

# Машинное обучение

Лекция 9  
Кластеризация

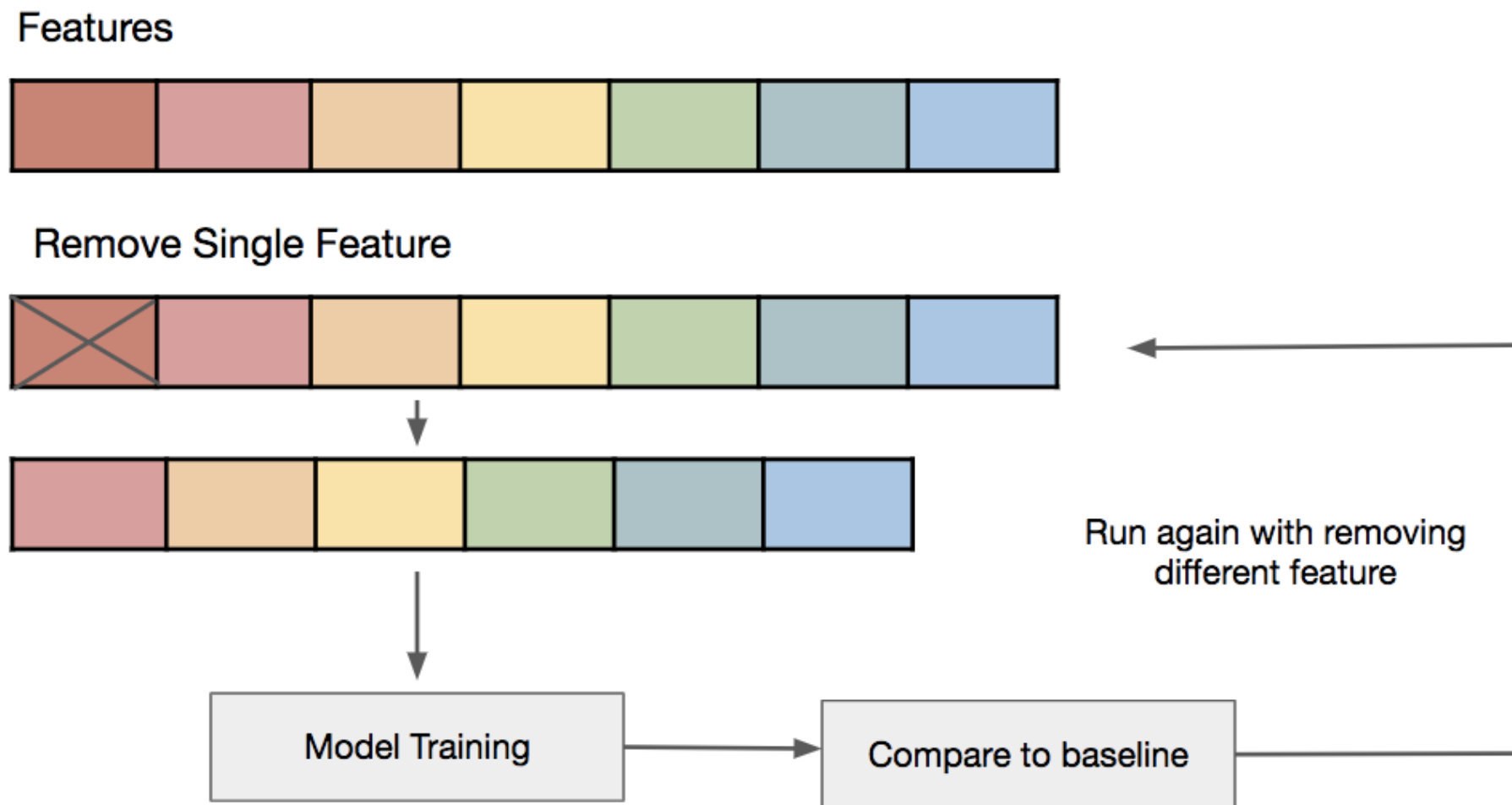
Михаил Гуцин  
[mhushchyn@hse.ru](mailto:mhushchyn@hse.ru)

НИУ ВШЭ, 2022



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# На прошлой лекции



# План

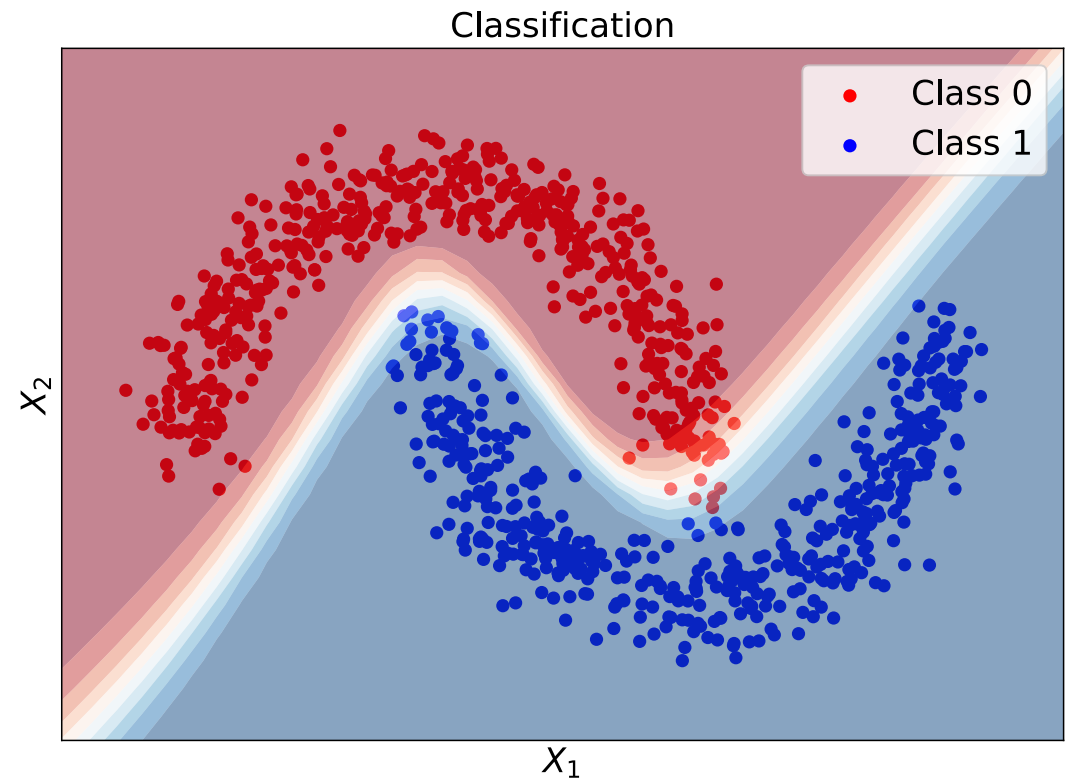
- ▶ Clustering
- ▶ K-Means algorithm
- ▶ Quality metrics
- ▶ Hierarchical clustering
- ▶ DBSCAN

# Clustering



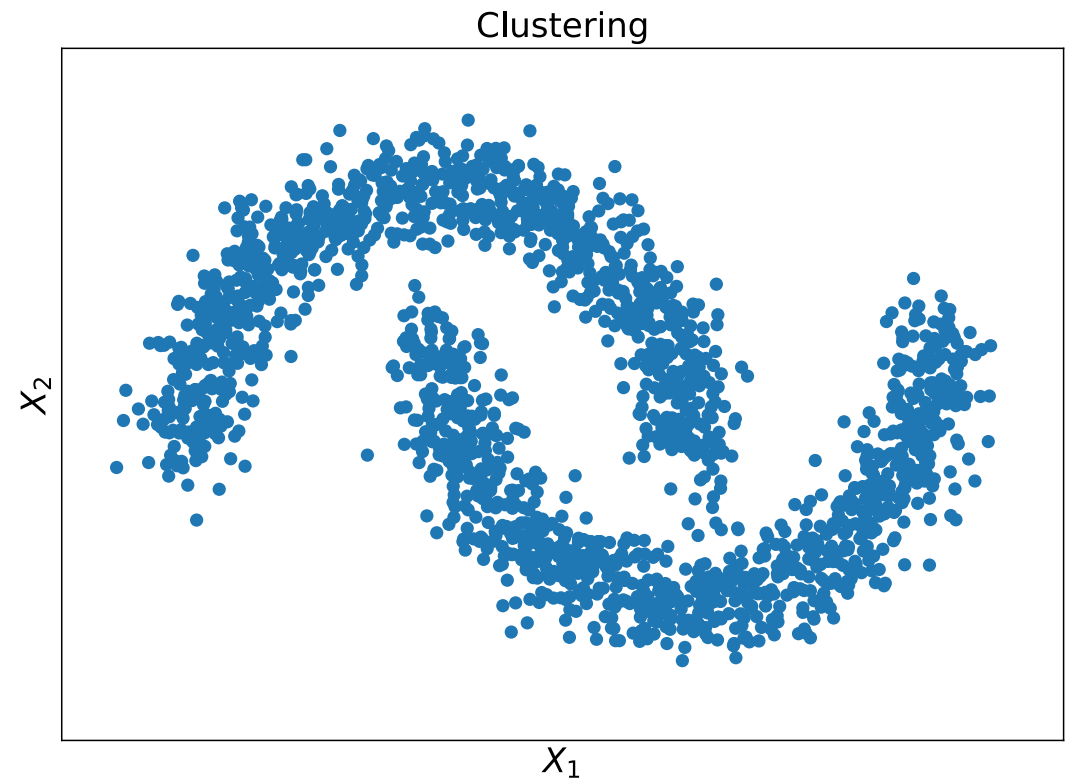
# Clustering vs classification

- ▶ In classification, we have object features  $X$  and class labels  $y \in \{0, 1\}$
- ▶ A classifier learns decision rule  $f$ , so that  $f(X) \approx y$
- ▶ The trained classifier predicts class labels for new objects

