IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Machine Learning Approaches : Clustering

B. Solaiman
Départ. Image &Traitement de l'Information
Brest, France

1. ***Clustering concept***
2. ***Clustering distances***
3. ***Clustering approaches***
4. ***Partitioning algorithms:***

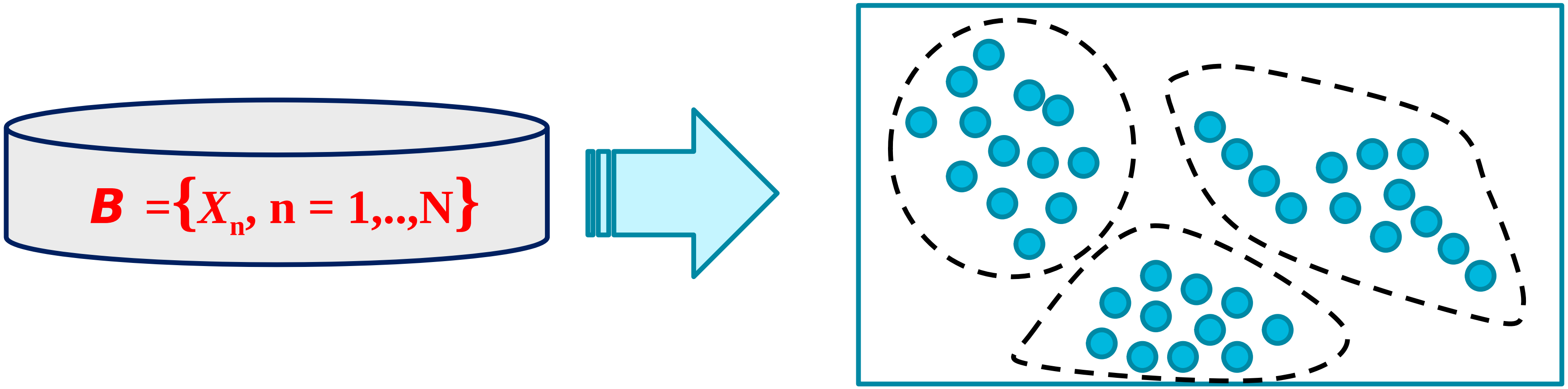
- ***K-means algorithm***
- ***Semi-Supervised K-means***
- ***K-medoids***
- ***Isodata algorithm***

5. ***K-Means algorithm applications***
6. ***Quality of clustering***

1. Clustering Concept

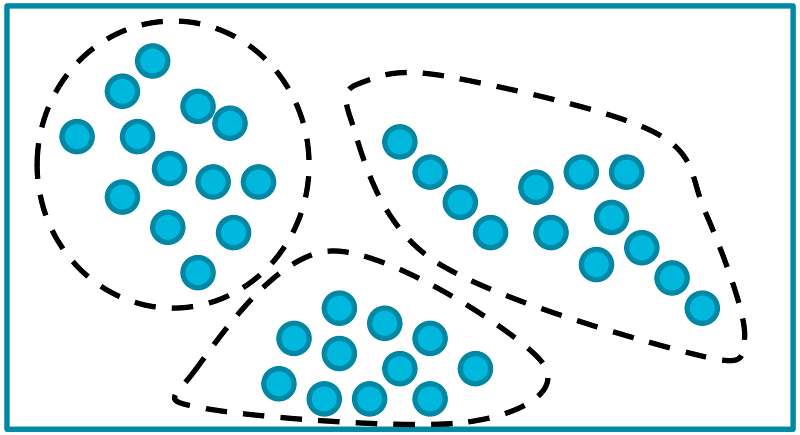
CLUSTERING:

The process of partitioning a set of instances / objects into several subsets (called <u>clusters</u>), so that the instances in each subset share some common trait (according to some predefined similarity measure)



CLUSTER

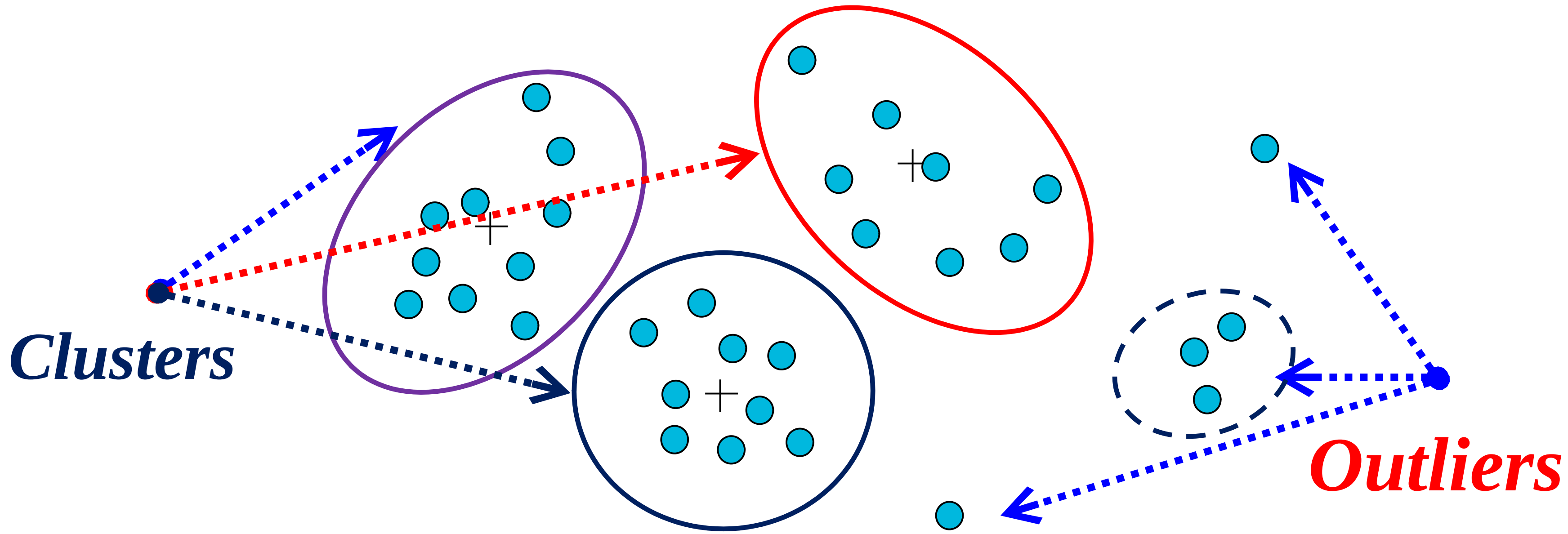
A collection/group of data instances "similar" to one another within the same group, and, dissimilar to the instances in other groups



CLUSTERING ANALYSIS: ***refers to the <u>use of "similarities"</u> between data instances and <u>unsupervised learning</u> techniques in order to group similar instances allowing, thus, to <u>find the intrinsic hidden structure within unlabeled data</u>***

CLUSTERING IMPORTANT ASPECTS

Outliers ***are instances that do not belong to any cluster (or instances forming clusters of very small cardinality)***

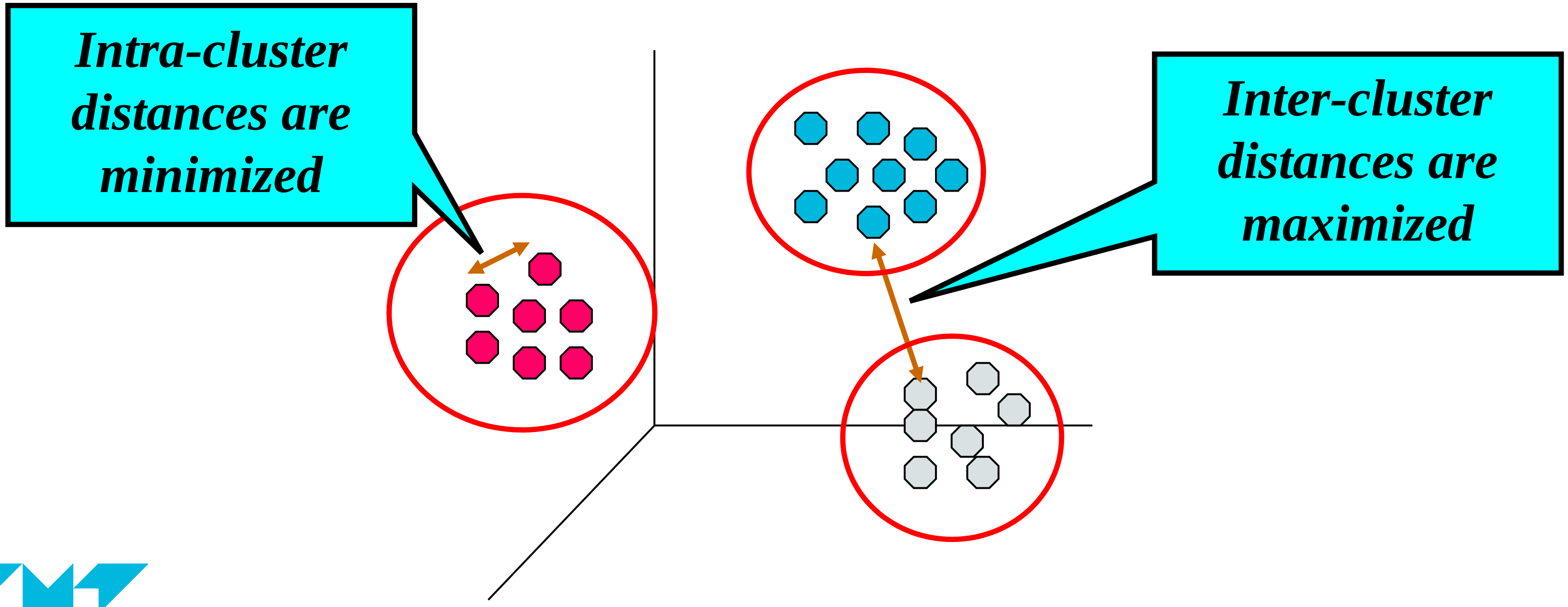


In some applications (*Rare Events detection*): we are interested in discovering outliers, not clusters (outlier analysis)

Clustering concept

CLUSTERING BASIC QUESTIONS

- *Clustering **quality** (How to evaluate the partition's quality, number of clusters….) ?*
- *What does **similar** mean ?*
- *Distance (similarity, or dissimilarity) function definition!*
- *Clustering **approach** leading to a good partition ?*

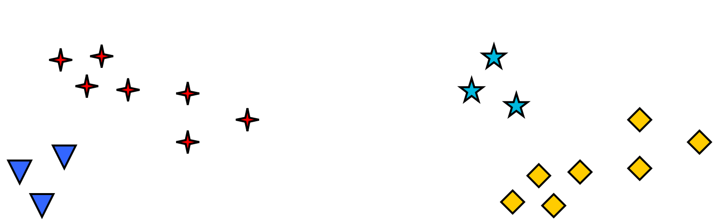


Clustering concept

Illustrative Example : how many clusters?

How many clusters?

