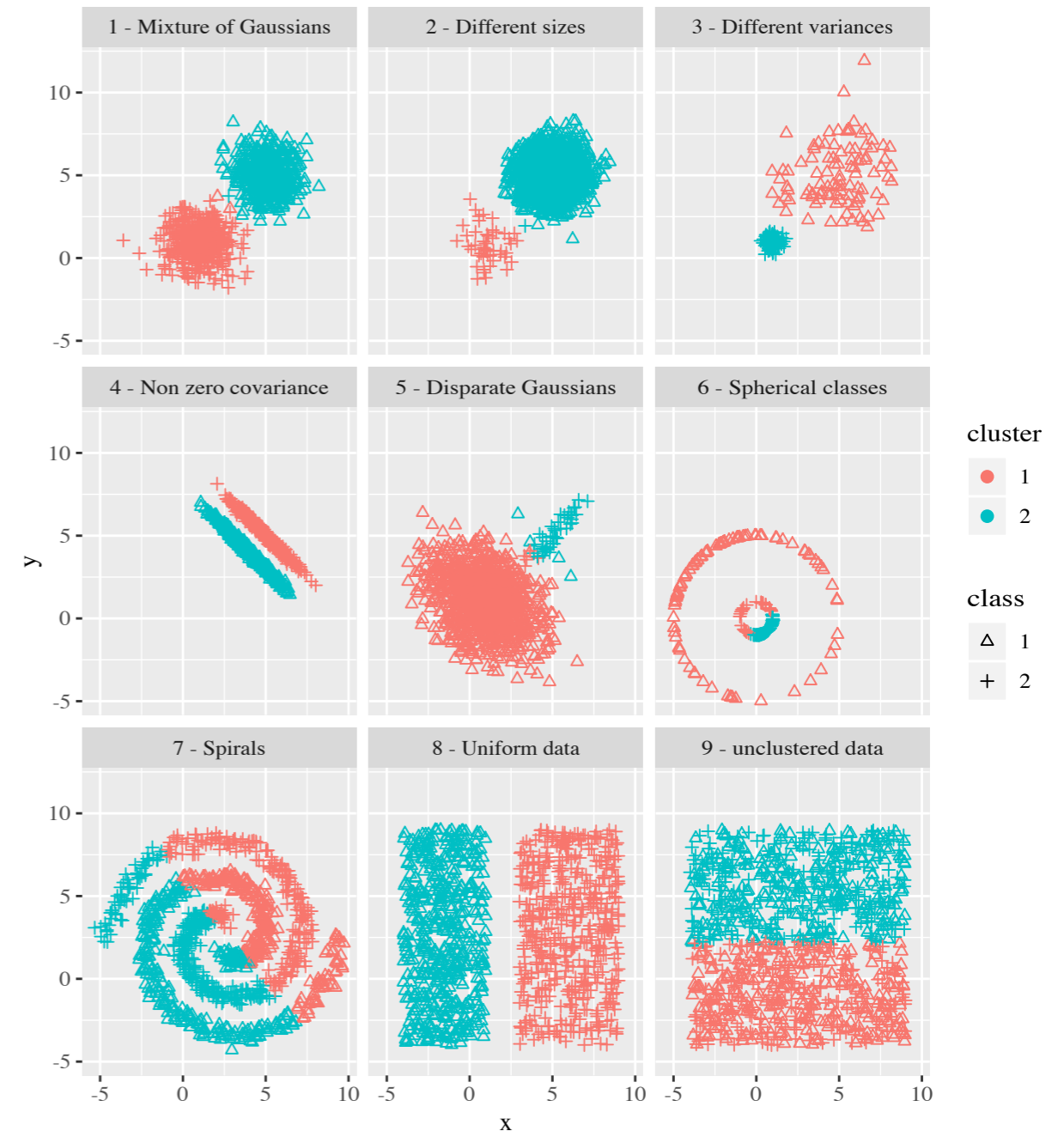
$$\Theta = \mu, \sigma, \pi$$

K-MEANS COMPARISON

K-means



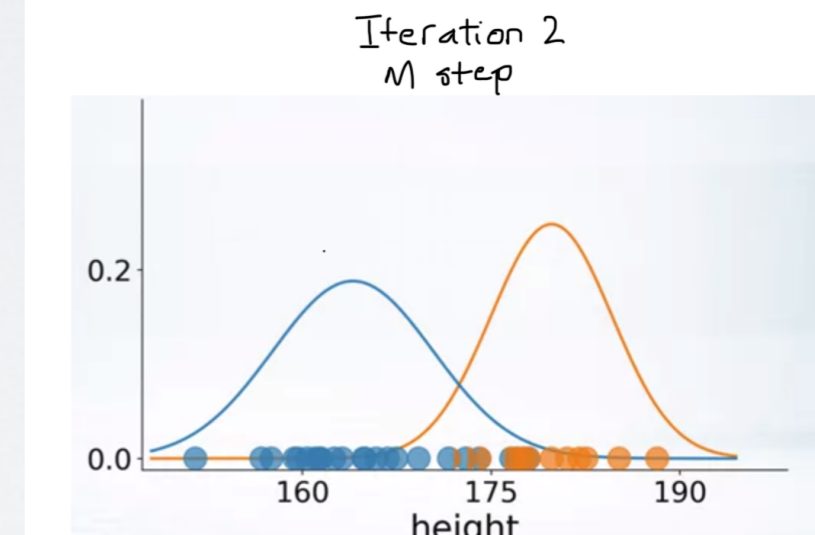
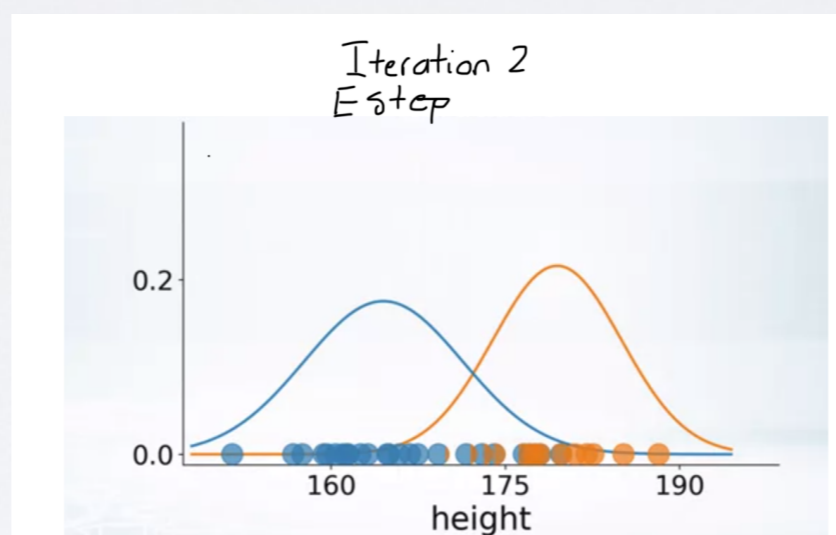
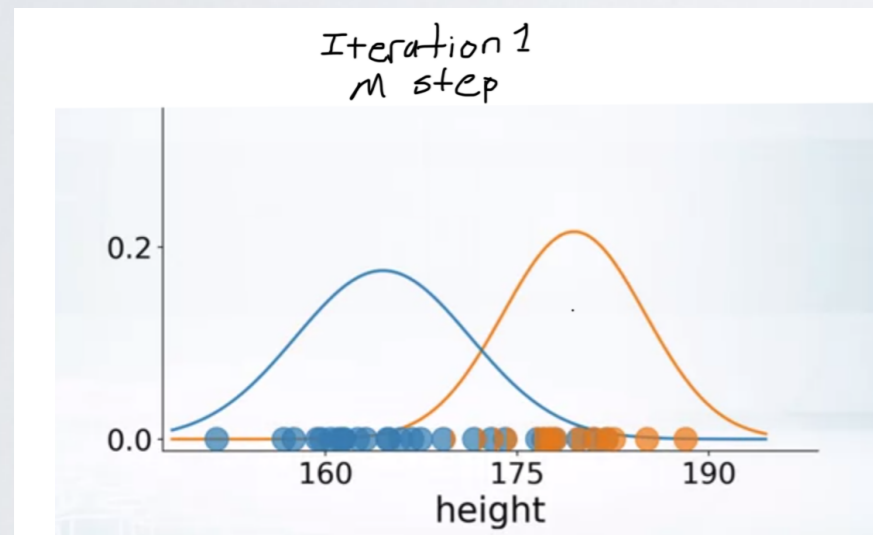
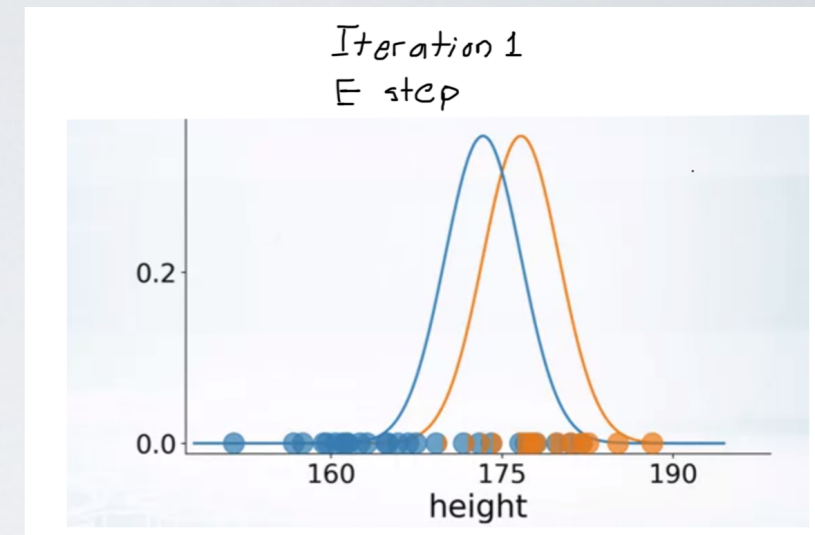
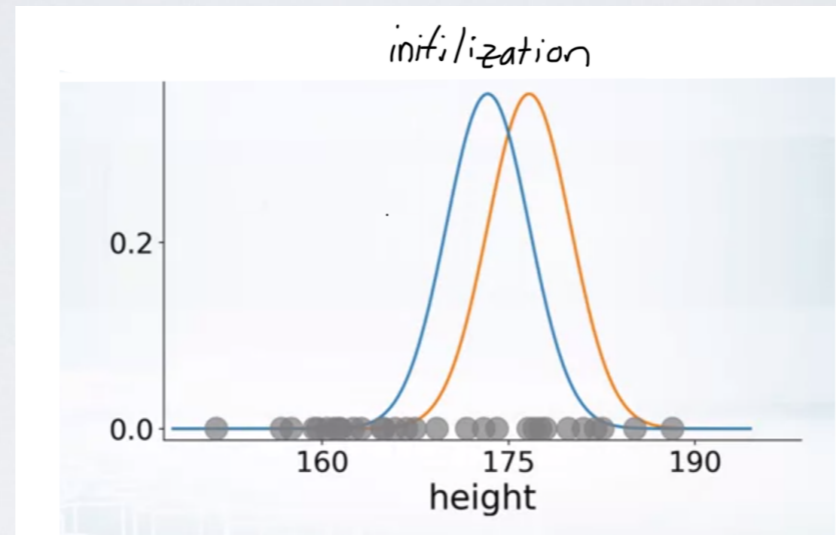
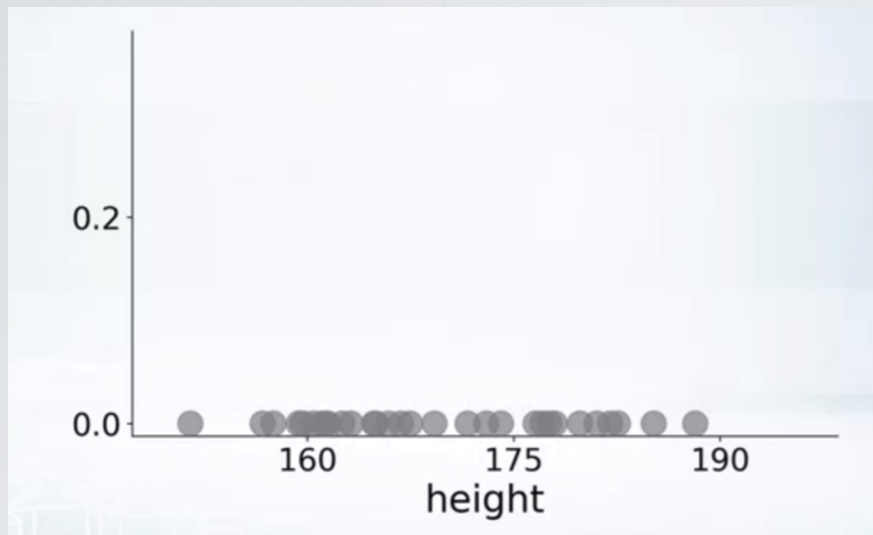
Full Gaussian Mixture