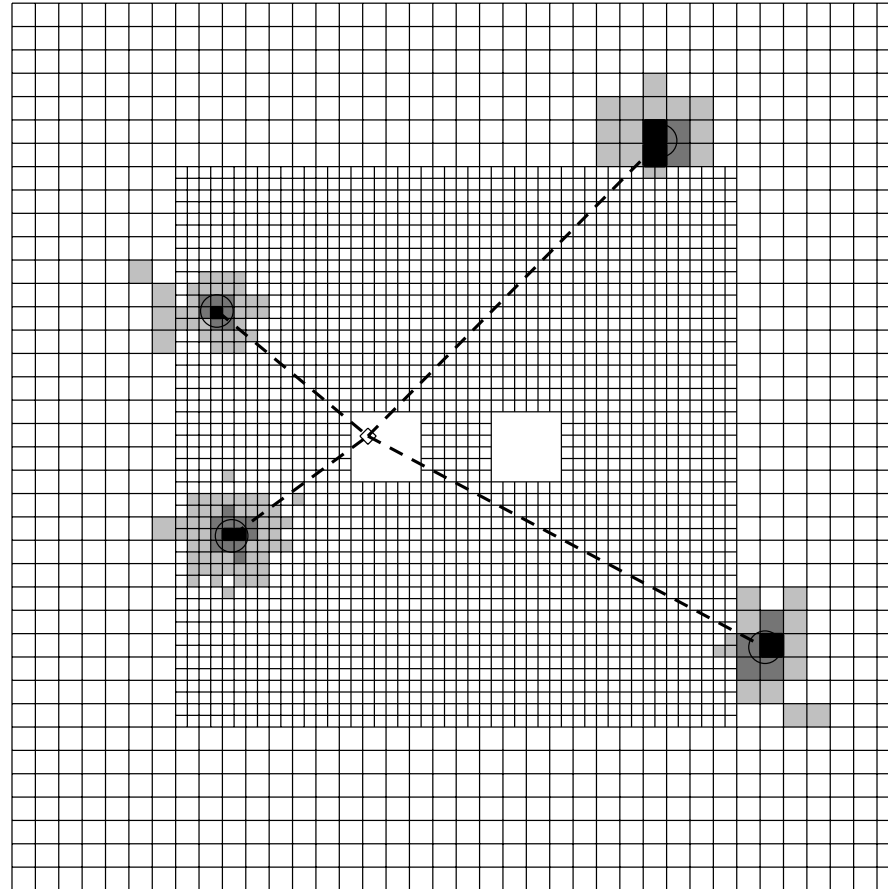


# Clustering vs classification

- ▶ In clustering, we don't have class labels  $y$
- ▶ The goal is to divide all objects into separate groups using only object features  $X$
- ▶ Objects inside groups are similar
- ▶ Objects from different groups are dissimilar



# Example of clustering



Clusters in EM calorimeter of KTEV experiment for  $K \rightarrow \pi^0 \pi^0$  decay.

# Clustering assumptions

Most of clustering algorithms are based on the following assumptions:

- ▶ Objects form dense clusters
- ▶ Objects from one cluster are similar
- ▶ Objects from different clusters are dissimilar
- ▶ Objects similarity is often based on distance between them
- ▶ Distances between neighbors within one cluster are smaller than between objects from different clusters

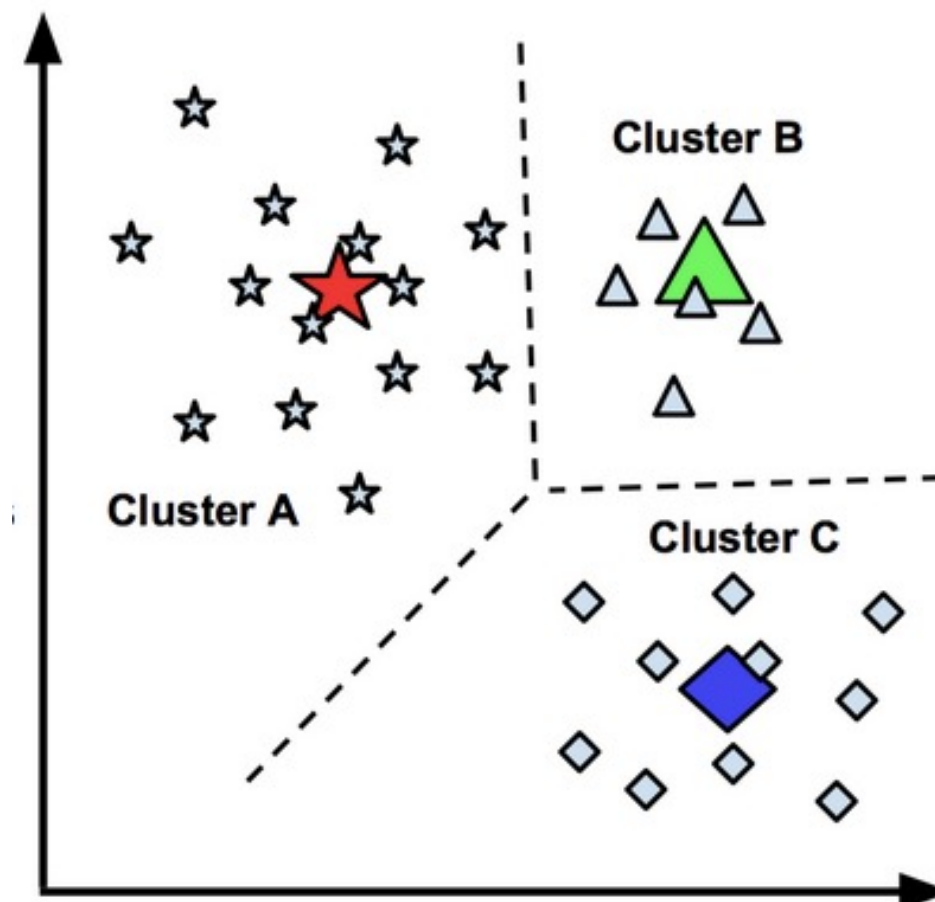


# K-Means



# Clustering intuition

- ▶ Each cluster is represented by its center
- ▶ All objects are assigned to the closest center
- ▶ The goal is to find such centers that form the most compact clusters



Link: <https://medium.com/@msdasila90/basics-k-means-clustering-algorithm-a77c539c9e00>

# Notations

- ▶ Consider a sample with  $N$  objects  $\{x_n\}_{n=1}^N$ .
- ▶ We will search for  $K$  clusters with centers  $\{\mu_1, \mu_2, \dots, \mu_K\}$ .
- ▶ Criterion to find the best centers is minimum of **within-cluster distance**:

$$Q = \sum_{n=1}^N \min_k \rho(x_n, \mu_k) \rightarrow \min_{\mu_1, \dots, \mu_K}$$

- ▶ Each object  $x_n$  is assigned to a cluster  $z_n \in \{1, 2, \dots, K\}$  as:

$$z_n = \arg \min_k \rho(x_n, \mu_k)$$

# General algorithm

```
initialize  $\mu_1, \dots, \mu_K$  from  
random training objects
```

```
WHILE not converged:
```

```
  FOR  $n = 1, 2, \dots, N$ :
```

```
     $z_n = \arg \min_k \rho(x_n, \mu_k)$  ← Assign each object to the  
                                         nearest center
```

```
  FOR  $k = 1, 2, \dots, K$ :
```

```
     $\mu_k = \arg \min_{\mu} \sum_{n: z_n = k} \rho(x_n, \mu)$  ← Update the centers
```

```
RETURN  $z_1, \dots, z_N$ 
```

# Algorithm variations

- ▶ Distance  $\rho(x_n, \mu_k)$  can be defined in different ways.
- ▶ If  $\rho(x_n, \mu_k) = \|x_n - \mu_k\|_2^2$ , we get **K-Means algorithm**
- ▶ If  $\rho(x_n, \mu_k) = \|x_n - \mu_k\|_1$ , we get **K-Medians algorithm**

# K-Means algorithm

Initialize  $\mu_j$ ,  $j = 1, 2, \dots, K$ .