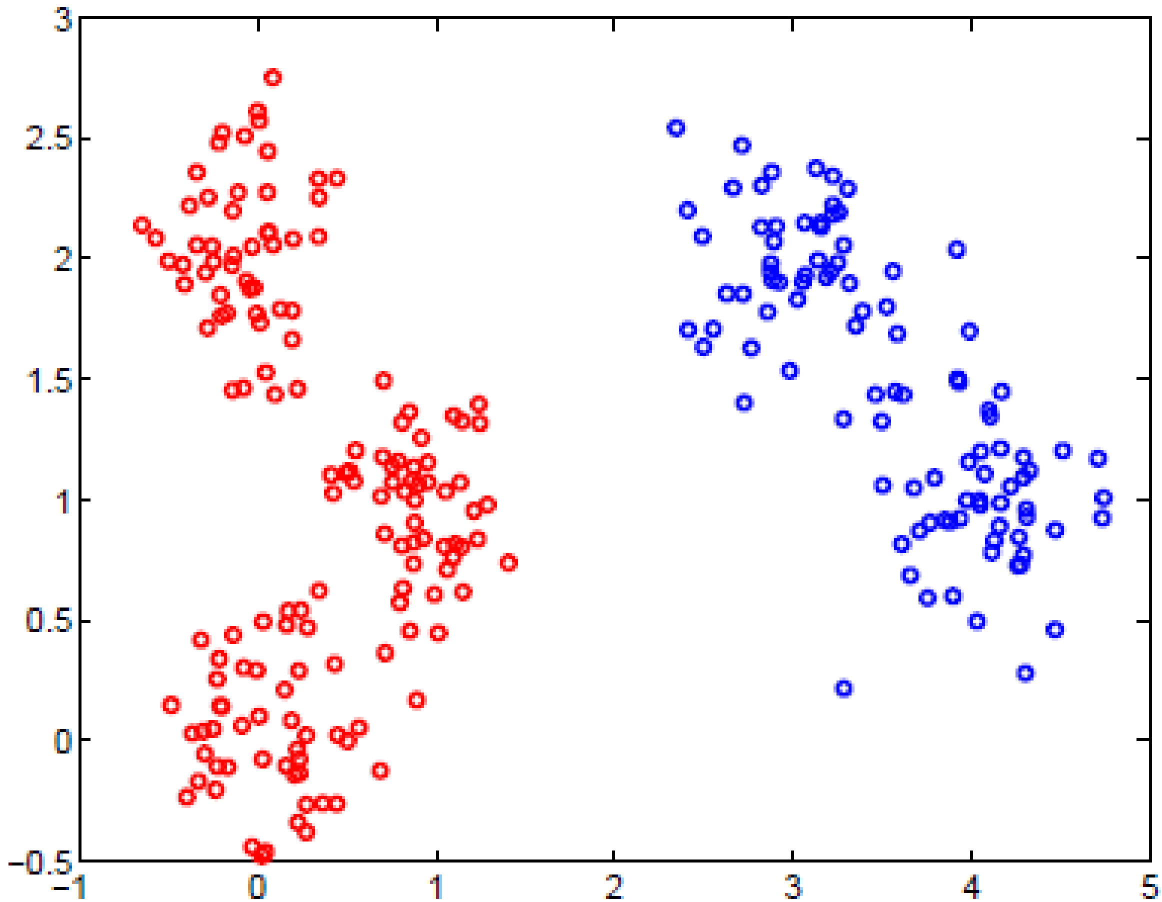
Two Clusters



Four Clusters

Six Clusters

Illustrative Example : how many clusters?



2. Clustering distances

Clustering distances

The clustering approaches depend on the choice of the *Similarity* (distance function) between clusters :

Single linkage **: distance between the closest neighbors**

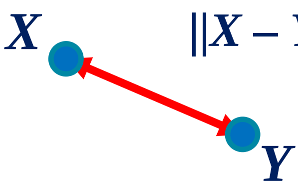
Complete linkage **: distance between the furthest neighbors**

Central linkage **: distance of centers (centroids)**

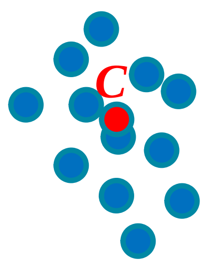
Average linkage **: average distance of all patterns in each cluster**

Clustering distances

Notations



$\|X - Y\|$: **Distance between two instances X and Y**



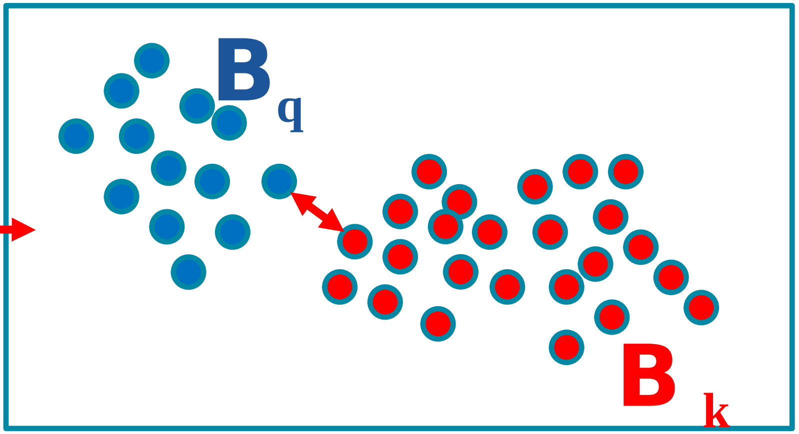
B : Cluster

C : **Centroid**

Clustering distances

Single Linkage distance

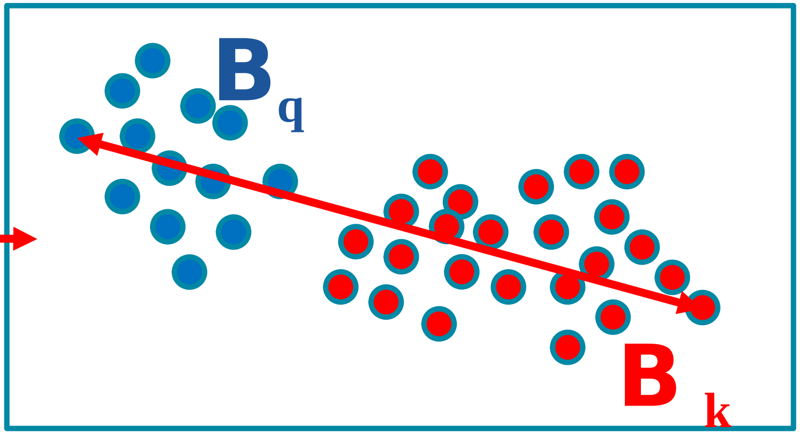
$$\text{Dist}_{min}(\mathbf{B}_k, \mathbf{B}_q) = \min_{X\in\mathbf{B}_k,\ Y\in\mathbf{B}_q} \|X - Y\|^2$$



Complete Linkage distance

$$\text{Dist}_{max}(\mathbf{B}_k, \mathbf{B}_q) = \max_{X\in\mathbf{B}_k,\ Y\in\mathbf{B}_q} \|X - Y\|^2$$

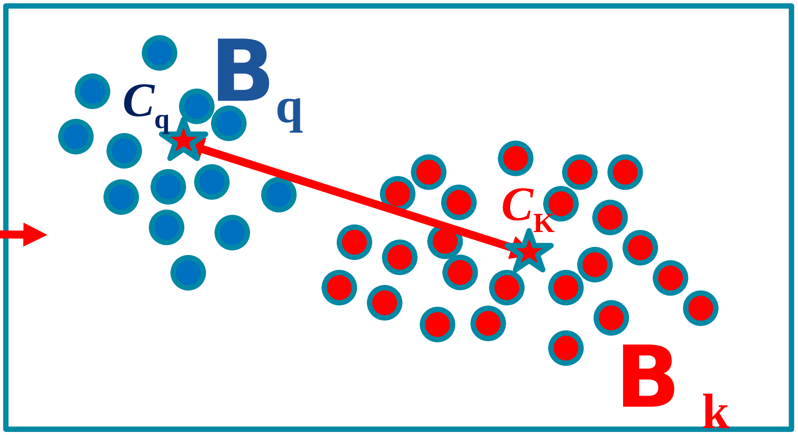
(Allows avoiding elongated clusters)



Clustering distances

Centroid Linkage distance

$$\text{Dist}_{\text{means}}(\mathbf{B}_k, \mathbf{B}_q) = \| C_K - C_q \|^2$$



Average distance

$$\text{Dist}_{\text{ave}}(\mathbf{B}_k, \mathbf{B}_q) = \frac{1}{|\mathbf{B}_k| \cdot |\mathbf{B}_q|} \sum_{X \in \mathbf{B}_k,\ Y \in \mathbf{B}_q} \|X - Y\|^2$$

3. Clustering approaches

1. Hierarchical clustering algorithms

Find successive clusters using previously established clusters

A. Agglomerative ("bottom-up") algorithms

Begin with each instance as a separate cluster and merge them into successively larger clusters

B. Divisive ("top-down") algorithms

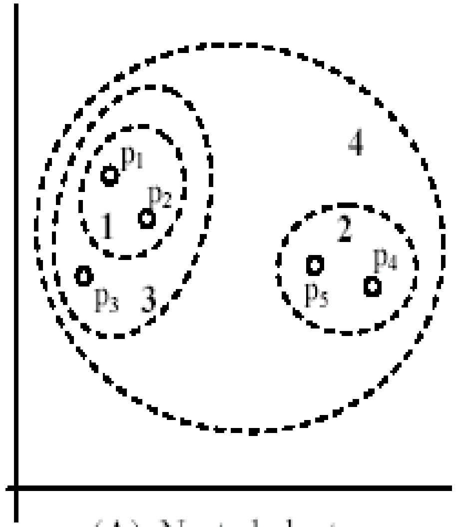
Begin with the whole set and proceed to divide it into successively smaller clusters

2. Partitional clustering algorithms

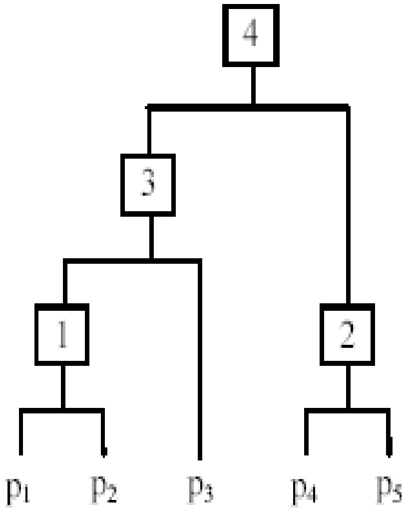
Construct a single partition of all clusters at once and then evaluate them by some criterion

. Hierarchical Clustering algorithms

Hierarchical Clustering: is a deterministic approach producing, iteratively, a nested sequence of clusters