EM ALGORITHM

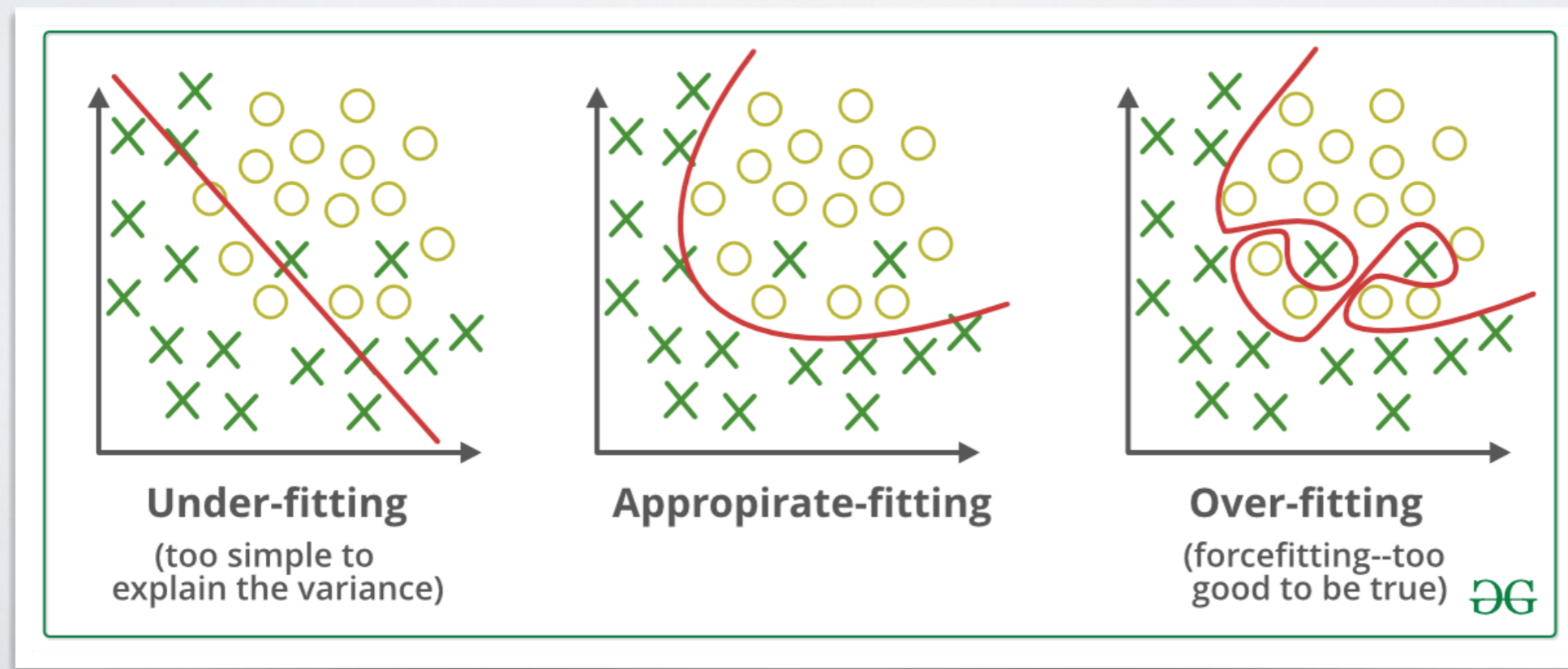
- To search for the parameters, we can use a method similar to naive k-means known as EM (Expectation Maximization)
 - Note Z the cluster assignation of items to their **most likely** clusters
 - 1) Initialize parameters Θ to random values
 - 2)(E) Compute Z , given Θ
 - 3)(M) Use assignations in Z to update values of Θ
 - 4) Iterate steps 2 and 3 until convergence