**WHILE** not converged:

**FOR**  $i = 1, 2, \dots, N$ :

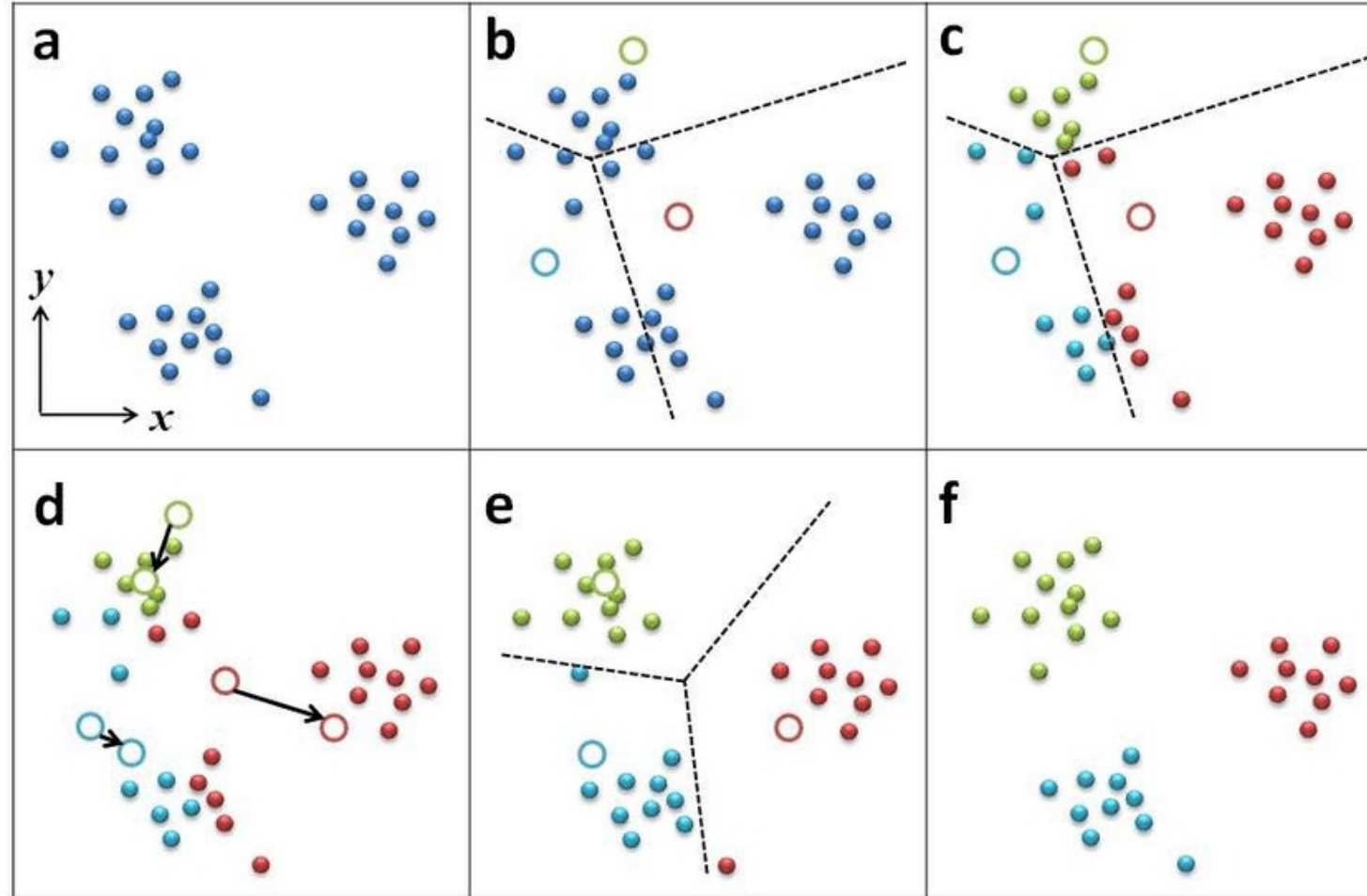
find cluster number of  $x_i$ :

$$z_i = \arg \min_{j \in \{1, 2, \dots, K\}} \|x_i - \mu_j\|_2^2$$

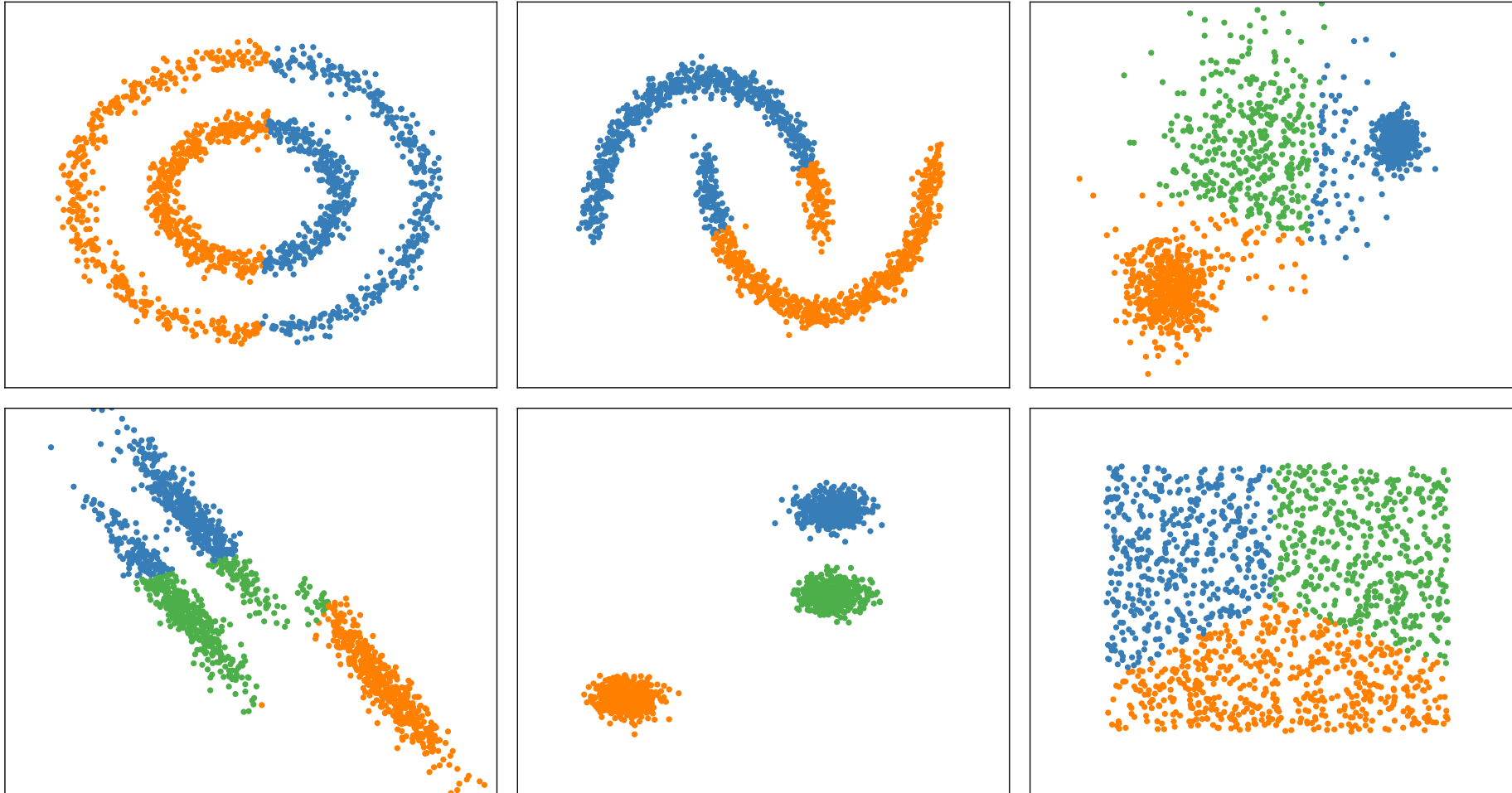
**FOR**  $j = 1, 2, \dots, K$ :

$$\mu_j = \frac{1}{\sum_{n=1}^N \mathbb{I}[z_n = j]} \sum_{n=1}^N \mathbb{I}[z_n = j] x_i$$

# K-Means demonstration



# K-Means examples





# Properties #1

- ▶ Initialization:
  - Centers  $\{\mu_k\}_{k=1}^K$  are usually initialized randomly from training objects
  - Number of clusters (and centers)  $K$  is fixed
- ▶ Convergence criteria:
  - Iterations limit is reached
  - Centers stop changing significantly
  - Cluster assignments  $\{z_n\}_{n=1}^N$  stop changing

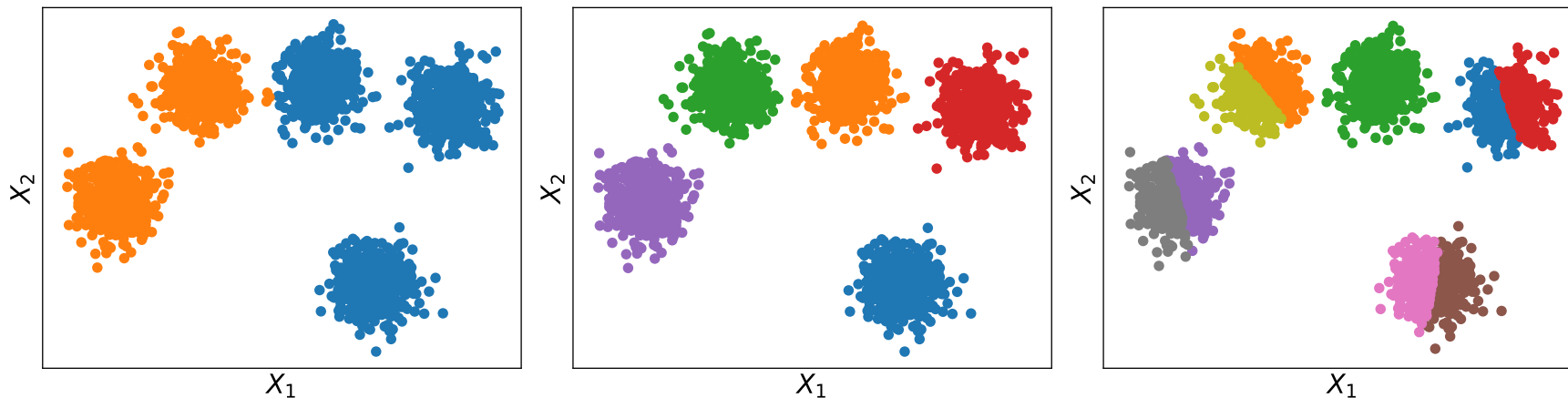
# Properties #2

- ▶ Solution
  - Depends on starting positions of centers
  - Sensitive to outliers, may create single-object clusters
  - It is recommended to run the algorithm with several different initializations and select solution with the minimal within-cluster distance  $Q$