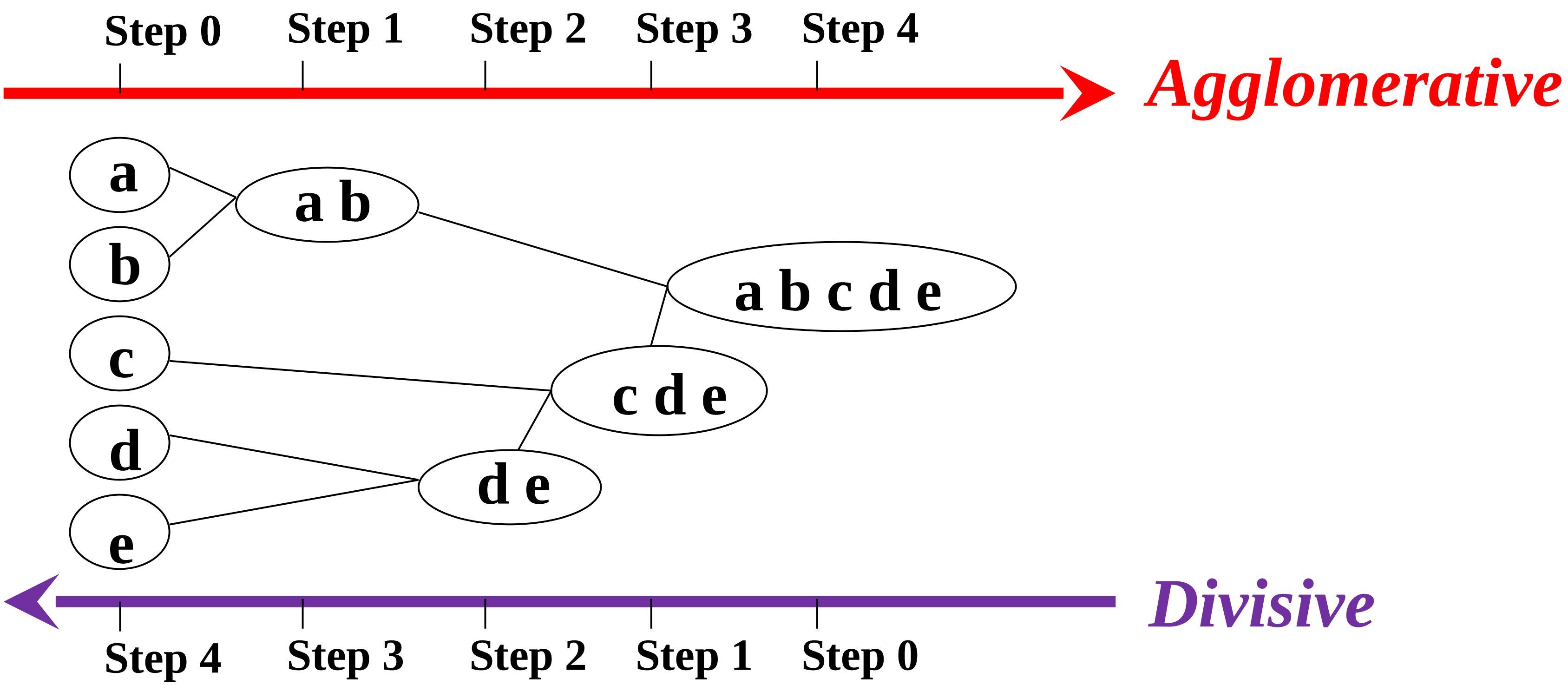


(A). Nested clusters



(B) Dendrogram

. Hierarchical Clustering algorithms



Clustering approaches

Agglomerative (*Bottom-Up*) clustering :

Start with each instance as its own cluster

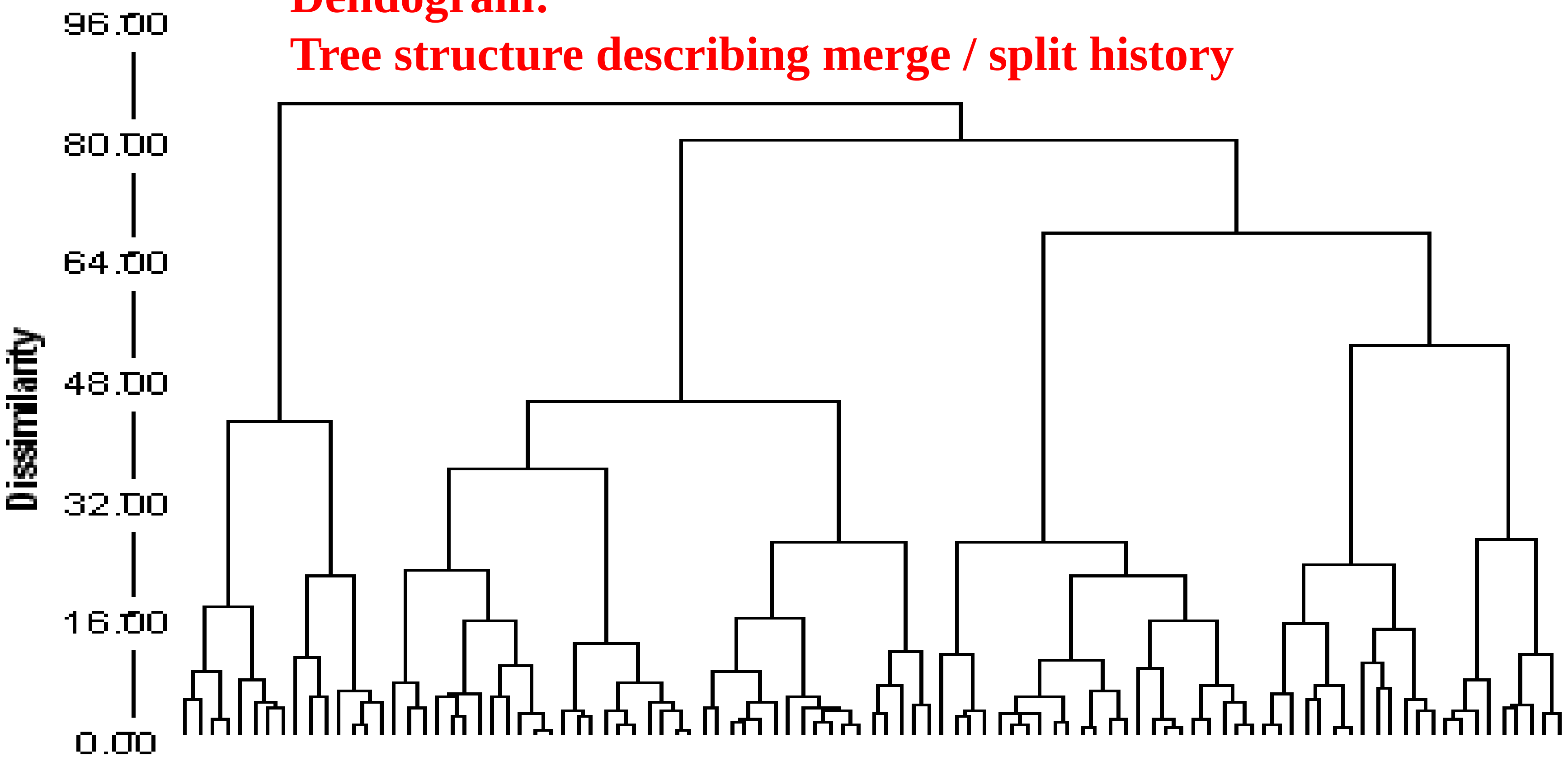
and iteratively

Find the best pair to merge the closest clusters

Repeat until all clusters are fused together

HIERARCHICAL AGGLOMERATIVE CLUSTERING

Dendogram:
Tree structure describing merge / split history

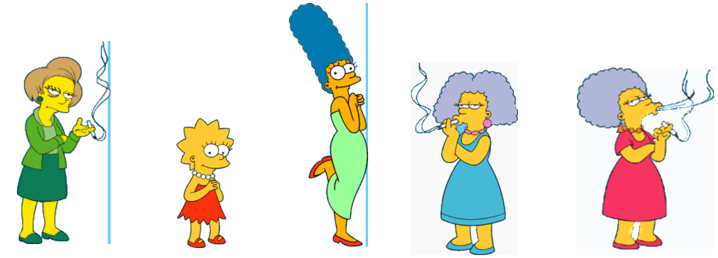


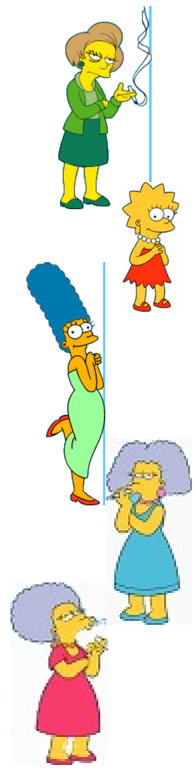
Clustering approaches

We begin with a distance matrix which contains the distances between every pair of instances in the database

D(,) = 8

D(,) = 1





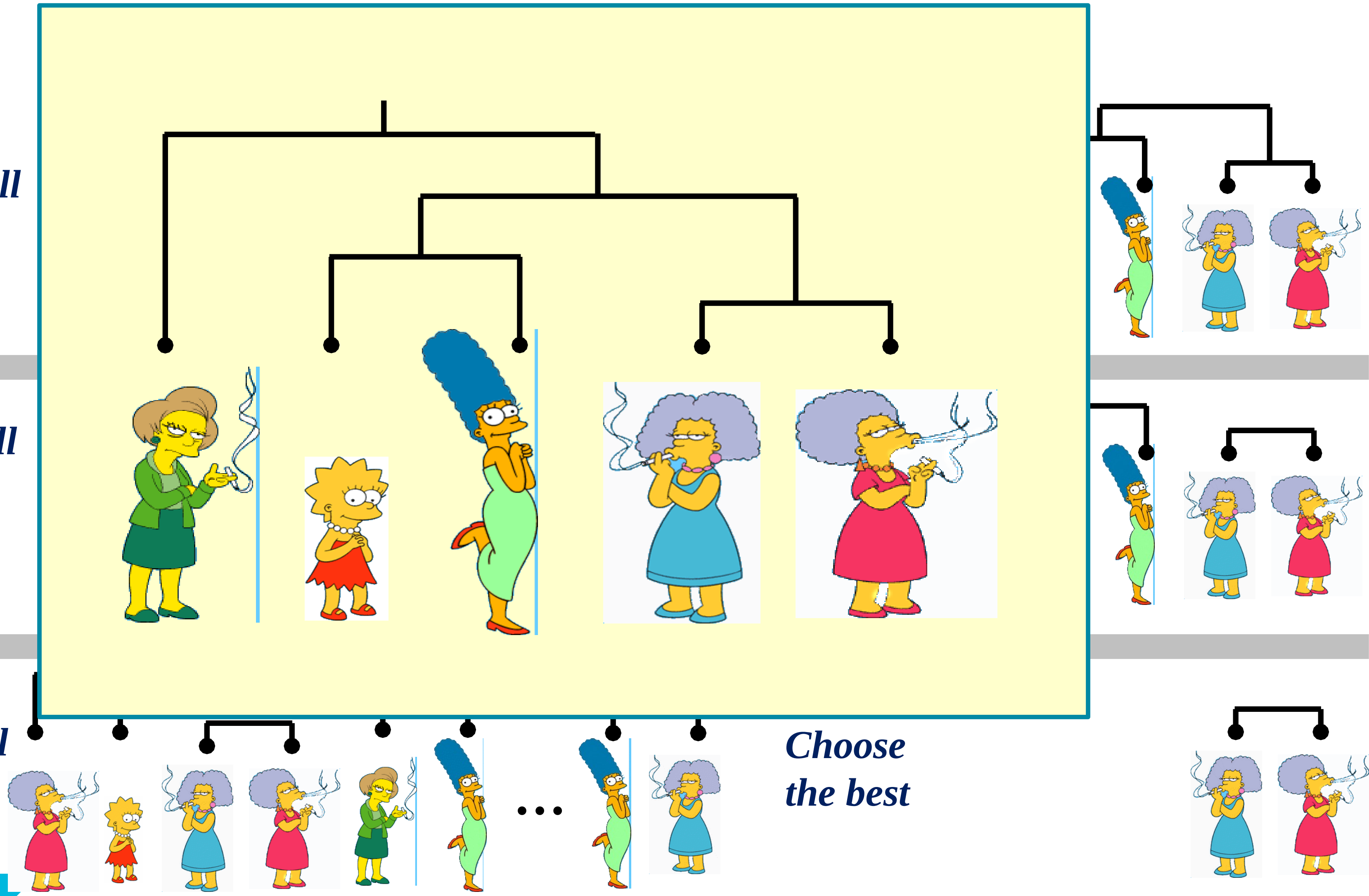
0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