EM ALGORITHM



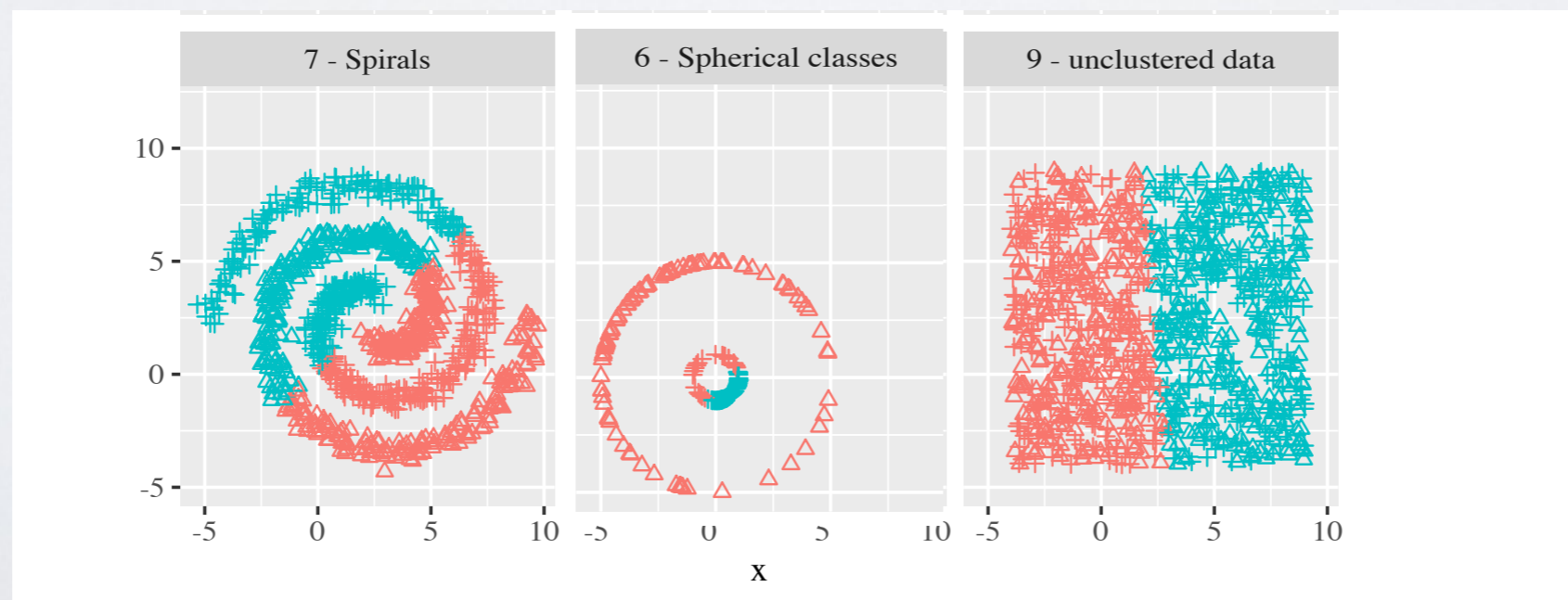
PROS AND CONS

- Gaussian mixture seems an improvement over k-means. Why not always using it?
 - Force of habits
 - Higher computational cost (More parameters => More complex problem)
 - Higher possibility of overfitting (More parameters => More overfit risk)



REMAINING PROBLEMS

- We can mention 3 problems remaining (at least)
 - The number of clusters still needs to be provided.
 - If allowed to change, it will always converge to the trivial solution with each item in its own cluster
 - If the data is completely random, the method still finds clusters
 - Impossible to discover non-convex structures, such as circles or spirals

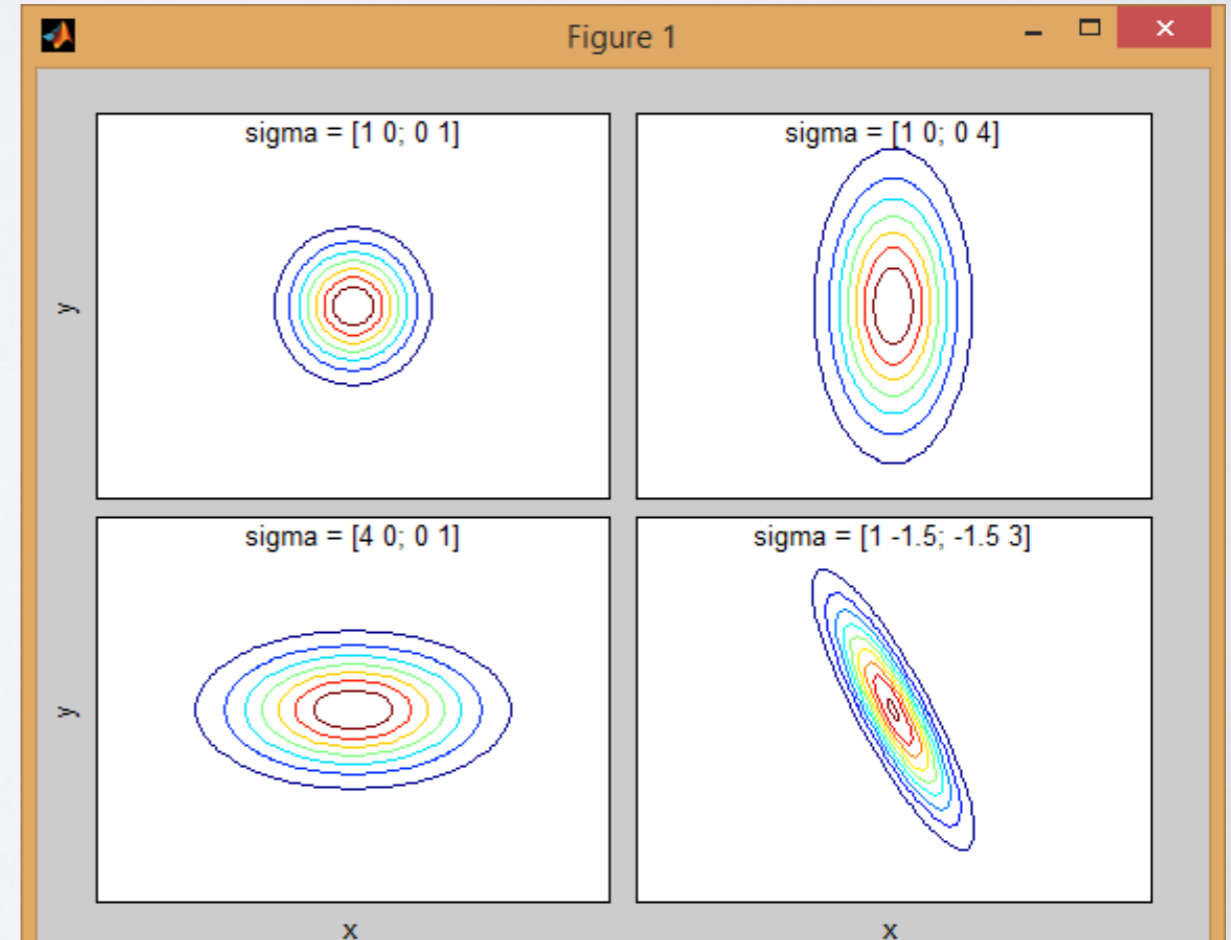


MDL

- Discovering automatically the number of clusters —and thus finding no clusters in random data— is possible using an MDL approach
- MDL = Minimum Description Length
- The principle is to search a solution maximizing the compression rate, i.e., minimizing the *cost* of the description, e.g., in bits.
- https://en.wikipedia.org/wiki/Minimum_description_length

NORMALIZATION

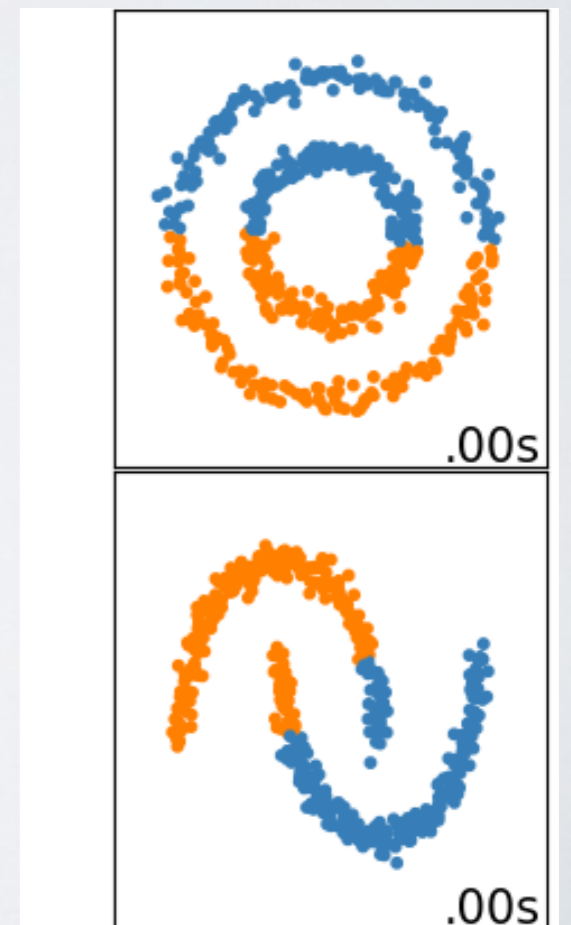
- Is normalization as important for full GM models as for k-means?