# Elbow method

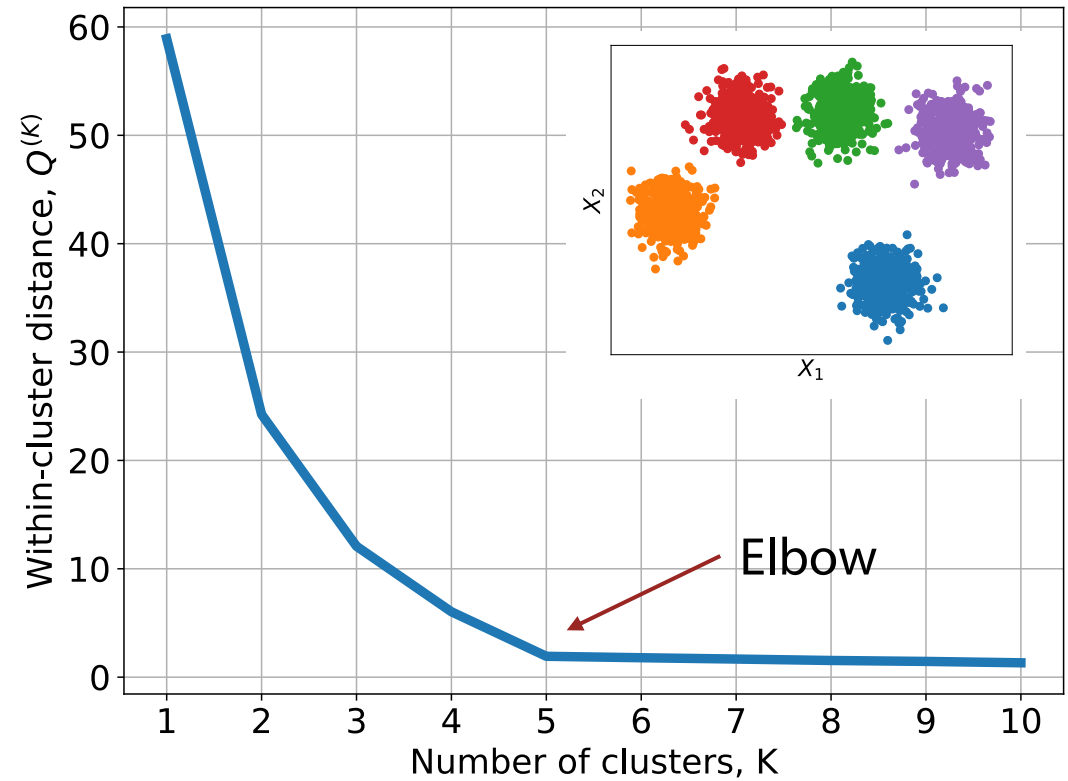
- ▶ How to estimate optimal number of clusters  $K$ ?
- ▶ Consider within-cluster distances  $Q^{(K)}$  for all possible  $K$ :

$$Q^{(K)} = \sum_{n=1}^N \|x_n - \mu_{z_n}\|_2^2 \rightarrow \min_{z_1, \dots, z_N, \mu_1, \dots, \mu_K}$$



# Elbow method

- ▶  $Q^{(K)}$  decreases with increasing  $K$
- ▶ The dependence has elbow at the optimal number of clusters ( $K = 5$ )
- ▶ Let's try to formalize it

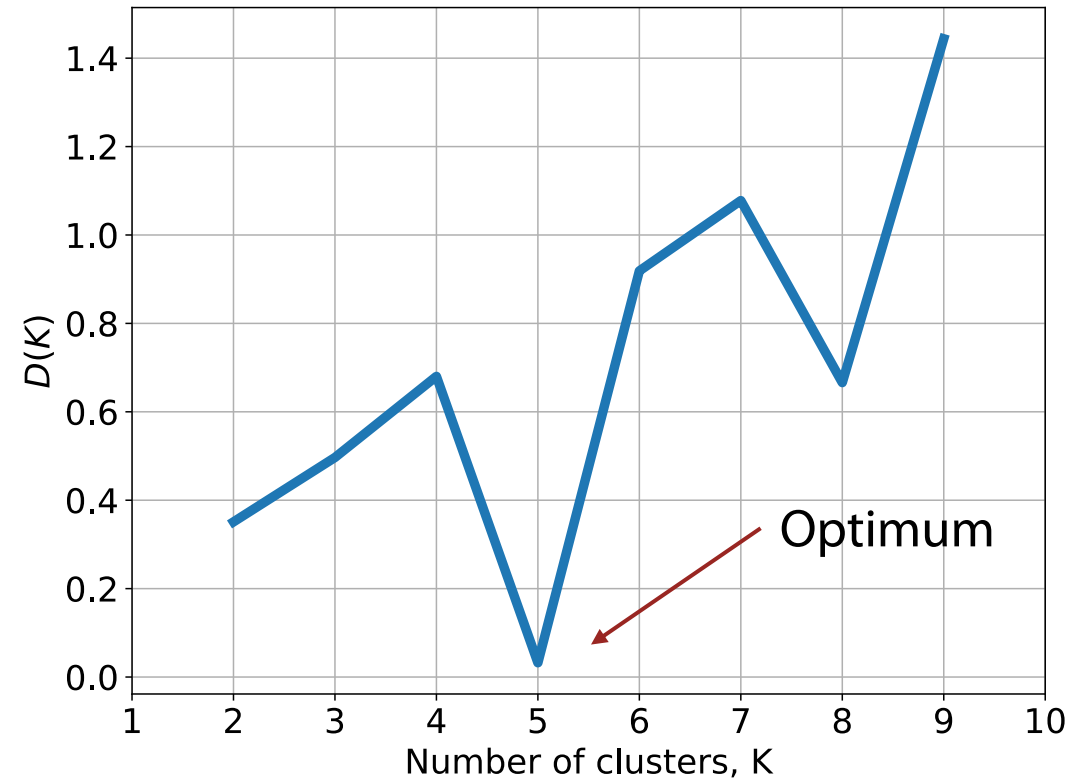


# Elbow method

- ▶ Let's define  $D(K)$ :

$$D(K) = \frac{|Q^{(K+1)} - Q^{(K)}|}{|Q^{(K)} - Q^{(K-1)}|}$$

- ▶ This function takes small value for the optimal number of clusters



# Quality Metrics



# Quality metrics

There are two kinds of quality metrics for clustering:

- ▶ Supervised

- Based on ground truth of object labels
- Invariant to cluster naming

- ▶ Unsupervised

- Based on intuition about “good” clusters:
  - Objects from the same cluster are similar / close to each other
  - Objects from different clusters are dissimilar / distant from each other

# Rand Index

Rand Index (RI) is supervised quality metric defined as:

$$RI = \frac{TP + TN}{TP + TN + FP + FN}$$

TP – number of pairs in the same cluster in predictions and the ground truth,

TN – number of pairs from different clusters in predictions and the ground truth,

FP – number of pairs in the same cluster in predictions, but from different clusters in the ground truth,

FN – number of pairs in the same cluster in the ground truth, but from the different clusters in predictions.



# Adjusted Rand Index

Adjusted Rand Index (ARI) is modification of RI:

$$ARI = \frac{RI - RI_{Expected}}{RI_{Max} - RI_{Expected}}$$

ARI has a value close to 0.0 for random labeling independently of the number of clusters and samples and exactly 1.0 when the clustering is ideal

# Metrics for classification

- ▶ Precision =  $\frac{TP}{TP + FN}$
- ▶ Recall =  $\frac{TP}{TP + FP}$
- ▶ F1 – score =  $\frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$
- ▶ Fowlkes-Mallows Index (FMI) =  $\frac{TP}{\sqrt{(TP+FP)(TP+FN)}}$
- ▶ others