Clustering approaches

Consider all possible merges…

Consider all possible merges…

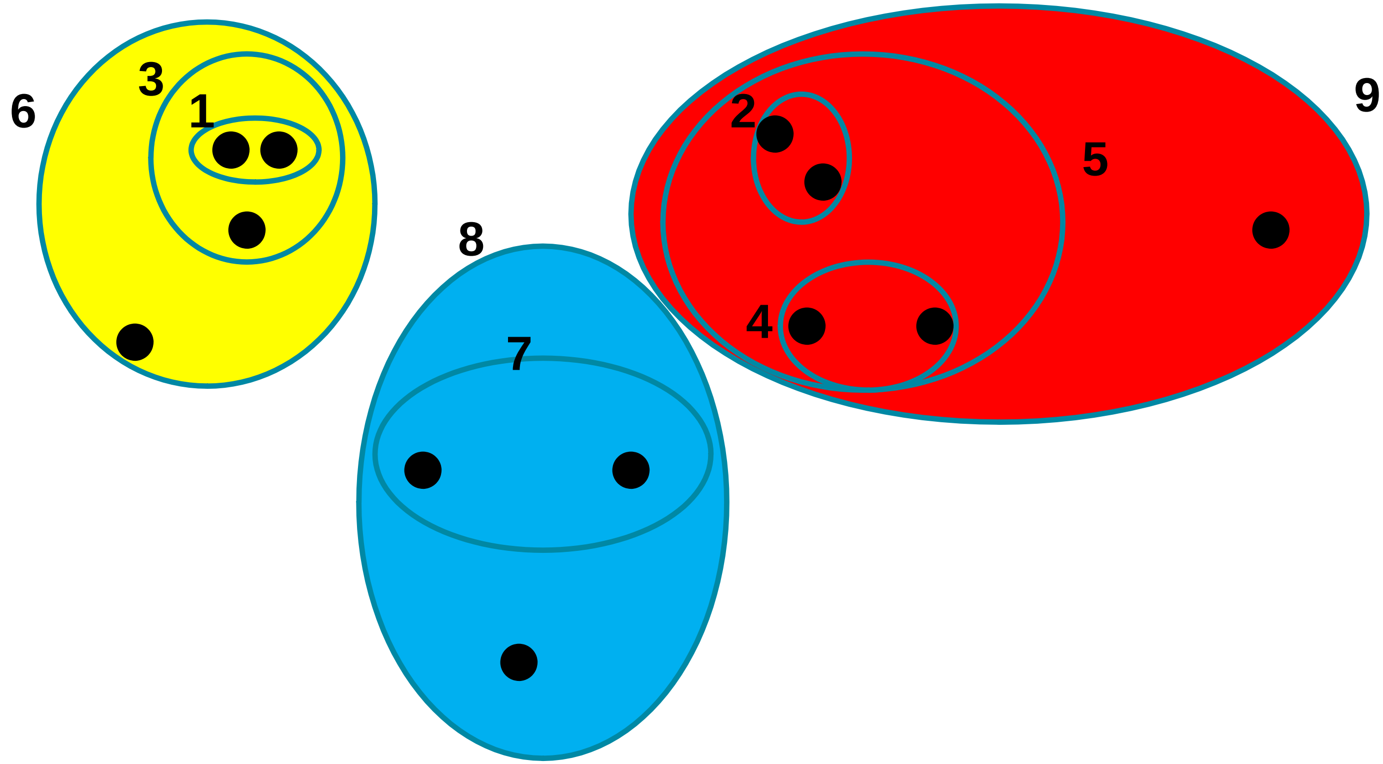
Consider all possible merges…



Agglomerative (*Bottom-Up*) clustering algorithm

1. *Calculate the distance between all instances*
2. *Cluster the instances to the initial clusters*
3. *Calculate the distance metrics between all clusters*
4. *Interatively cluster most similar clusters into a higher level cluster*
5. *Repeat steps 3 and 4 for the most high-level clusters*

Agglomerative (*Bottom-Up*) clustering algorithms



Clustering approaches

ample: Airports agglomerative clustering

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	$\boxed{138}$
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

BA FI MI NA RM TO

BA FI MI - TO NA RM

Clustering approaches

	BA	FI	MI/ TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	$\boxed{219}$
RM	412	268	564	219	0

BA **FI** **MI** **NA** **RM** **TO**

BA **FI** **MI - TO** **NA** **RM**

BA **FI** **MI - TO** **NA - RM**

Clustering approaches

	BA/NA/RM	FI	MI/TO
BA/NA/RM	0	268	564
FI	268	0	295
MI/TO	564	295	0

BA FI MI NA RM TO

BA FI MI - TO NA RM

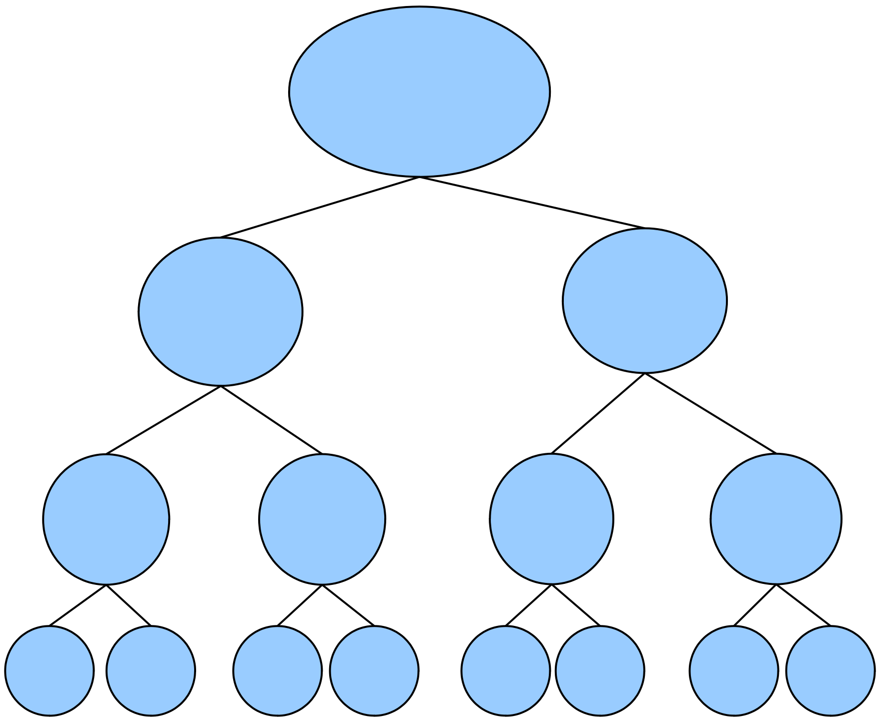
BA FI MI - TO NA - RM

FI MI - TO BA - NA - RM

MI - TO FI - BA - NA - RM

Divisive (*Top-Down*) clustering algorithms

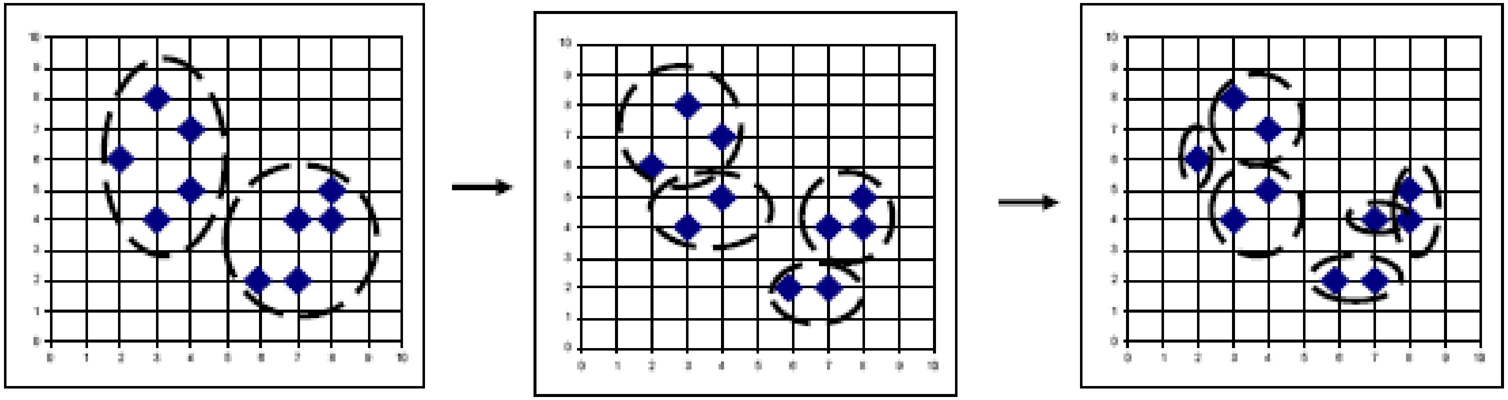
Starting with all the data in a single cluster, consider every possible way to divide the cluster into two. Choose the best division and recursively operate on both sides



Divisive (*Top-Down*) clustering algorithm

All instances are considered to be in one super-cluster

- ***Start at the top with all instances in one cluster***
- ***The cluster is split using a flat clustering algorithm***
- ***This procedure is applied recursively until each pattern is in its own singleton cluster***



4. Partitioning algorithms:

- K-means algorithm

- Semi-Supervised K-means

- K-medoids

- ISODATA

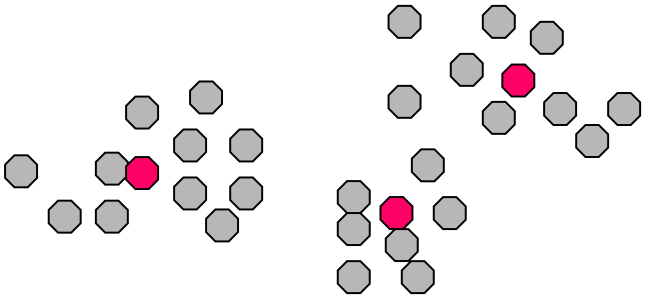
A partitioning approach

An algorithm allowing to construct, AT ONCE, a partition of a set of N instances into a set of K clusters, where:

- ***Each instance belongs to exactly one cluster***
- ***The number of clusters K is given in advance***

K-means algorithm

K-Means Problem: Given a set $\mathbf{B} = \{X_n, n=1, .., N, X_n \in R^d\}$ of N points (objects, samples, instances, …) in a d-dimensional space and an integer K.