DBSCAN

K-MEANS/GM LIMITS

- The problem of spiral/Circular/weird shaped clusters comes from the assumption that items of a cluster should be “normally distributed” around their mean



LOCAL DEFINITIONS

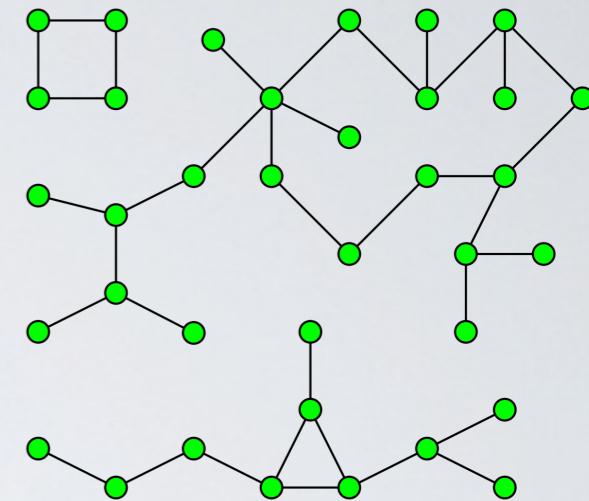
- To overcome this problem, several methods propose local definitions of clusters
 - ▶ Does not explicitly optimize a global function
 - ▶ Items belong to clusters because they are close enough, locally, to other items in that cluster
 - ▶ Clusters exist because there is continuum between all items in it, locally

DBSCAN

- Define some local parameters:
 - ▶ ϵ , the distance threshold above which items are considered “too different”
 - ▶ *minPts*, a minimal number of reachable points
 - ▶ No need to define a number of clusters !
- Define:
 - ▶ An item p is a *core point* if it has at least *minPts* items at distance less than ϵ
 - Including p itself

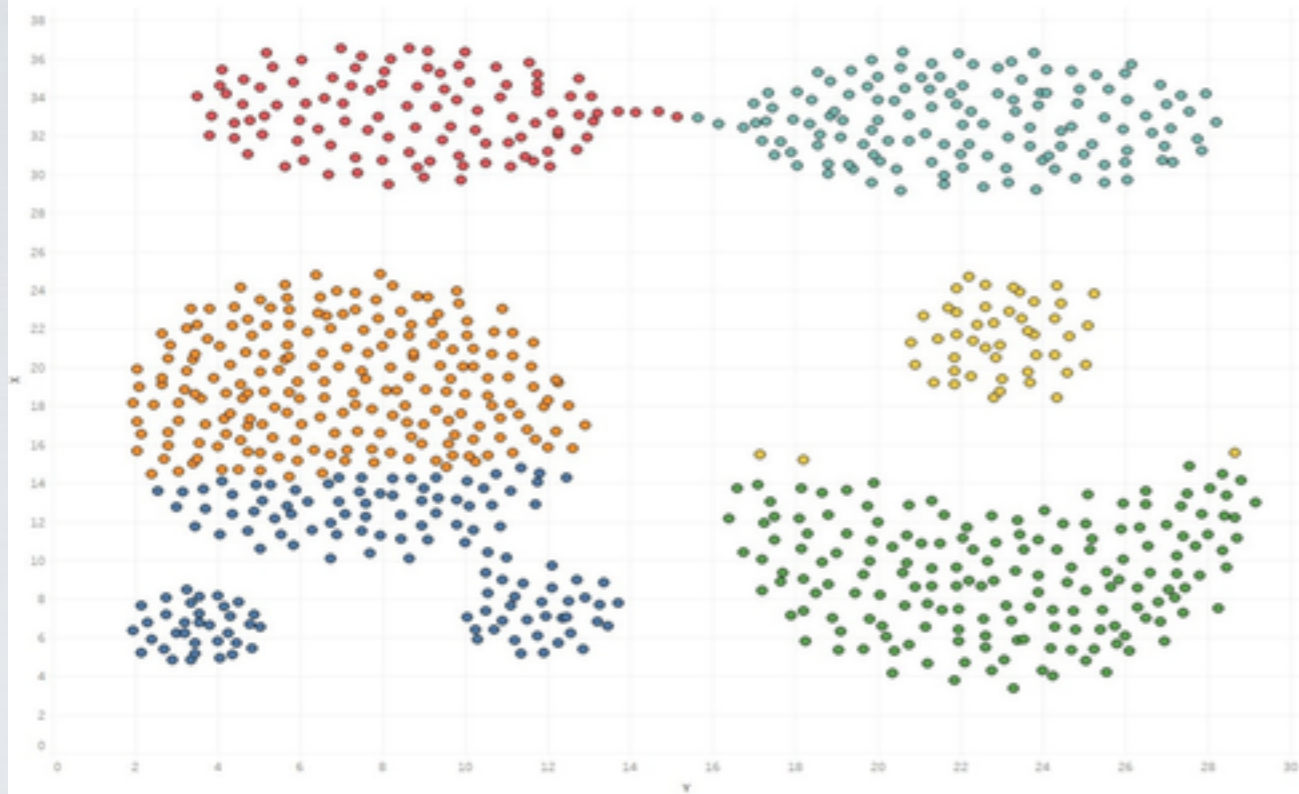
DBSCAN: GRAPH DEFINITION

- 1) Build a graph such as
 - Each core node is a node
 - A link exist between core nodes if they are at $d < \epsilon$
- 2) Detect the connected components of the graph
 - 2 nodes belong to the same connected components if there is a path between them
- 3) For all non-core nodes:
 - If they have no core points directly reachable, discard them as noise
 - Else, attribute them to (one of) the clusters for which one core point is directly reachable
 - Variant DBSCAN* => ignore those points as noise