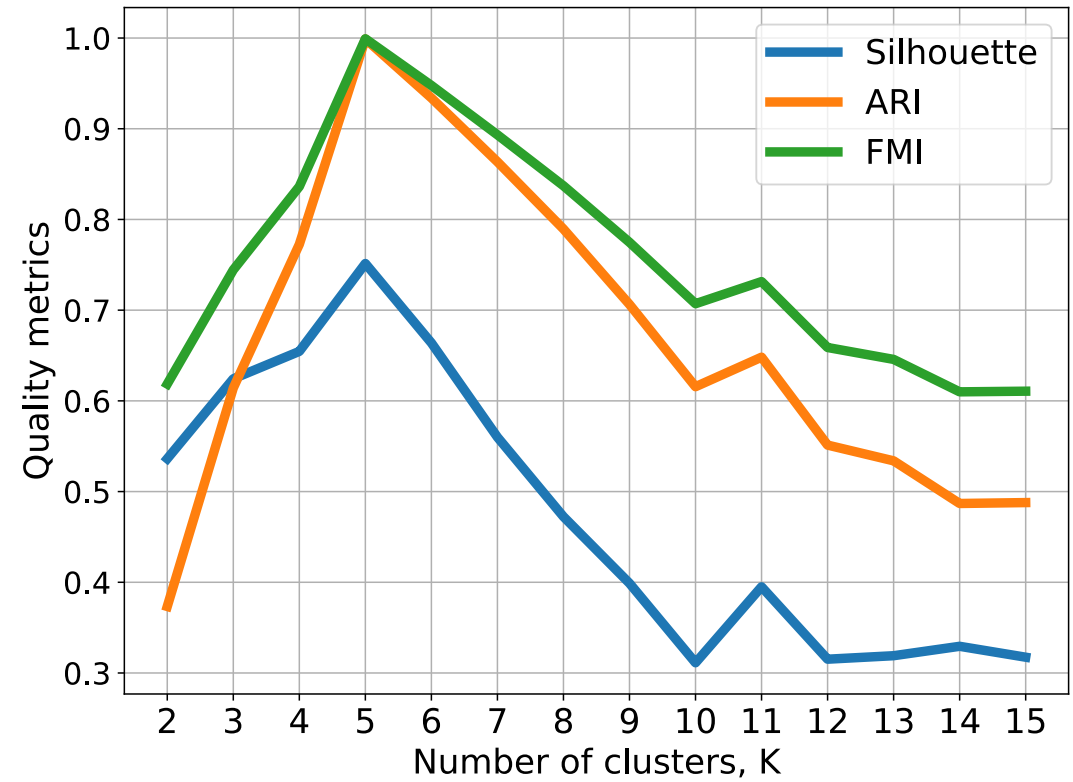
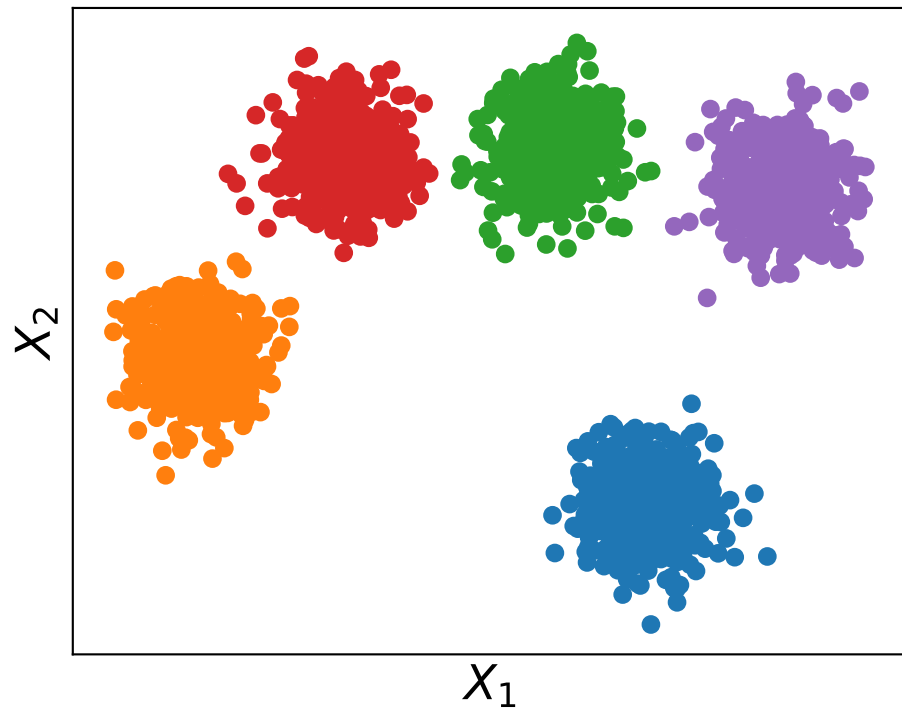
# Silhouette

Silhouette is unsupervised quality metric defined as:

$$\text{Silhouette} = \frac{1}{N} \sum_{i=1}^N \frac{d_i - s_i}{\max\{d_i, s_i\}}$$

$s_i$  - mean distance between the  $i$ -th object and all objects in the same cluster,  
 $d_i$  - mean distance between the  $i$ -th object and all objects in the nearest cluster.

# Example

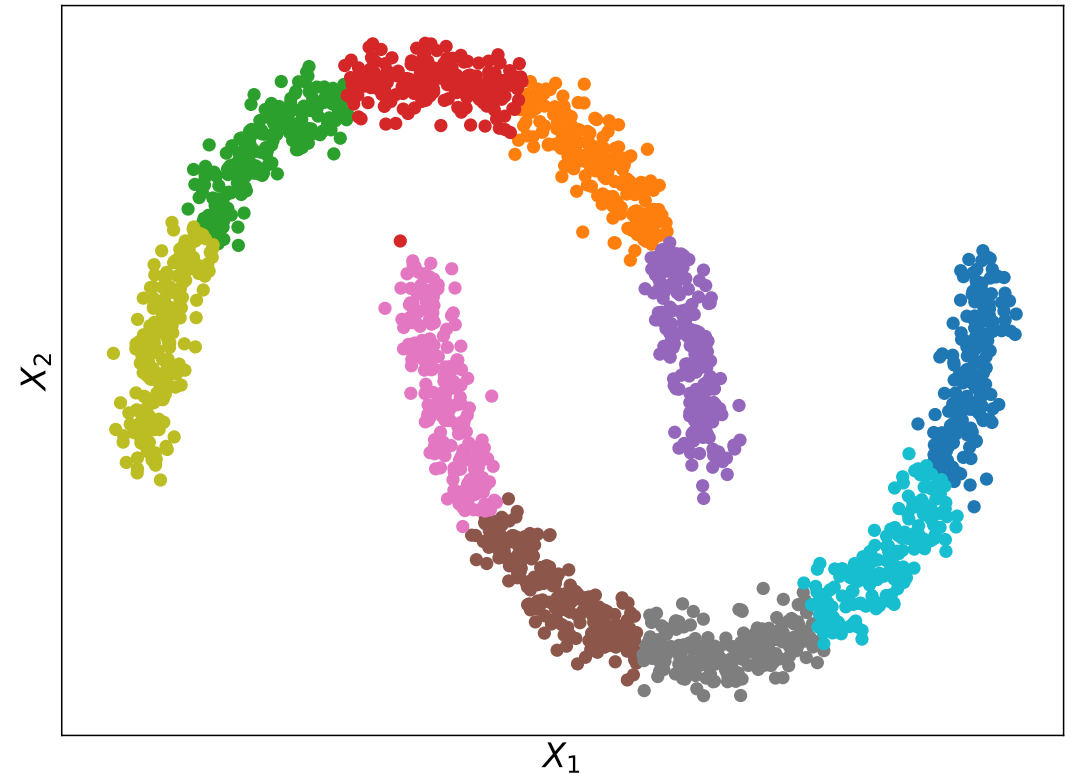


# Hierarchical Clustering

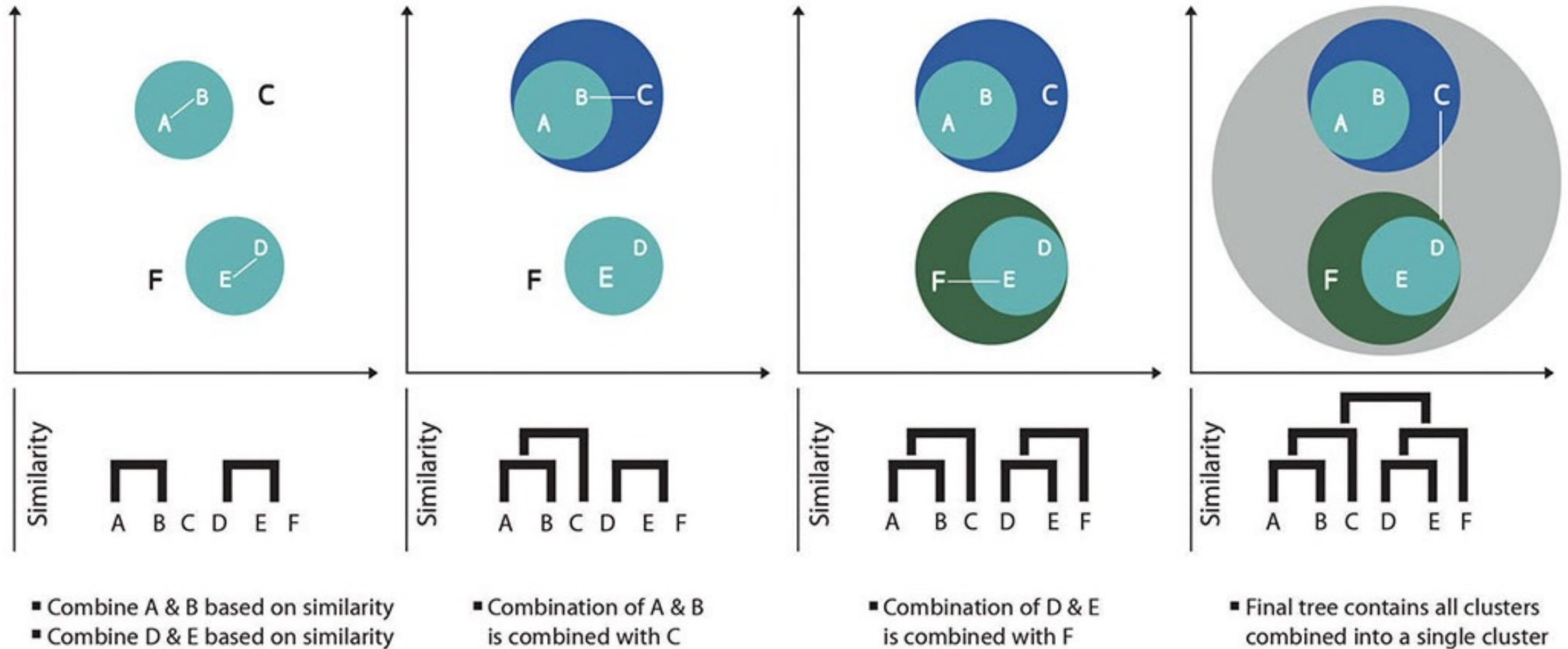


# Intuition

- ▶ Let's ask K-Means to find many clusters
- ▶ Each found cluster will be inside a real cluster
- ▶ Now, let's unite neighbor found clusters into one
- ▶ In result, we will get clusters with more complex shapes