Task: find a set of K points $\mathbf{C} = \{C_1, C_2, \ldots, C_K\}$ in R^d to form clusters $\{\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_K\}$ such that:

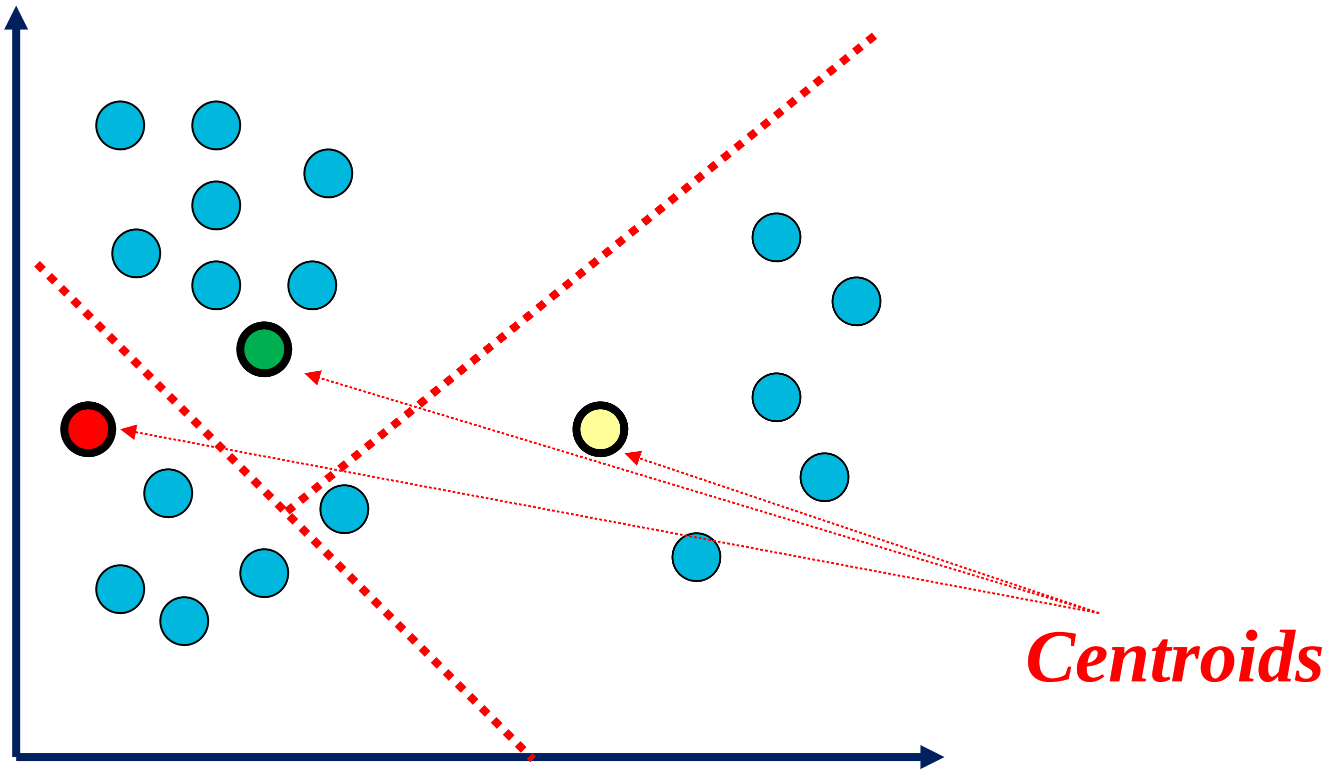
$$Cost(\mathbf{C}) = \sum_{k=1,..,K} \sum_{X \in \mathbf{B}_k} \mathrm{dist}^2(X, C_k)$$

is minimized

K-means algorithm

K-means algorithm: ***One way to solve the K-means problem:***

- *Each cluster is "iteratively" associated with a* ***centroid*** *(center point)*
- *Each point is assigned to the cluster with the closest centroid*



K-means algorithm

***K-means algorithm:* One way to solve the K-means problem:**

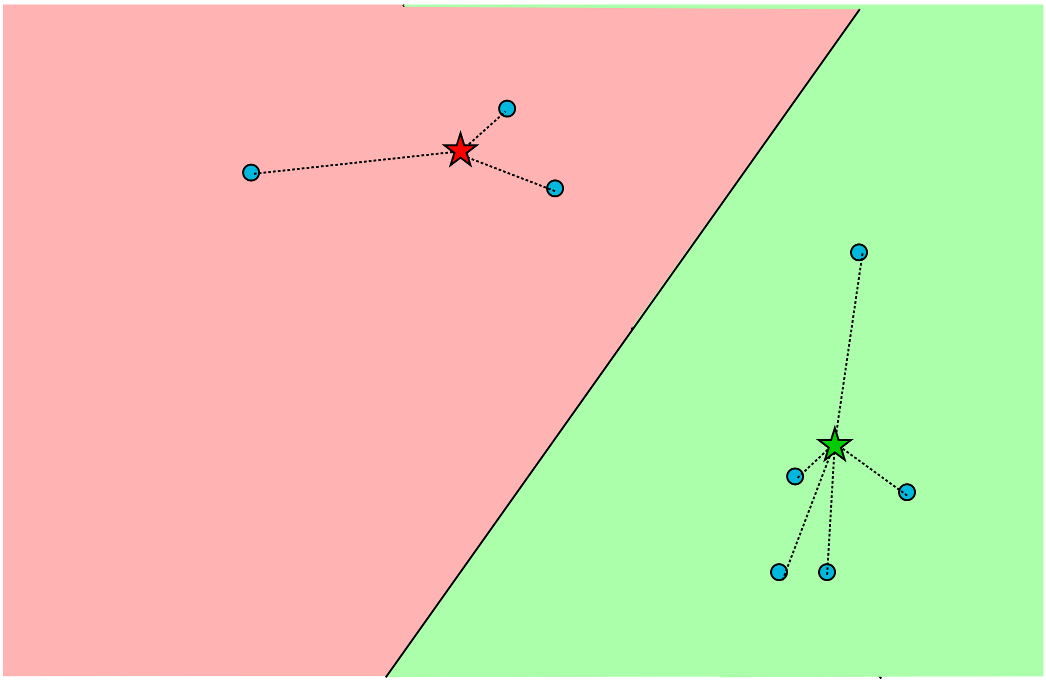
- ***Each cluster is "iteratively" associated with a centroid (center point);***
- ***Each point is assigned to the cluster with the closest centroid***

- ***Randomly pick* K *initial cluster centroids* $\{C_1, C_2, \ldots, C_K\}$**
- ***Repeat until convergence (i.e., centroids don't change)***

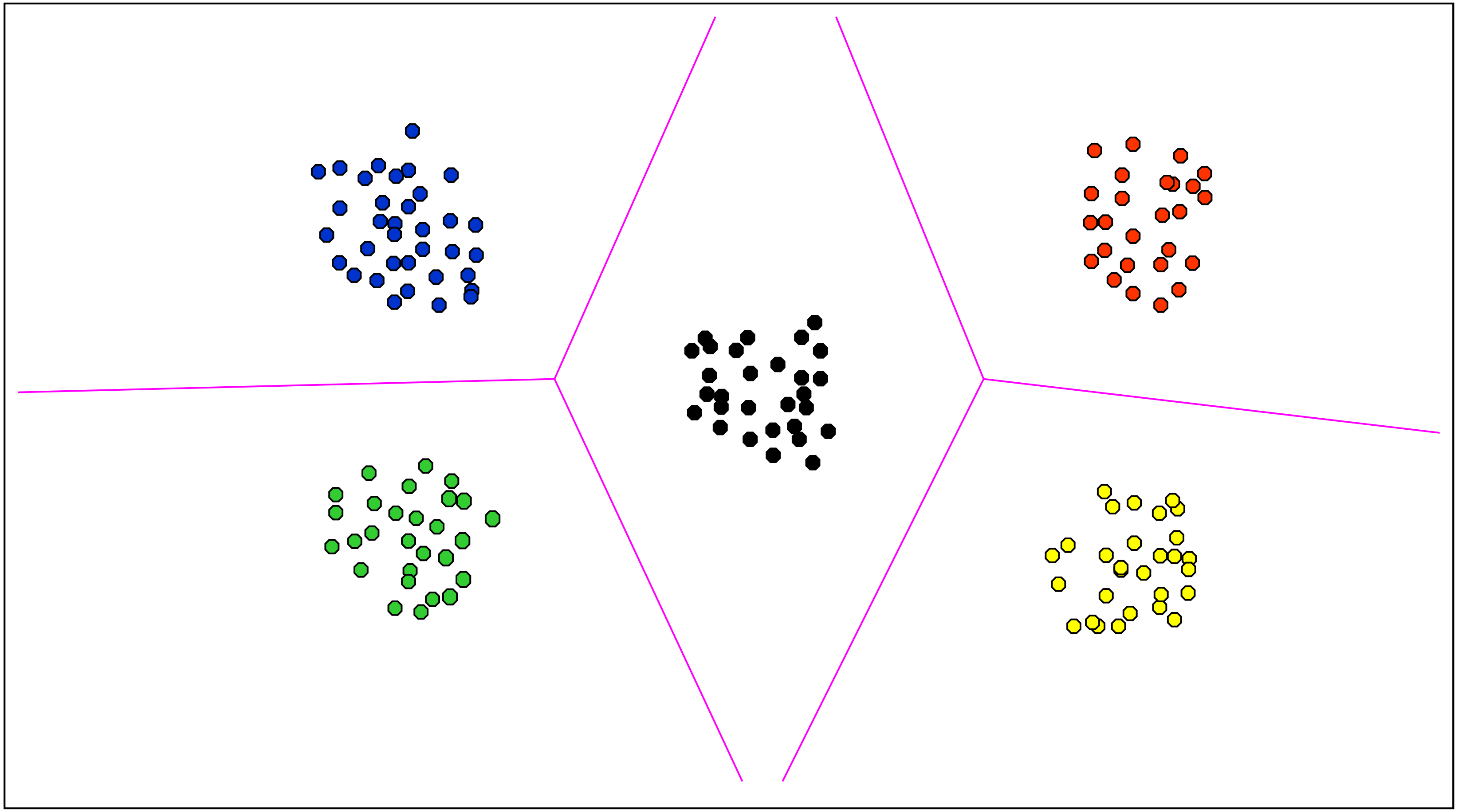
***For each* k:**

- ***Form the cluster* $\mathbf{B}_k$ *as the set of instances in* $\mathbf{B}$ *that are closer to* C_K than they are to other C_q *for all* $q \neq k$**
- ***For each* k, *recompute* C_K *as the center of cluster* $\mathbf{B}_k$ *(mean of the vectors in* $\mathbf{B}_k$*)***