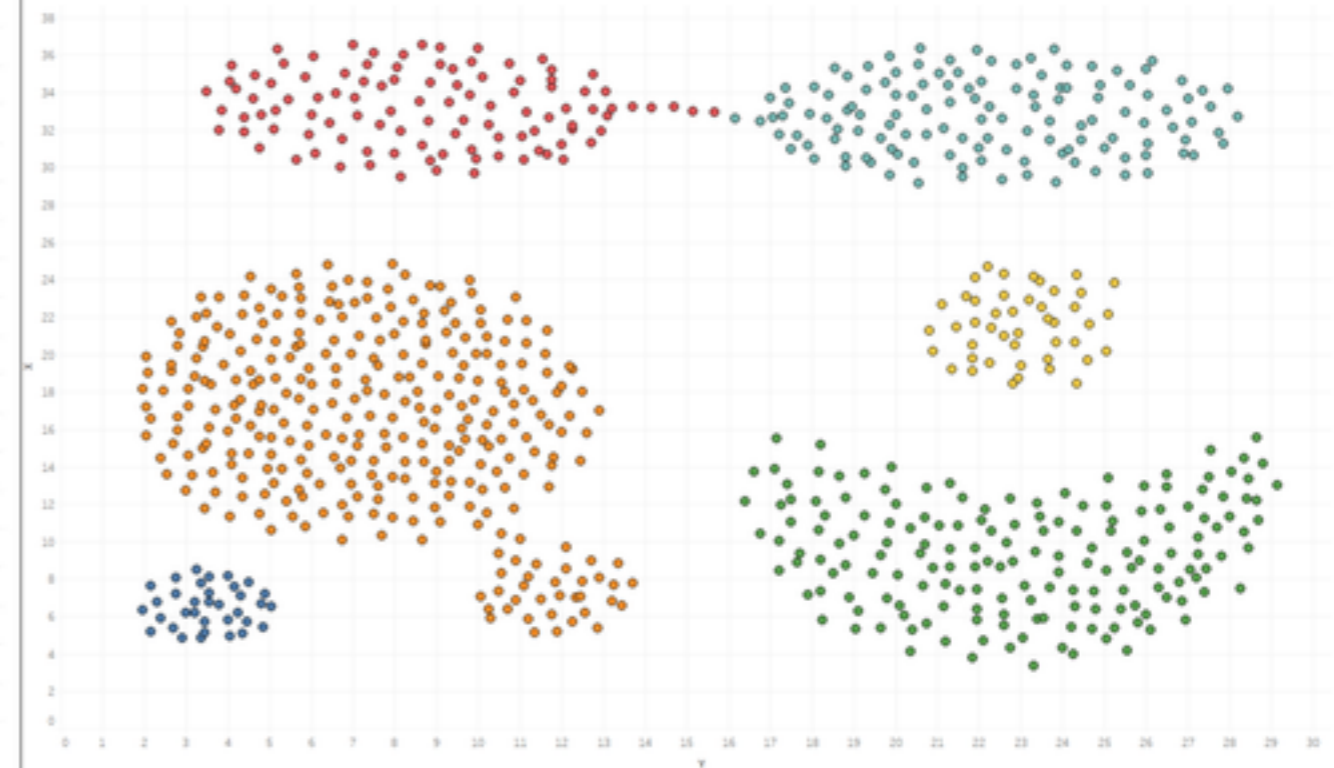


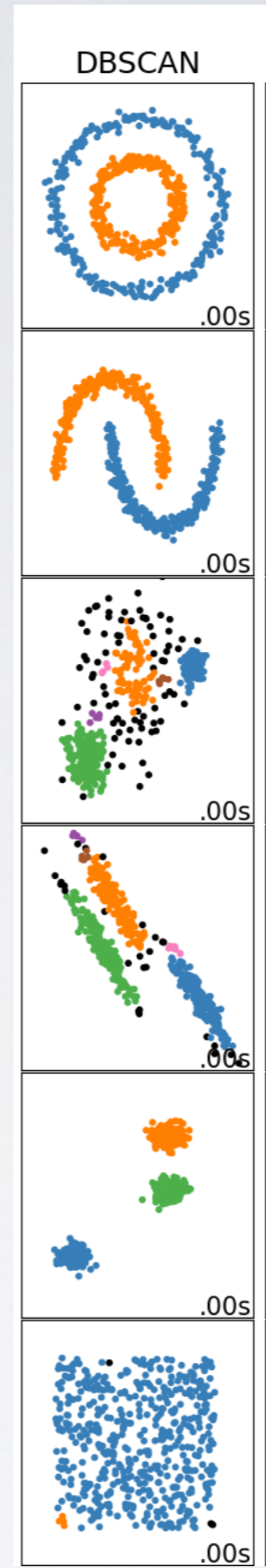
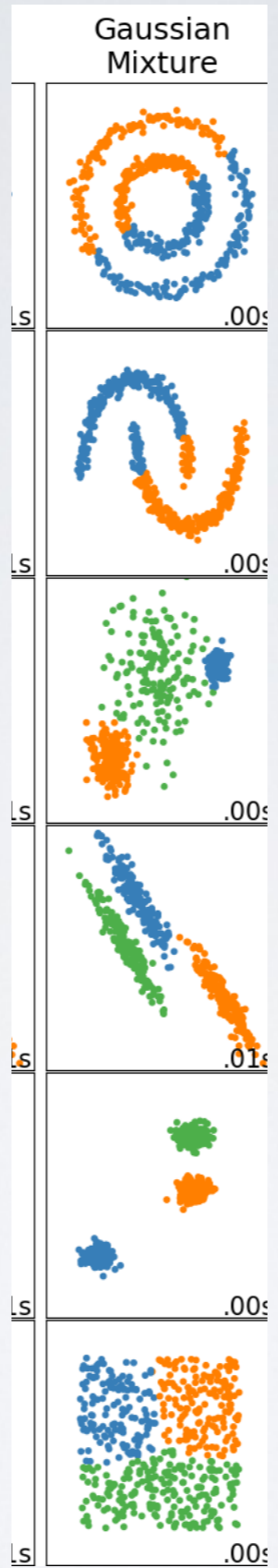
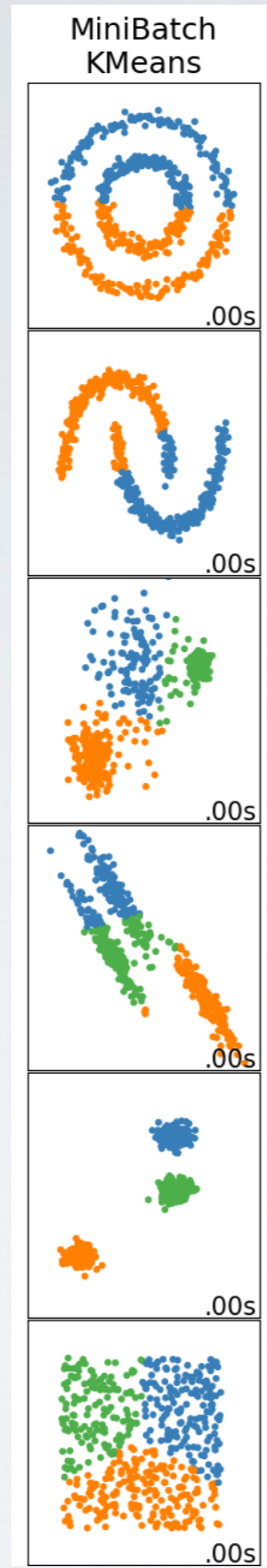
DBSCAN

Traditional Clustering (K-means)



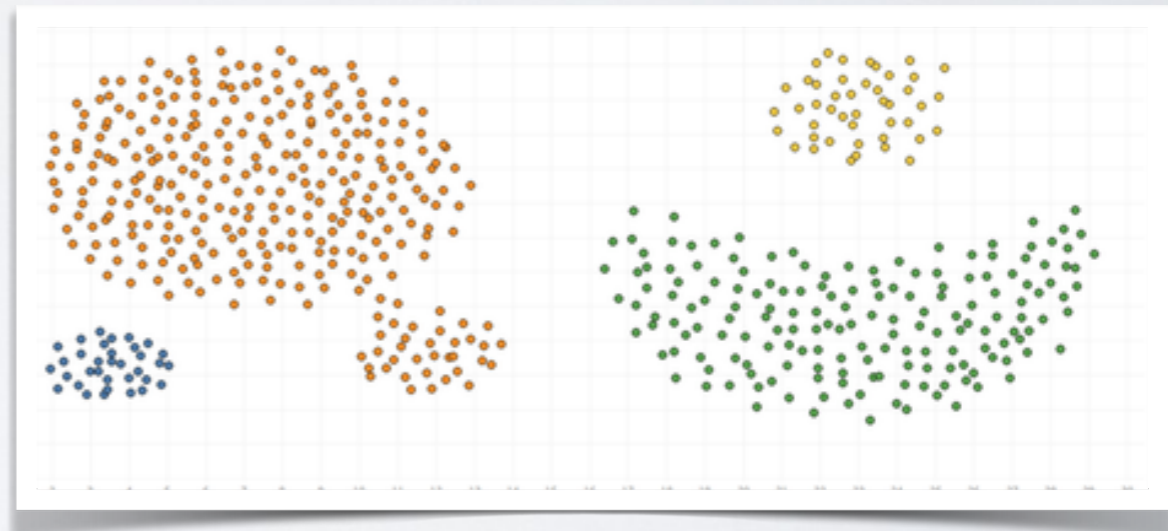
DBSCAN





DBSCAN

- Strength:
 - ▶ No need to define the number of clusters
 - ▶ Can discover arbitrarily-shaped clusters
 - ▶ A notion of noise
- Weaknesses
 - ▶ Defining ϵ is extremely difficult
 - Similar to the number of clusters.
 - In fact it determines the number of clusters...
 - ▶ Despite safeguards, risk of the stretched clusters effect



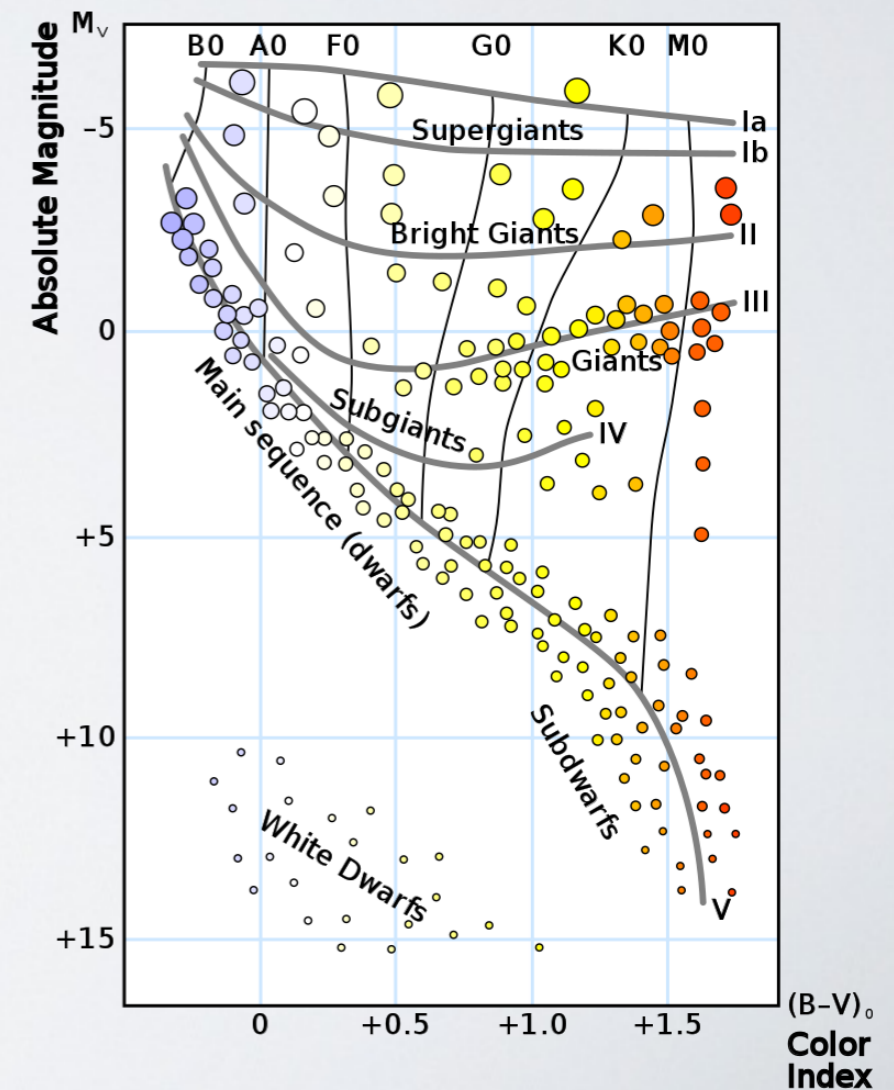
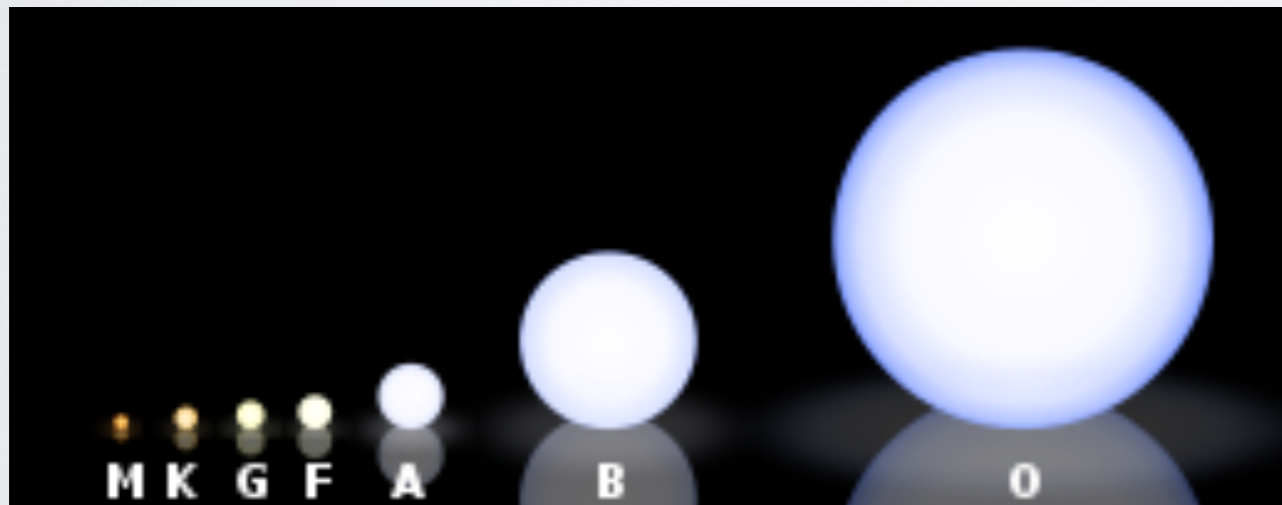
CLUSTERING EVALUATION