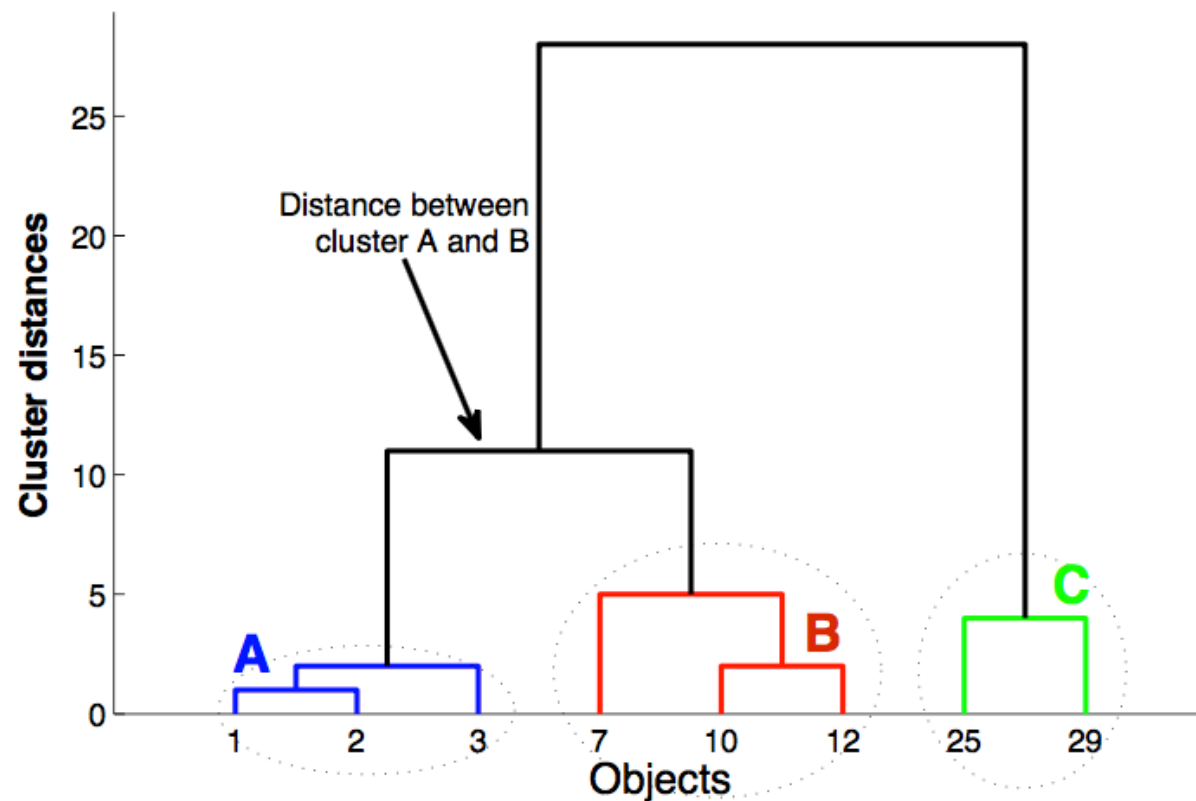
# Agglomerative clustering



Link: <https://www.brandidea.com/hierarchicalclustering.html>

# Dendrogram

- ▶ Agglomerative clustering algorithms build a dendrogram
- ▶ Dendrogram shows hierarchy of clusters in a data sample
- ▶ It contains information about objects inside each cluster and distances between these clusters





# Algorithm

initialize distance matrix  $M \in \mathbb{R}^{N \times N}$  between  
singleton clusters  $\{x_1\}, \dots, \{x_N\}$

**REPEAT:**

- 1) pick closest pair of clusters  $i$  and  $j$
- 2) merge clusters  $i$  and  $j$
- 3) delete rows/columns  $i, j$  from  $M$  and add  
new row/column for merged cluster
- 4) recalculate distances between clusters

**UNTIL** 1 cluster is left

**RETURN** hierarchical clustering of objects

# Distance between clusters #1

- ▶ Nearest neighbor (single link):

$$\rho(A, B) = \min_{a \in A, b \in B} \rho(a, b)$$

- ▶ Furthest neighbor (complete link):

$$\rho(A, B) = \max_{a \in A, b \in B} \rho(a, b)$$

where  $A = \{x_{i_1}, x_{i_2}, \dots\}$  and  $B = \{x_{j_1}, x_{j_2}, \dots\}$  are two clusters

# Distance between clusters #2

- ▶ Average (group average link):

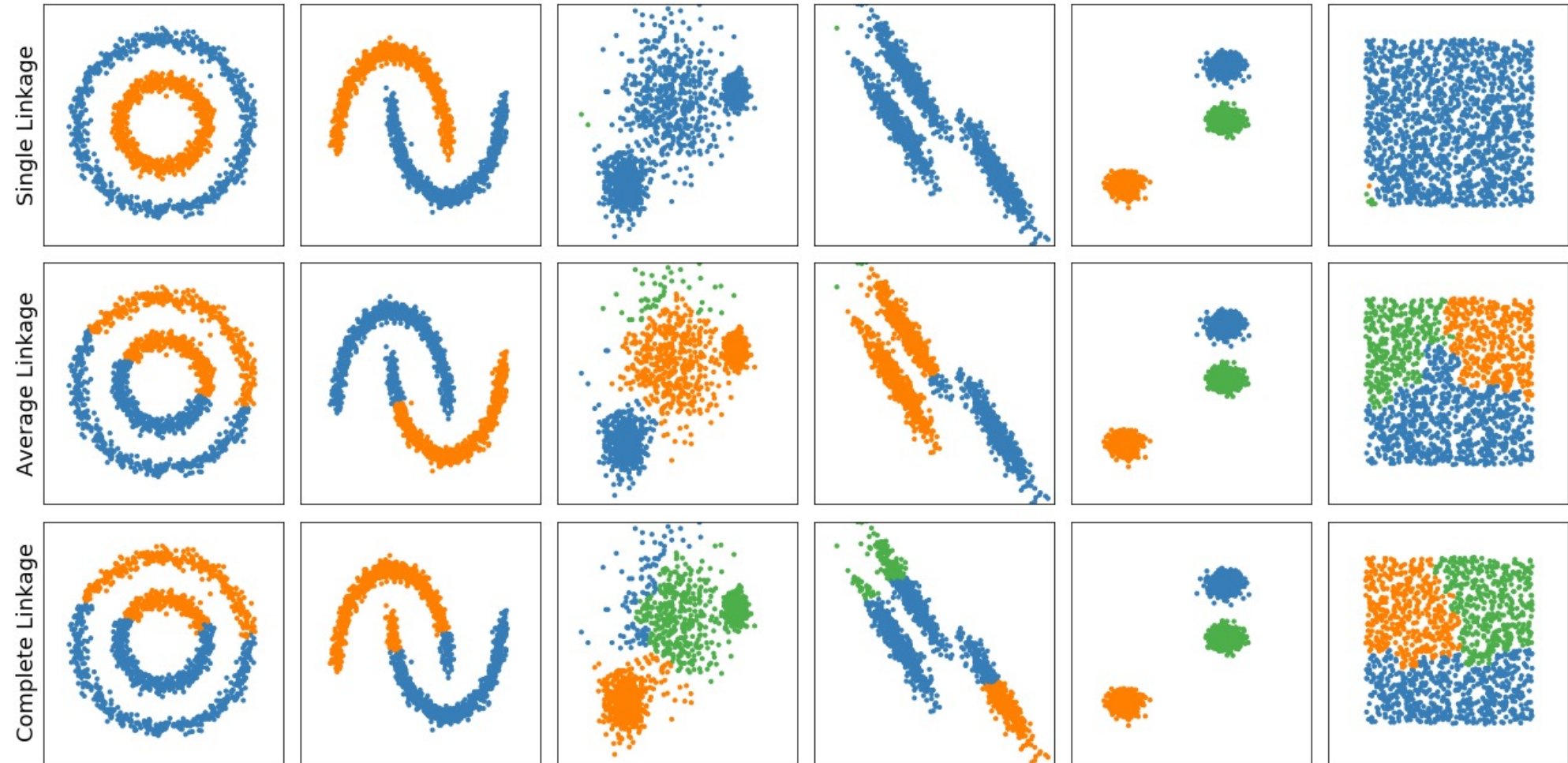
$$\rho(A, B) = \frac{1}{N_A N_B} \sum_{a \in A, b \in B} \rho(a, b)$$

- ▶ Closest centroid (centroid link):