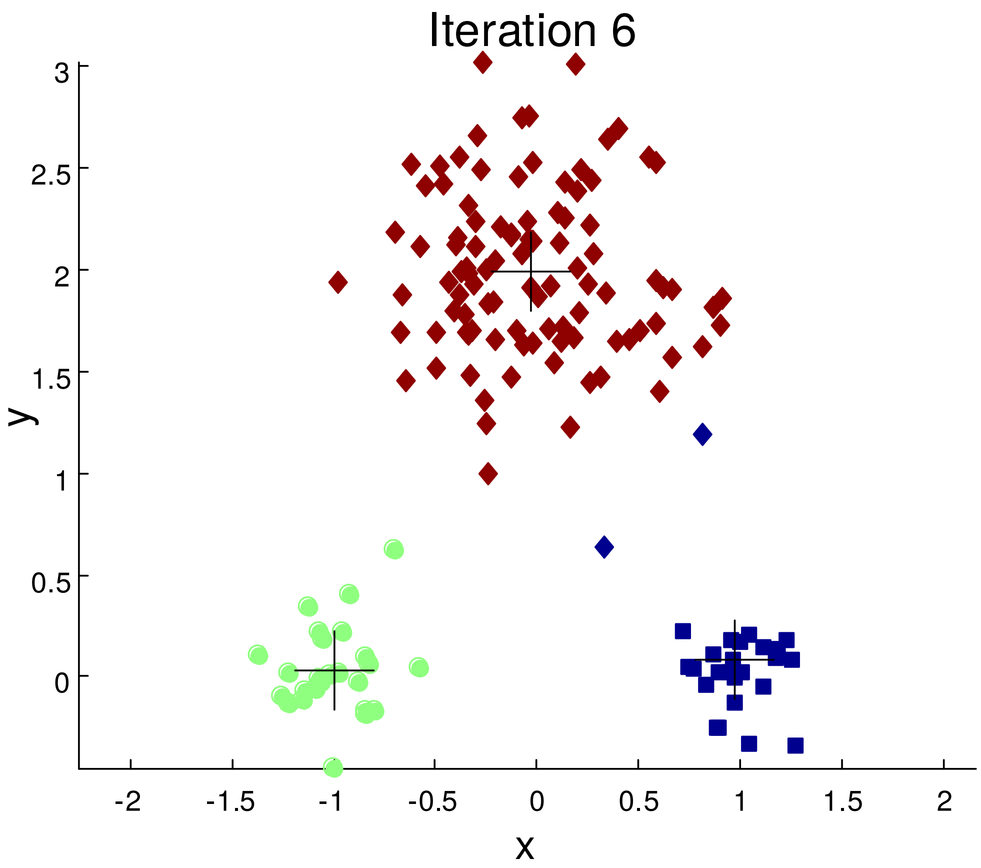
K-means algorithm: Example 1



K-means algorithm: Example 2



K-means algorithm: Example 3



K-means Evaluation

Strength

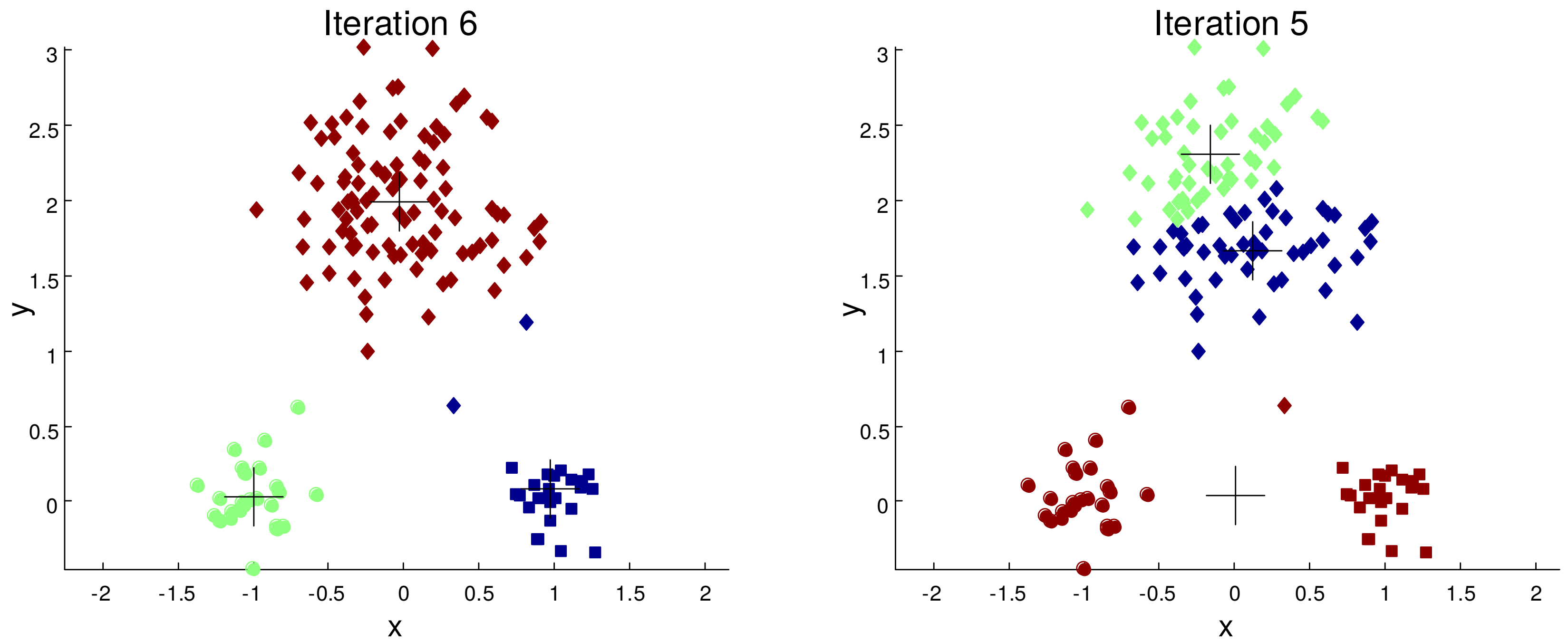
- ***Relatively efficient*: O(TKN), *where* N *is the* n^b *of instances*, K *is the* n^b *of clusters, and* T *is the* n^b *of iterations* (K, T << N)
- ***Guaranteed to converge*** *to at least a local optima*

Weakness

- *Applicable only when mean is defined (what about categorical data?)*
- *Need to specify K, the number of clusters, in advance*
- *Unable to handle noisy data and outliers*
- *Not suitable for clusters with non-convex shapes*
- *Very sensitive to initial centroids assignment*

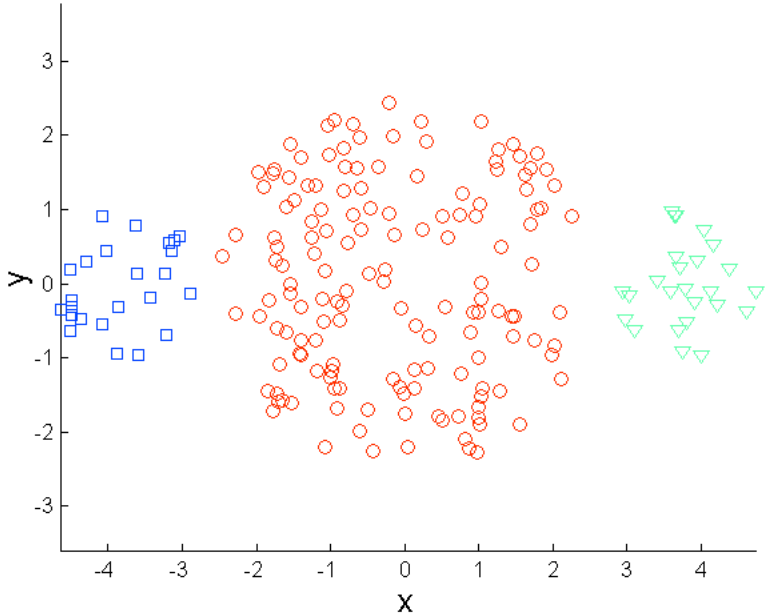
K-means algorithm: Importance of centroids initialization

Sensitivity to the initial random assignments

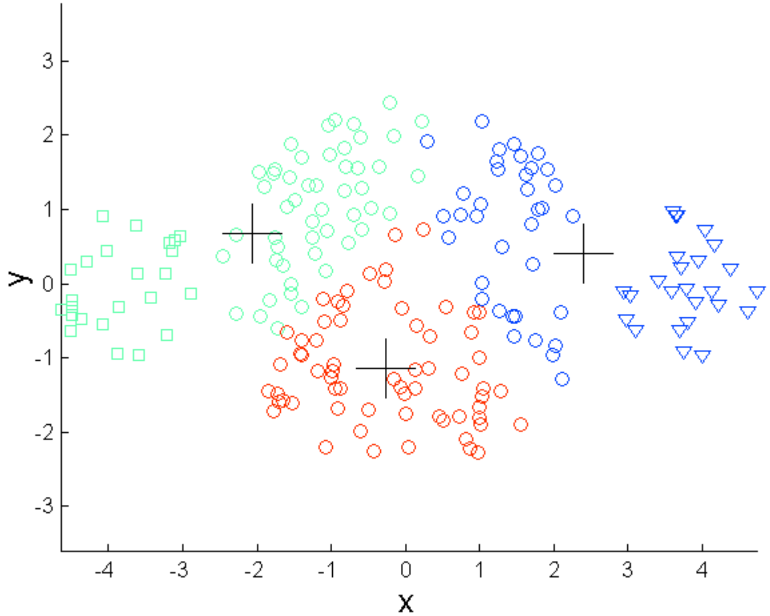


K-means algorithm: Size of instances classes

Sensitivity to the Size



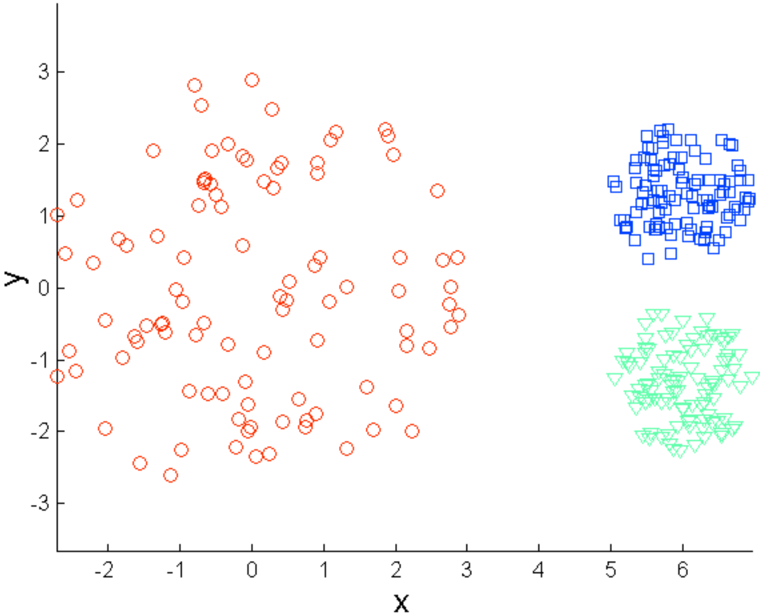
Original instances



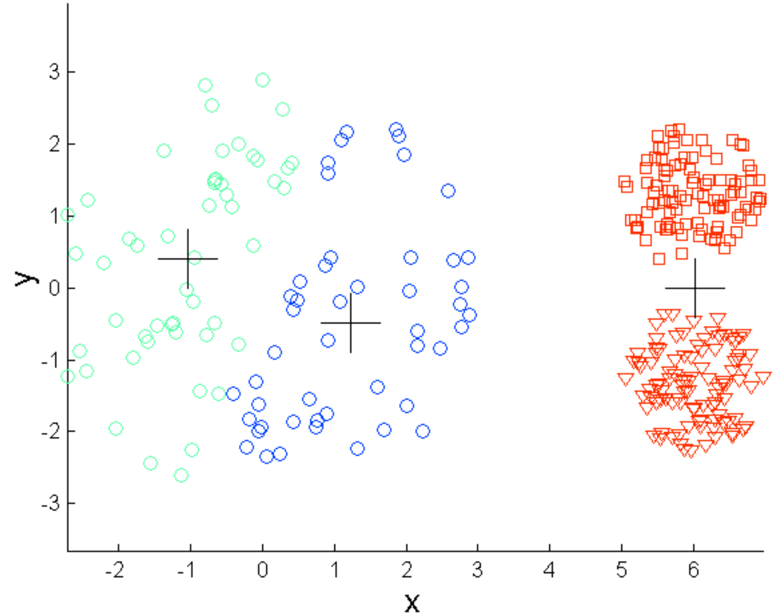
K-means (3 Clusters)