INTERNAL/EXTERNAL

- Two types of evaluation: internal or external
- **External** Evaluation (extrinsic):
 - ▶ Similarly to supervised learning, compares the clusters found with a “ground truth”
 - ▶ The ground truth can be exactly the right clustering desired
 - So we are just validating the method, since we already know the answer...
 - ▶ The ground truth can be a proxy to what we want
 - e.g., we have a manual ground truth, done by an expert. Not perfect, costly, and not generalizable to newer data, so supervised cannot work. We can check that clustering find something close.

INTERNAL/EXTERNAL

Class	Effective temperature ^{[2][3]}	Vega-relative chromaticity ^{[4][5][a]}	Chromaticity (D65) ^{[6][7][4][b]}	Main-sequence mass ^{[2][8]} (solar masses)	Main-sequence radius ^{[2][8]} (solar radii)	Main-sequence luminosity ^{[2][8]} (bolometric)	Hydrogen lines	Fraction of all main-sequence stars ^[9]
O	≥ 30,000 K	blue	blue	≥ 16 M_{\odot}	≥ 6.6 R_{\odot}	≥ 30,000 L_{\odot}	Weak	~0.00003%
B	10,000–30,000 K	blue white	deep blue white	2.1–16 M_{\odot}	1.8–6.6 R_{\odot}	25–30,000 L_{\odot}	Medium	0.13%
A	7,500–10,000 K	white	blue white	1.4–2.1 M_{\odot}	1.4–1.8 R_{\odot}	5–25 L_{\odot}	Strong	0.6%
F	6,000–7,500 K	yellow white	white	1.04–1.4 M_{\odot}	1.15–1.4 R_{\odot}	1.5–5 L_{\odot}	Medium	3%
G	5,200–6,000 K	yellow	yellowish white	0.8–1.04 M_{\odot}	0.96–1.15 R_{\odot}	0.6–1.5 L_{\odot}	Weak	7.6%
K	3,700–5,200 K	light orange	pale yellow orange	0.45–0.8 M_{\odot}	0.7–0.96 R_{\odot}	0.08–0.6 L_{\odot}	Very weak	12.1%
M	2,400–3,700 K	orange red	light orange red	0.08–0.45 M_{\odot}	≤ 0.7 R_{\odot}	≤ 0.08 L_{\odot}	Very weak	76.45%



INTERNAL/EXTERNAL

- Two types of evaluation: internal or external
- **Internal** Evaluation (Intrinsic):
 - ▶ We have no ground truth to compare to
 - ▶ We evaluate the intrinsic properties of our clusters, typically
 - If their elements are similar
 - If clusters are far apart /if elements in different clusters are different.

INTERNAL EVALUATION

AD-HOC SCORES

- Several clustering methods define their own objective to minimize. This objective can be used as a score for clusters obtained by this method or others
 - k-means minimizes inter-cluster variance
 - Gaussian mixture maximizes the likelihood
- But can lead to unfair comparisons:
 - Using inter-cluster variance to compare k-means and another method such as DBscan is unfair.
 - One explicitly minimizes this objective, the other no...
- The choice of a score is equivalent to choosing a definition of cluster...