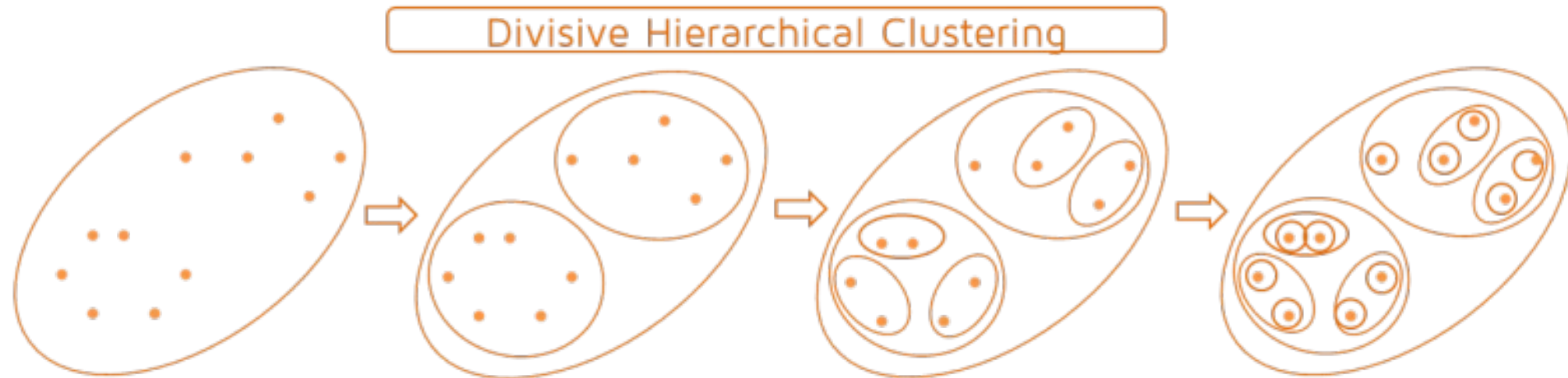
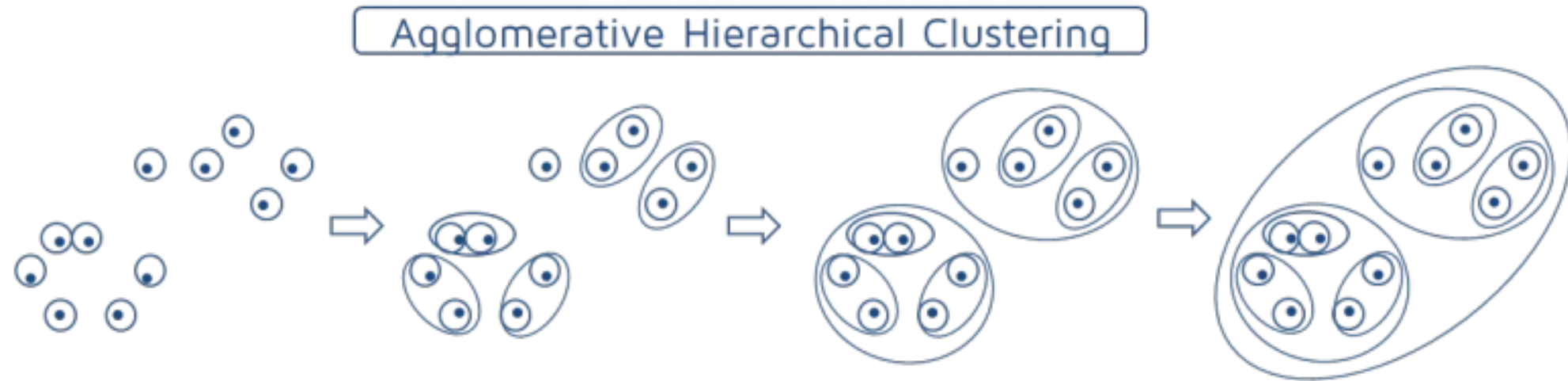
$$\rho(A, B) = \rho(\mu_A, \mu_B)$$

where  $\mu_A$  and  $\mu_B$  are cluster centers

# Demonstration



# Alternative

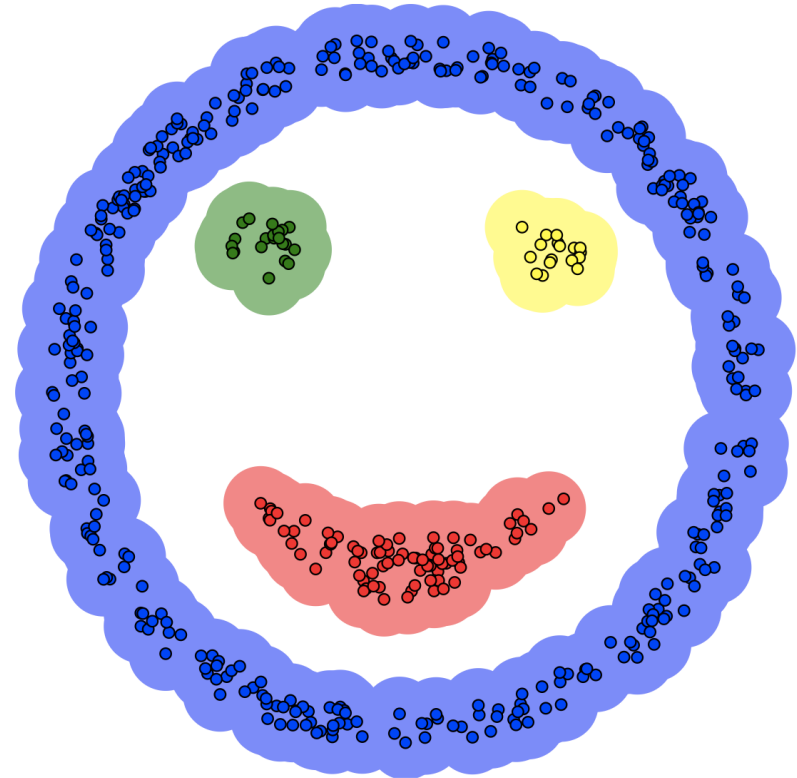


# DBSCAN



# Intuition

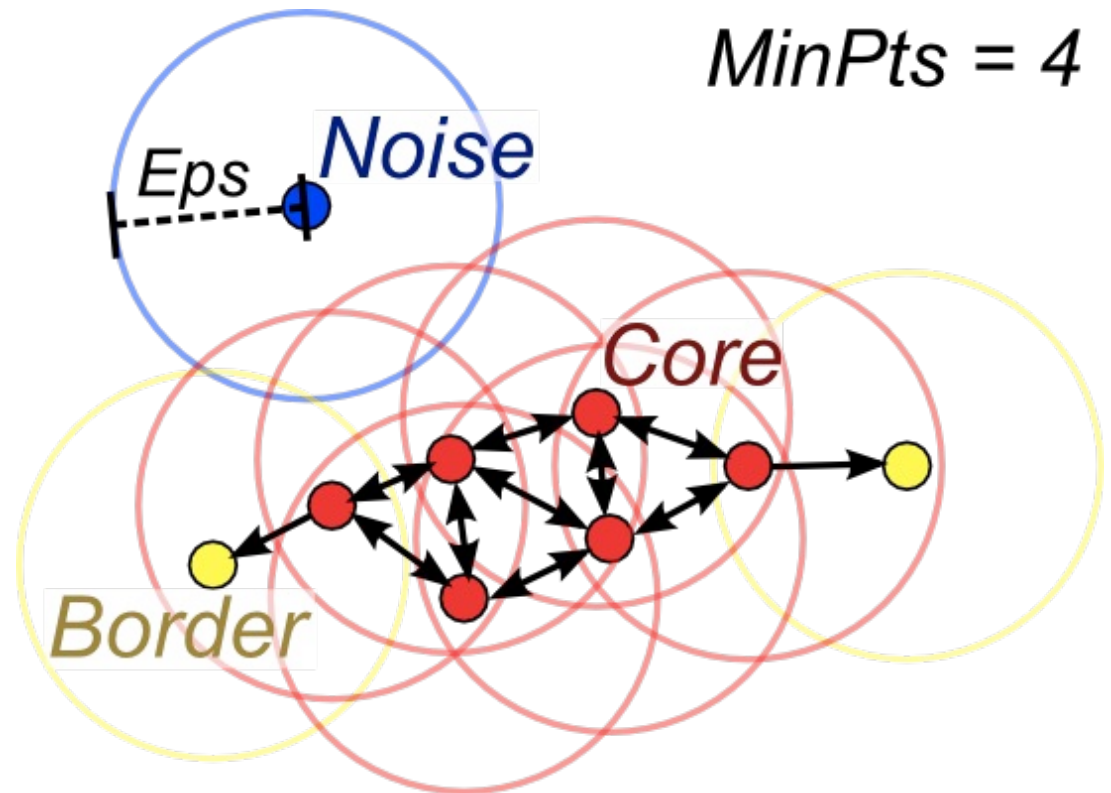
- ▶ It supposed that clusters form dense groups of objects
- ▶ Areas between the clusters are sparse, with very low densities
- ▶ Let's start from a random object and grow up a cluster by adding neighbor objects within some radius



# DBSCAN idea #1

DBSCAN has two parameters:

- ▶  $\epsilon$  – radius of neighborhood of each object
- ▶ **MinPts** – minimal number of objects inside the neighborhood

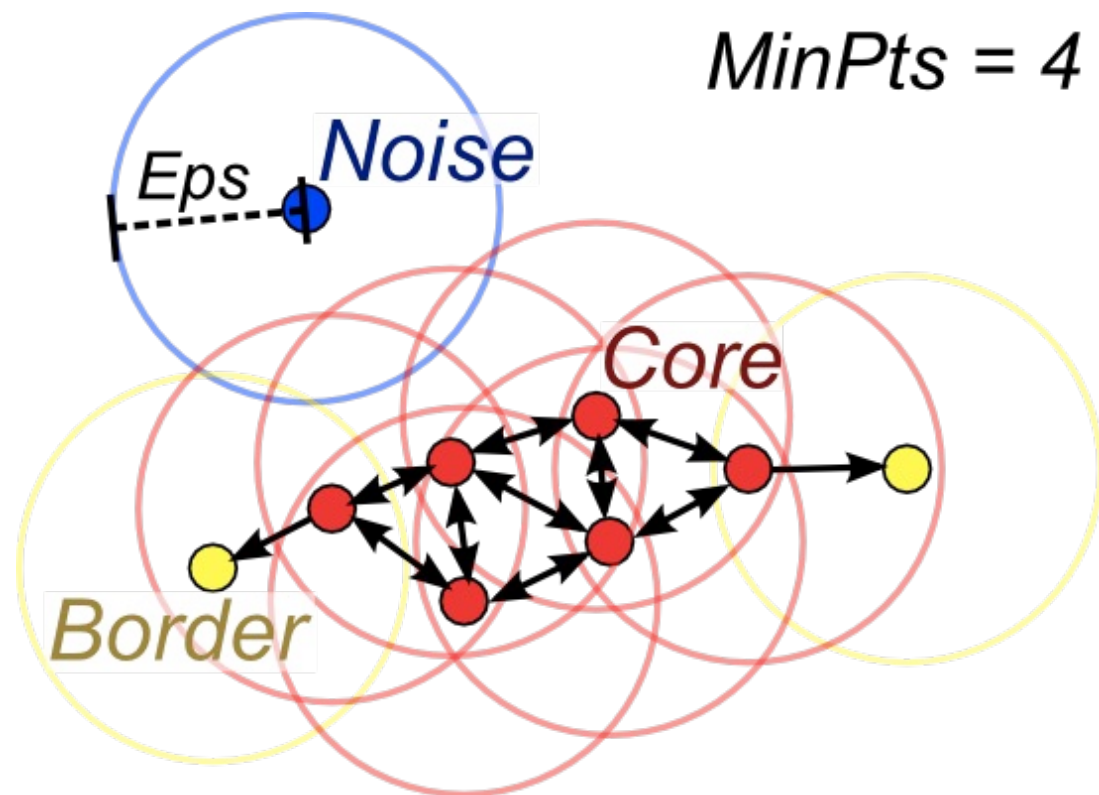




# DBSCAN idea #2

Three types of objects:

- ▶ **Core:** has  $\geq \text{MinPts}$  objects within its  $\epsilon$  neighborhood
- ▶ **Border:** not core object, has at least 1 core object within its  $\epsilon$  neighborhood
- ▶ **Noise:** neither a core nor a border point



# Algorithm (short)

---

**Algorithm 1:** DBSCAN algorithm.

---

- 1: Label all objects as core, border, or noise objects.
  - 2: Eliminate noise objects.
  - 3: Put an edge between all core objects that are within  $\epsilon$  of each other.
  - 4: Make each group of connected core objects into a separate cluster.
  - 5: Assign each border object to one of the clusters of its associated core objects.
-

# Algorithm (detailed)

```
1.function dbscan(X, eps, min_pts):
2.  initialize NV = X # not visited objects
3.  for x in NV:
4.      remove(NV, x) # mark as visited
5.      nbr = neighbours(x, eps) # set of neighbours
6.      if nbr.size < min_pts:
7.          mark_as_noise(x)
8.      else:
9.          C = new_cluster()
10.         expand_cluster(x, nbr, C, eps, min_pts, NV)
11.         yield C
```

Link: [https://shestakoff.github.io/hse\\_se\\_ml/2020/l14-cluster/lecture-clust.slides#/4/5](https://shestakoff.github.io/hse_se_ml/2020/l14-cluster/lecture-clust.slides#/4/5)

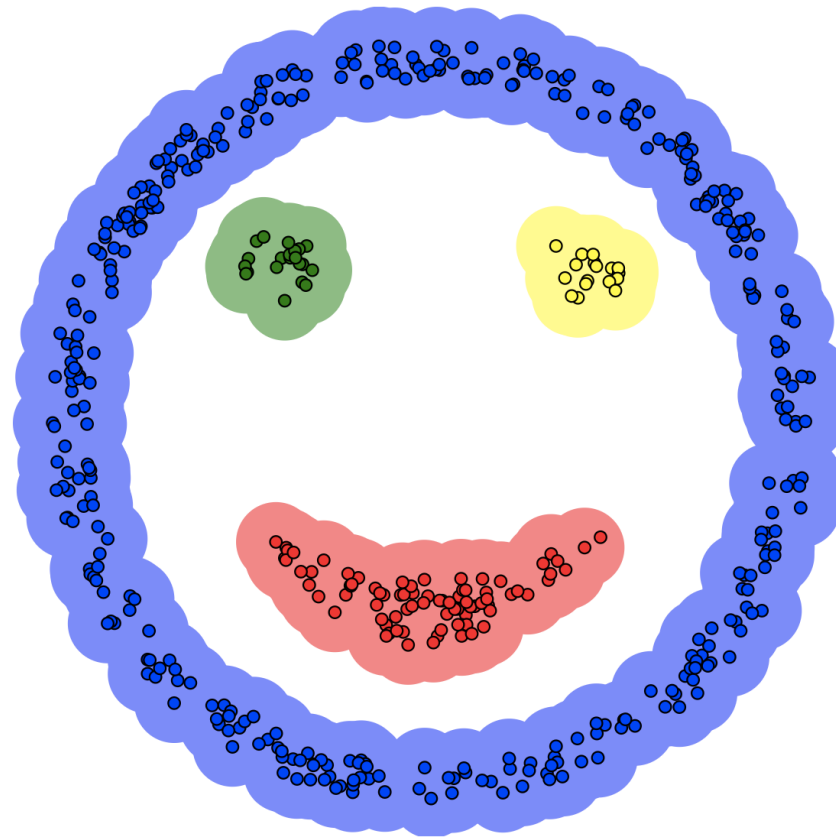
# Algorithm (detailed)

```
1. function expand_cluster(x, nbr, C, eps, min_pts, NV):
2.   add(x, C)
3.   for x1 in nbr:
4.     if x1 in NV: # object not visited
5.       remove(NV, x1) # mark as visited
6.       nbr1 = neighbours(x1, eps)
7.       if nbr1.size >= min_pts:
8.         # join sets of neighbours
9.         merge(nbr, nbr1)
10.    if x1 not in any cluster:
11.      add(x1, C)
```

Link: [https://shestakoff.github.io/hse\\_se\\_ml/2020/l14-cluster/lecture-clust.slides#/4/5](https://shestakoff.github.io/hse_se_ml/2020/l14-cluster/lecture-clust.slides#/4/5)

# Demonstration

Demo: <https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/>



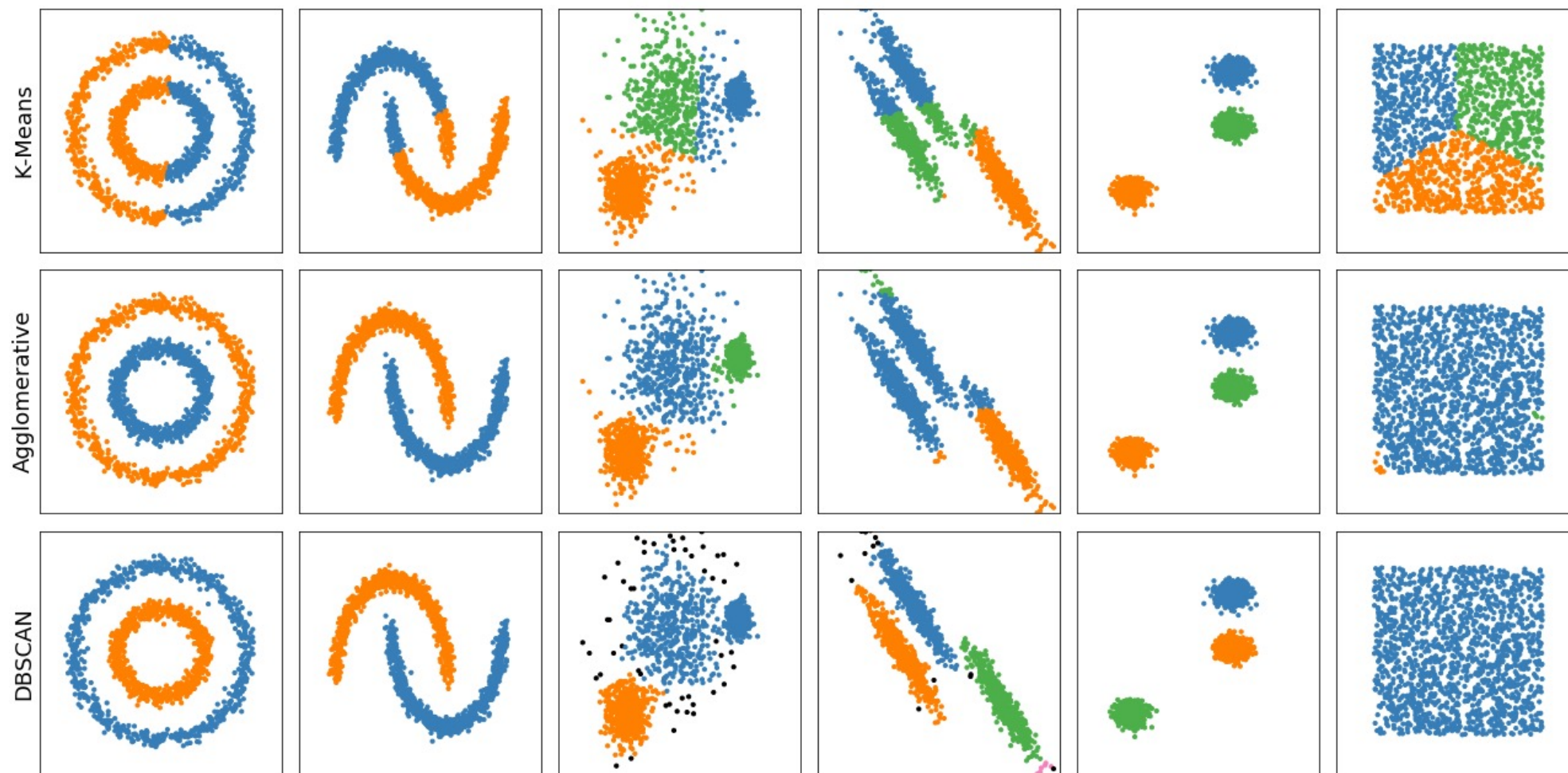
# Properties

- ▶ Number of clusters is estimated automatically
- ▶ Robust to outliers. They are recognized as a noise
- ▶ Can find clusters with complex shapes
- ▶ Sensitive to objects density variations

# Заключение



# Резюме





# Вопросы

- ▶ Что такое задача кластеризации? Как измеряется качество в задаче кластеризации? Запишите формулы для внутрикластерного и межкластерного расстояний.
- ▶ Опишите, как работает метод K-Means. Какой критерий он оптимизирует?
- ▶ Опишите, как работает метод DBSCAN.
- ▶ Как работает подход к кластеризации, основанный на графах? Как работает иерархическая кластеризация?