K-means algorithm: Density of instances

Sensitivity to the Density



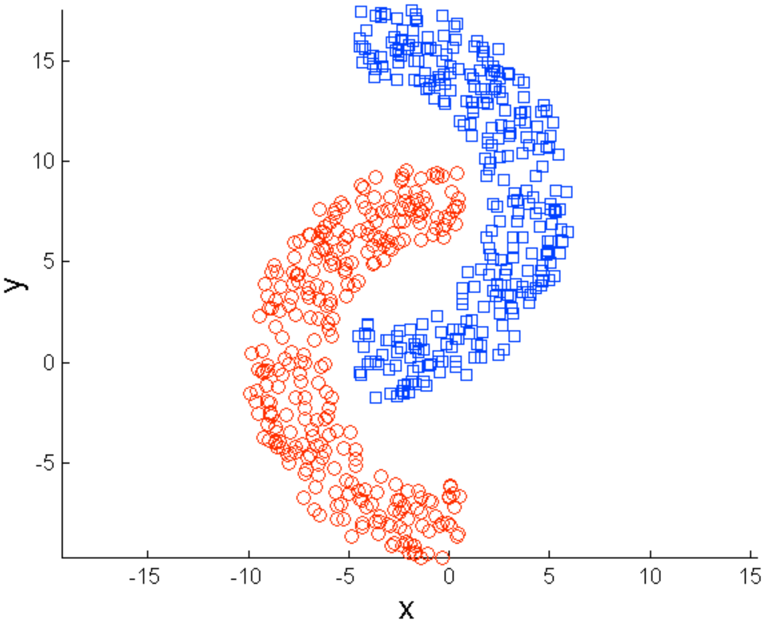
Original instances



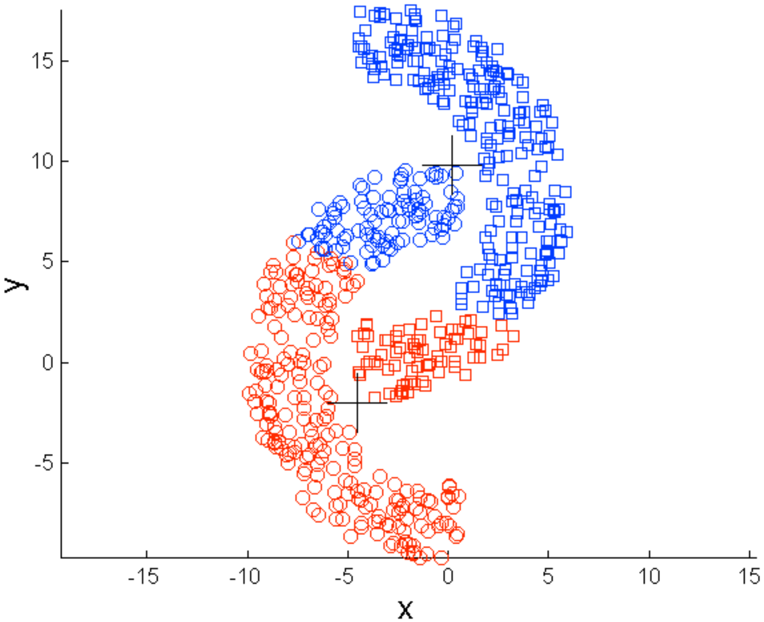
K-means (3 Clusters)

K-means algorithm: Non globular shapes

Sensitivity to the Shape



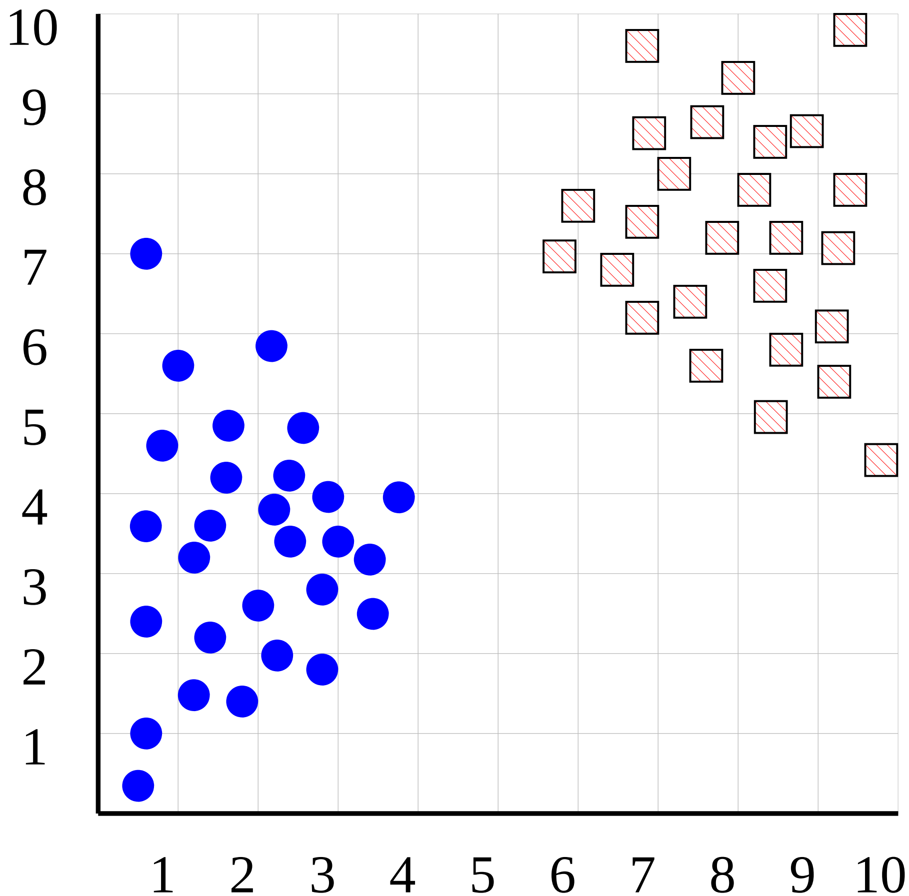
Original instances



K-means (2 Clusters)

How can we tell the right number of clusters?

In general, this is a unsolved problem. However there are many approximate methods…..

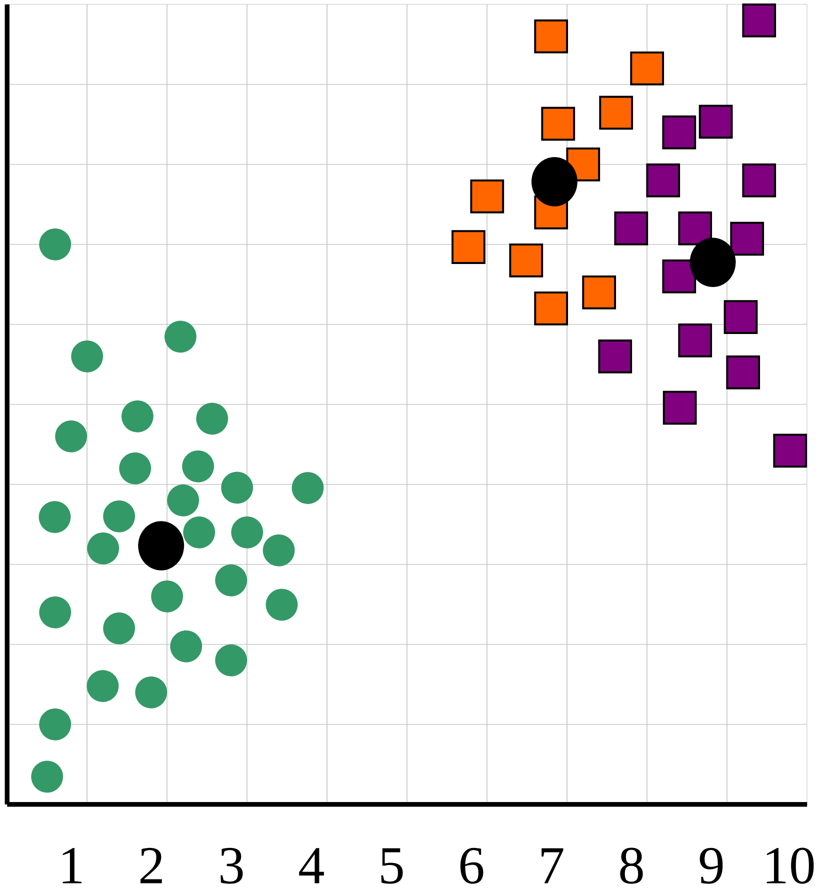


K-means algorithm

When k = 1, the objective function is 873.0

When k = 2, the objective function is 173.1

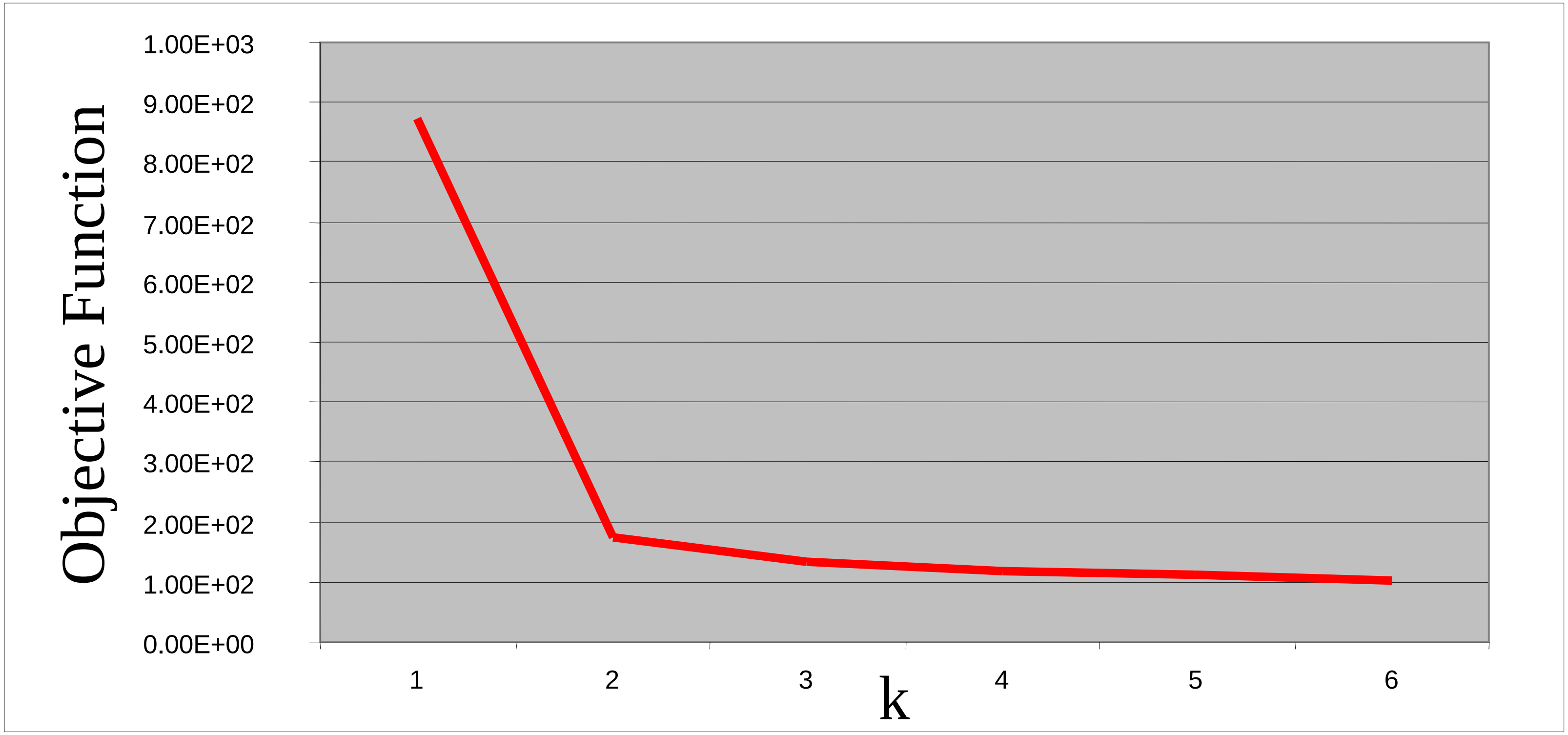
When k = 3, the objective function is 133.6



K-means algorithm

“Knee finding” or “Elbow finding” technique :