SILHOUETTE SCORE

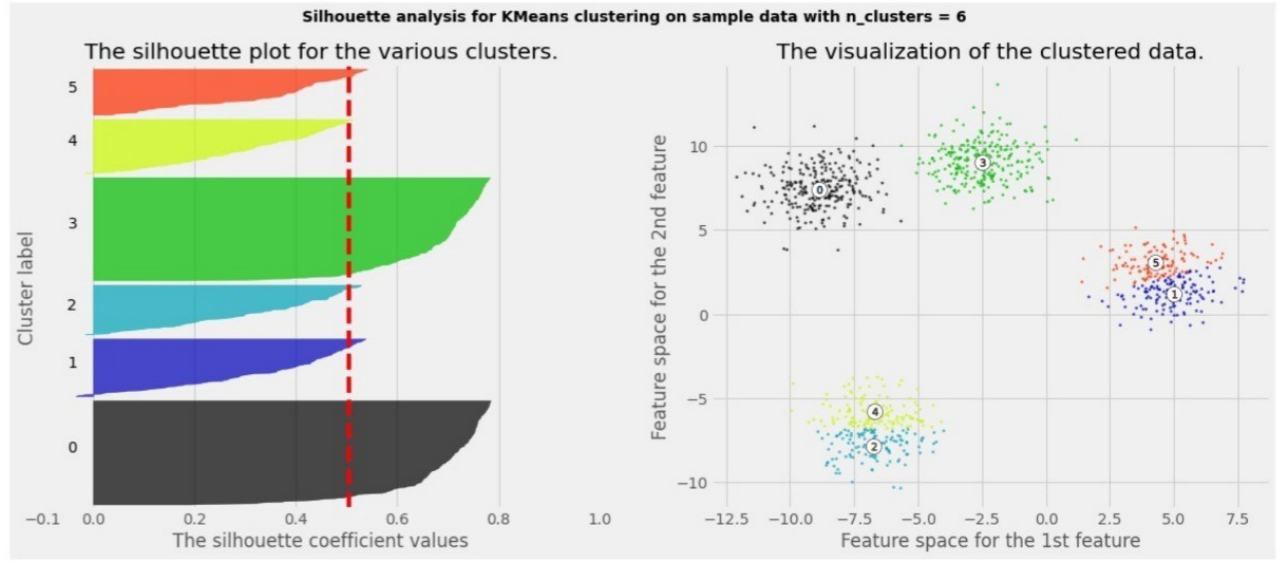
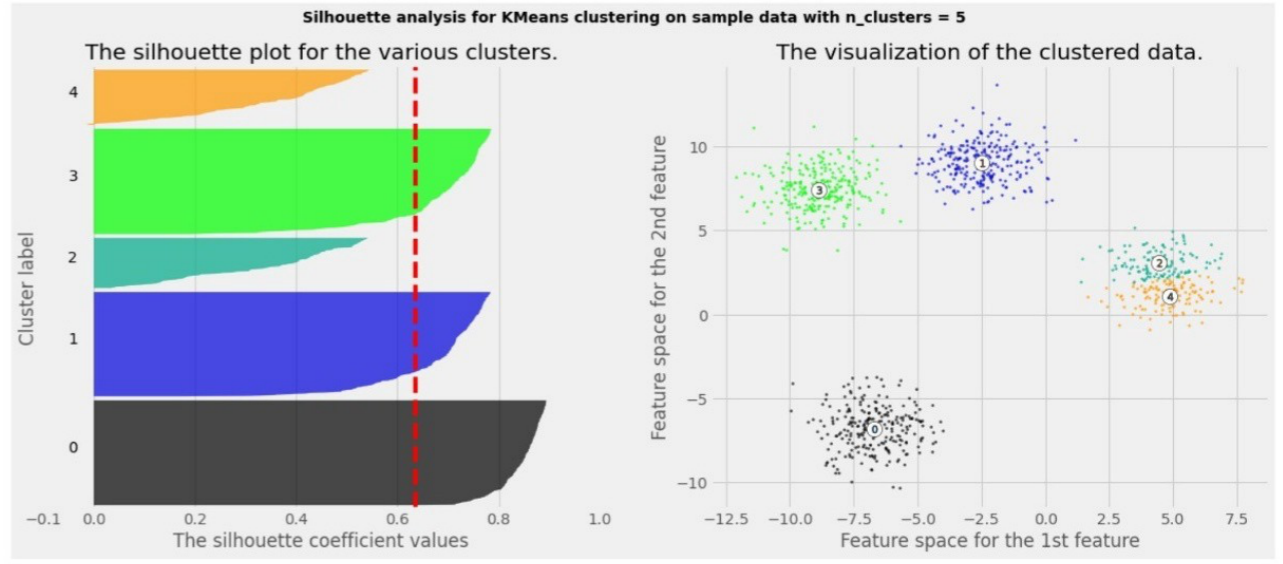
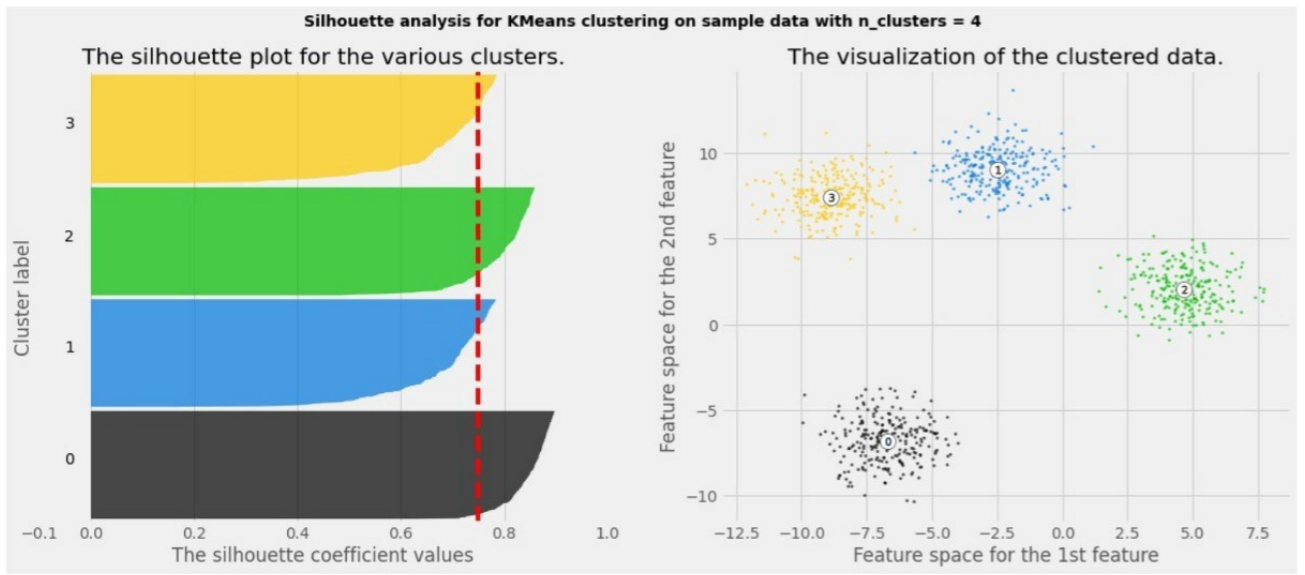
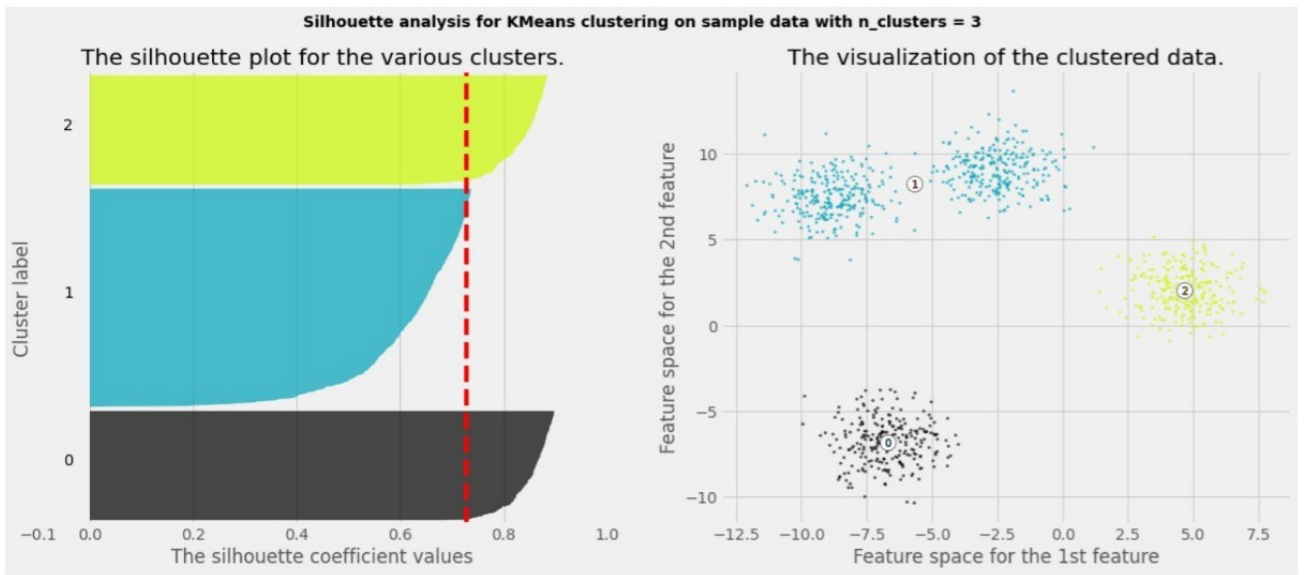
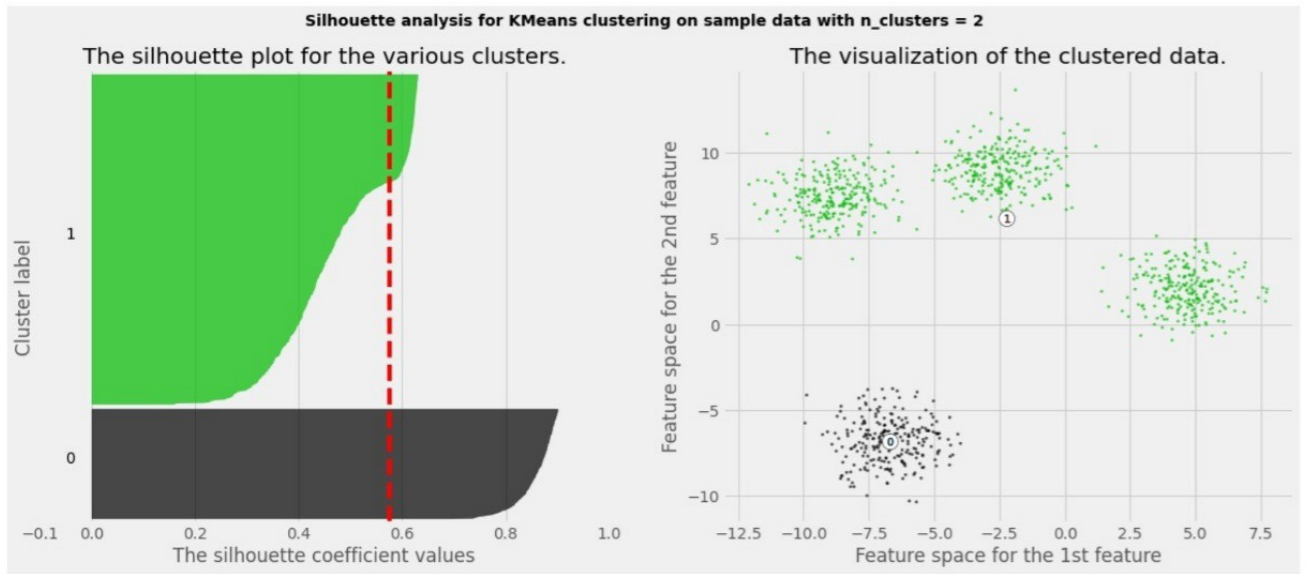
- Silhouette score of 1 observation:

- ▶ 1) Compute $a(i)$, average distance to all other observations of the same cluster
- ▶ 2) Compute $b(i)$, min of “average distance to all observations of another cluster”

- ▶ 3) Silhouette: $s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$

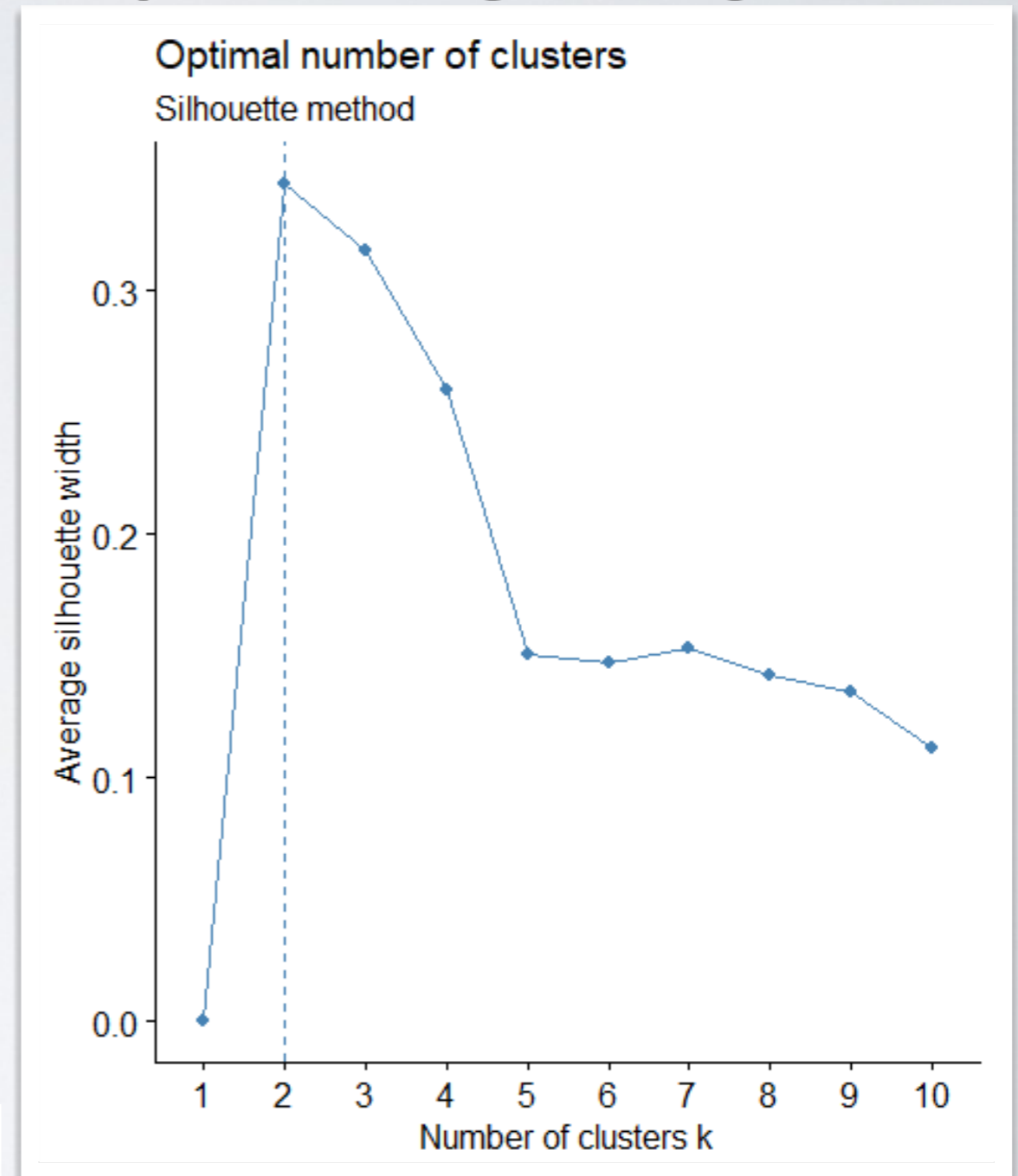
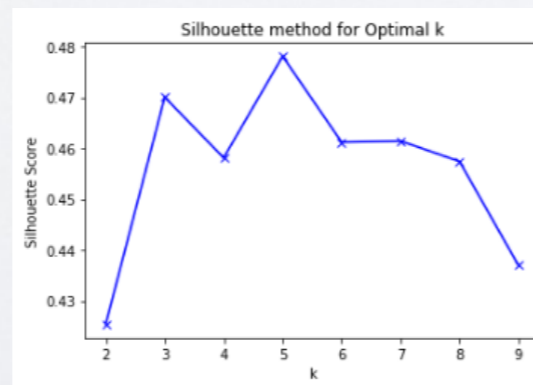
- Silhouette coefficient:

- ▶ Average of all individual Silhouette scores.



AUTOMATIC K SELECTION

- The Silhouette score can be used to choose automatically the number of clusters:
 - We vary the number of clusters k , and search for the maximum



AUTOMATIC K SELECTION

- Better than the elbow method on real data

OTHER SCORE FUNCTIONS

- **Davies-Bouldin Index (DBI):** The average similarity ratio of each cluster with its most similar cluster,
 - ▶ where similarity is the ratio of within-cluster distances to between-cluster distances;
 - ▶ lower DBI values suggest better clustering.

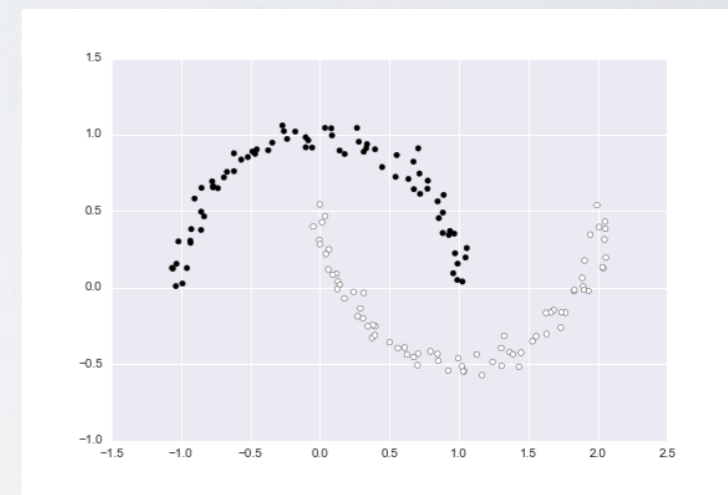
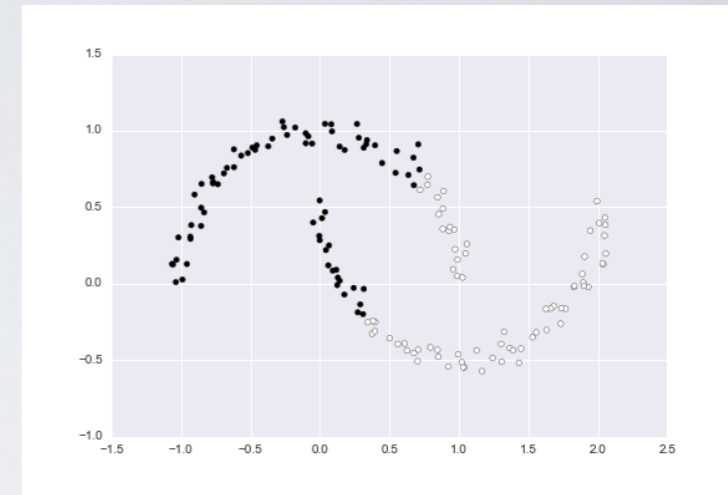
DUNN INDEX

$$DI_m = \frac{\min_{1 \leq i < j \leq m} \delta(C_i, C_j)}{\max_{1 \leq k \leq m} \Delta_k}$$

- With
 - $\delta(C_i, C_j)$ a measure of distance between clusters
 - e.g., distance between closest points, average distance...
 - Δ_k a measure of the dispersion of the cluster
 - e.g., max distance between two cluster points

NON-SPHERICAL CLUSTERS

- Remember the difference between k-means clusters and DB-scan clusters
- Previous scores are reliable only in k-means-like clusters.
- Specific (less known) scores for arbitrary clusters
 - Density-based silhouette
 - DBCV(Density-Based Clustering Validation)



STABILITY

- If clusters are not clear, multiple runs of the same method might discover different clusters
- Evaluating the stability of those clusters might be a way to assess their quality
- To better assess the quality, one can introduce noise:
 - Comparing clustering on sub-sets (random samples, independent samples...)
 - Adding noise (fake data points, outliers, removing low-quality data...)

CONSENSUS CLUSTERING

- Let's consider that we have multiple candidate clusterings
 - ▶ From the same method ran multiple times
 - ▶ From the same method with different parameters
 - ▶ From different methods
- One can compute a “consensus”
 - ▶ Create the consensus matrix C_{ij} counts the number of times data points i, j were grouped together
 - ▶ Apply your favorite clustering method on that matrix, considering that $\frac{1}{C_{ij}}$ gives the *distance* between data points.

MANY OTHER CLUSTERINGS

- Hierarchical clustering
- Spectral clustering
- Mean-Shift clustering
- Affinity Propagation
- OPTICS (Ordering Points To Identify the Clustering Structure)

NO FREE LUNCH THEOREM

- “Any two optimization algorithms are equivalent when their performance is averaged across all possible problems”
 - ▶ Two clustering algorithms with different objective functions are fully comparable, one is not intrinsically better than another.
 - ▶ Each is the best for the objective function it defines
 - ▶ What is “the best” cluster? Depends on your definition.
- Does not mean that some methods are not more appropriate than other for what most people consider as clusters...