The abrupt change at k = 2, is highly suggestive of two clusters in the data



Semi-Supervised K-Means

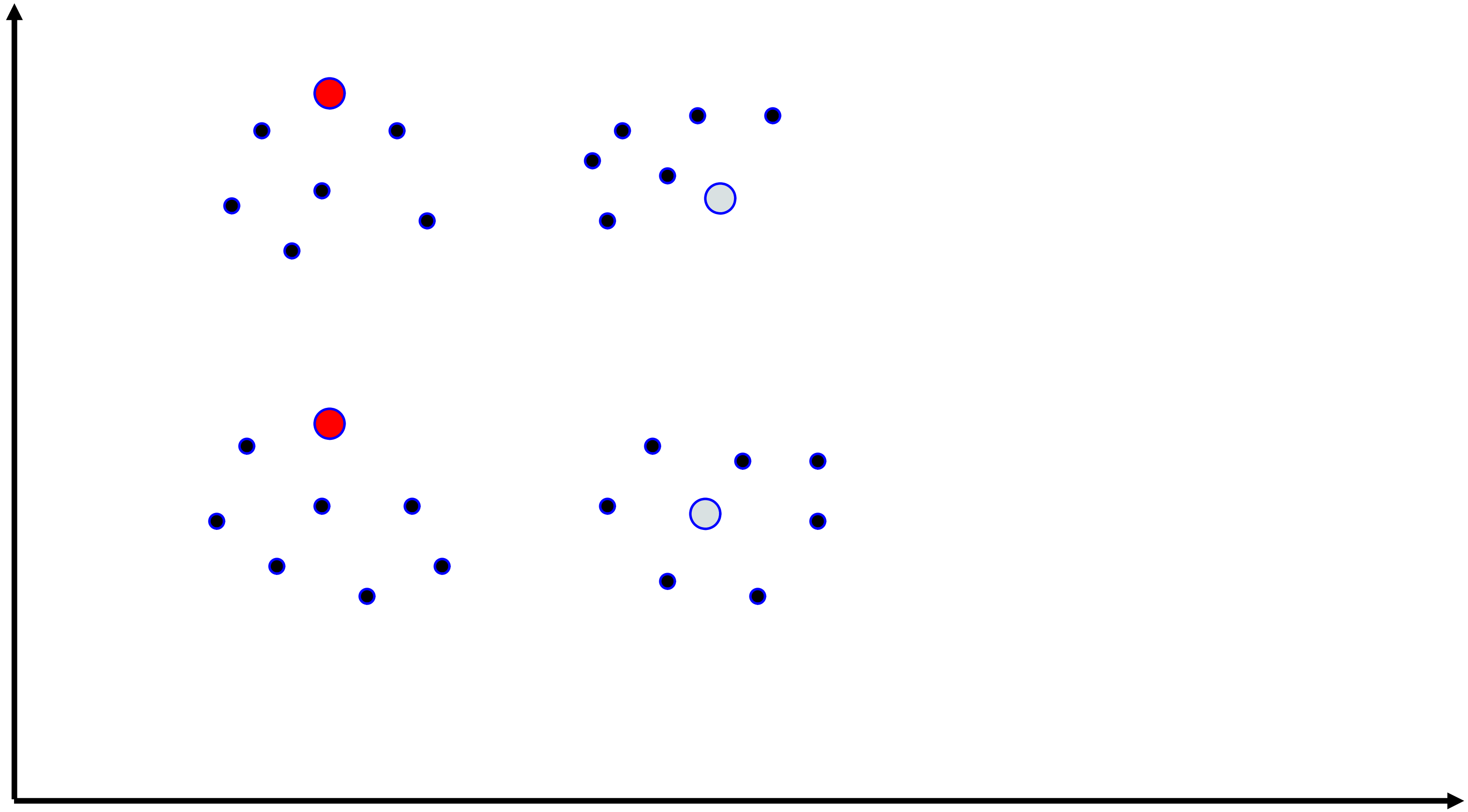
Seeded K-Means

- Labeled data provided by user are used for initialization
- Initial center for cluster *i* is the mean of the seed points having label *i*
- Seed points are only used for initialization, and not in subsequent steps

Constrained K-Means

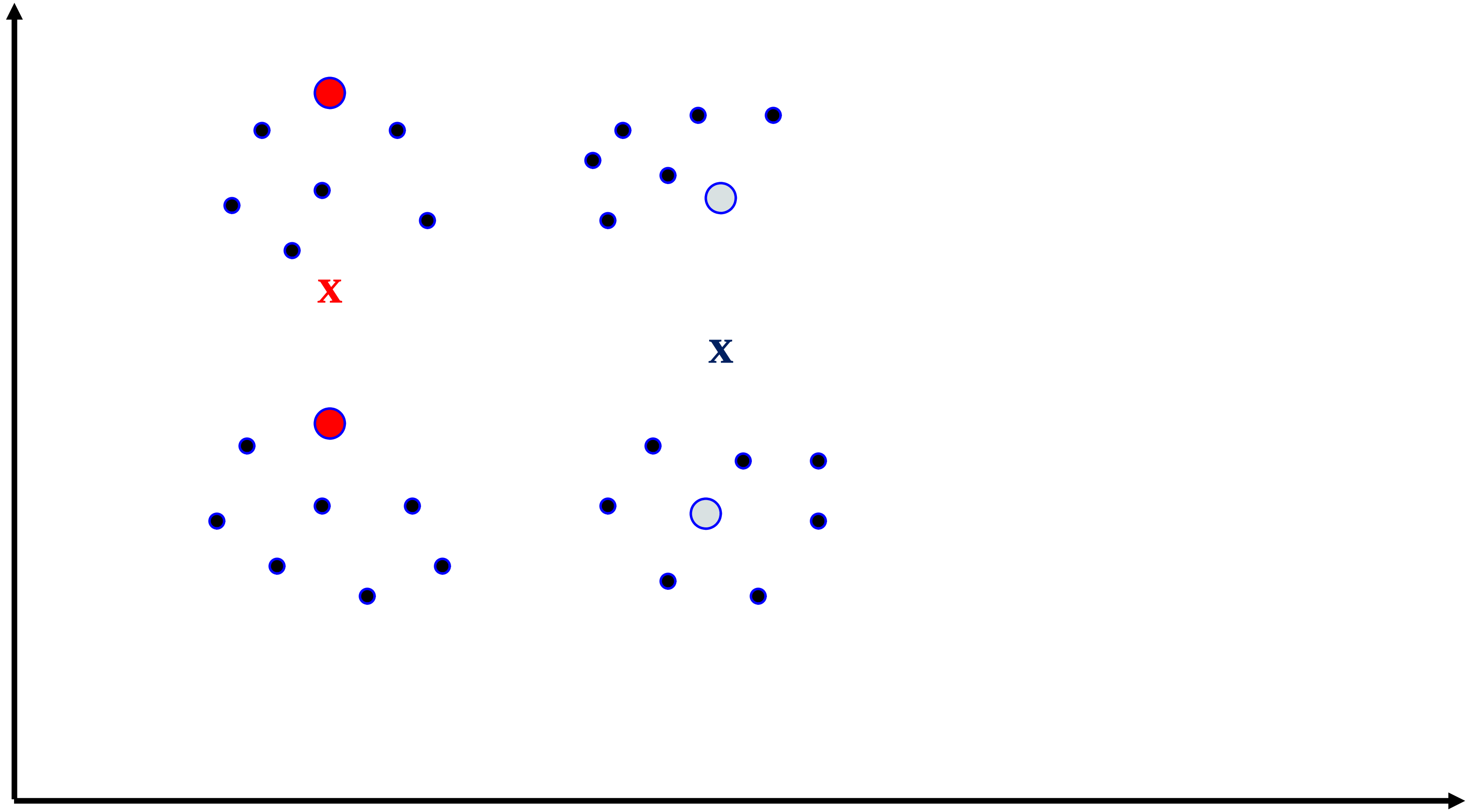
- Labeled data provided by user are used to initialize K-Means algorithm
- Cluster labels of seed data are kept unchanged in the cluster assignment steps, and only the labels of the non-seed data are re-estimated

Semi-Supervised K-Means Example :



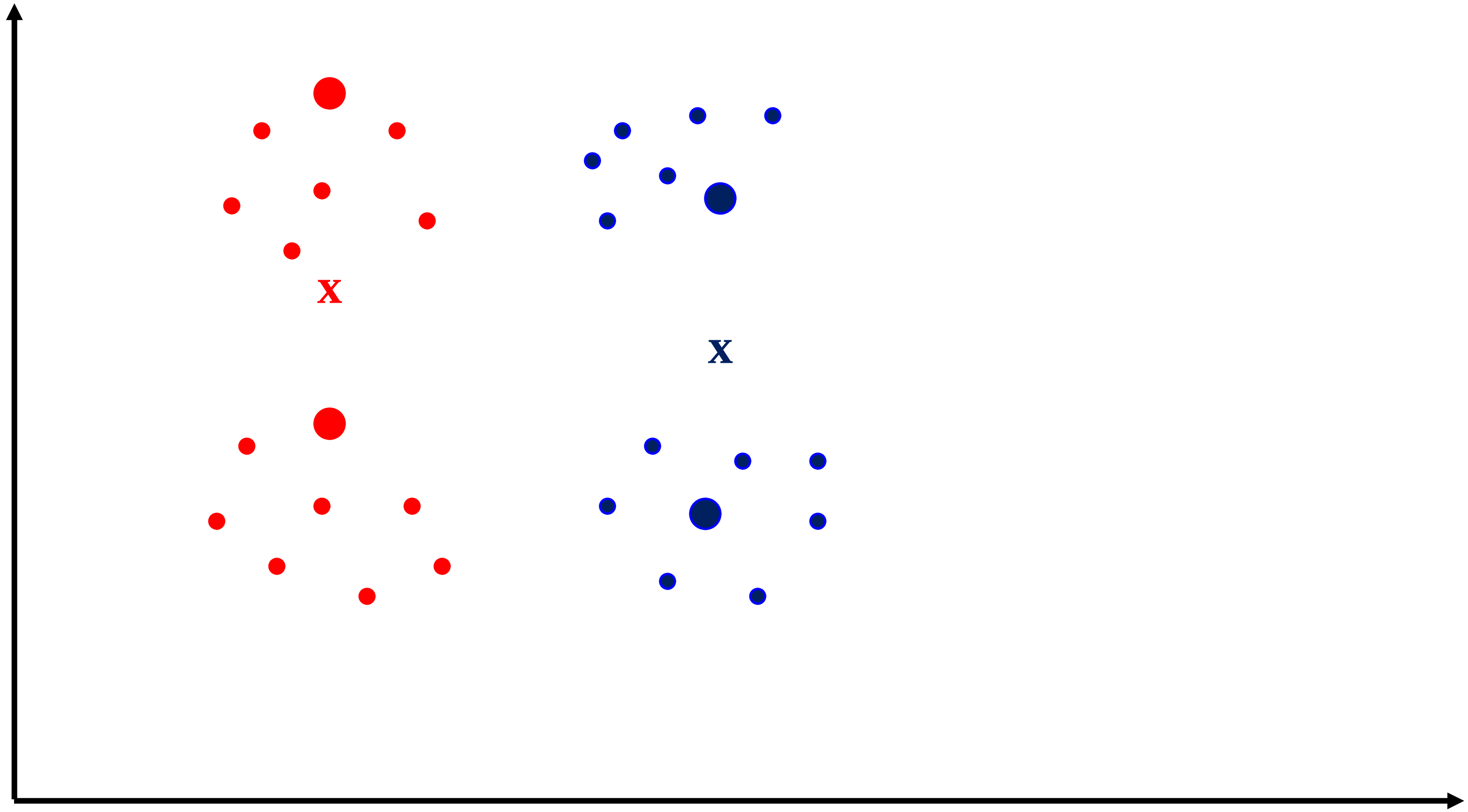
Semi-Supervised K-Means Example :

INITIALIZE MEANS USING LABELED DATA



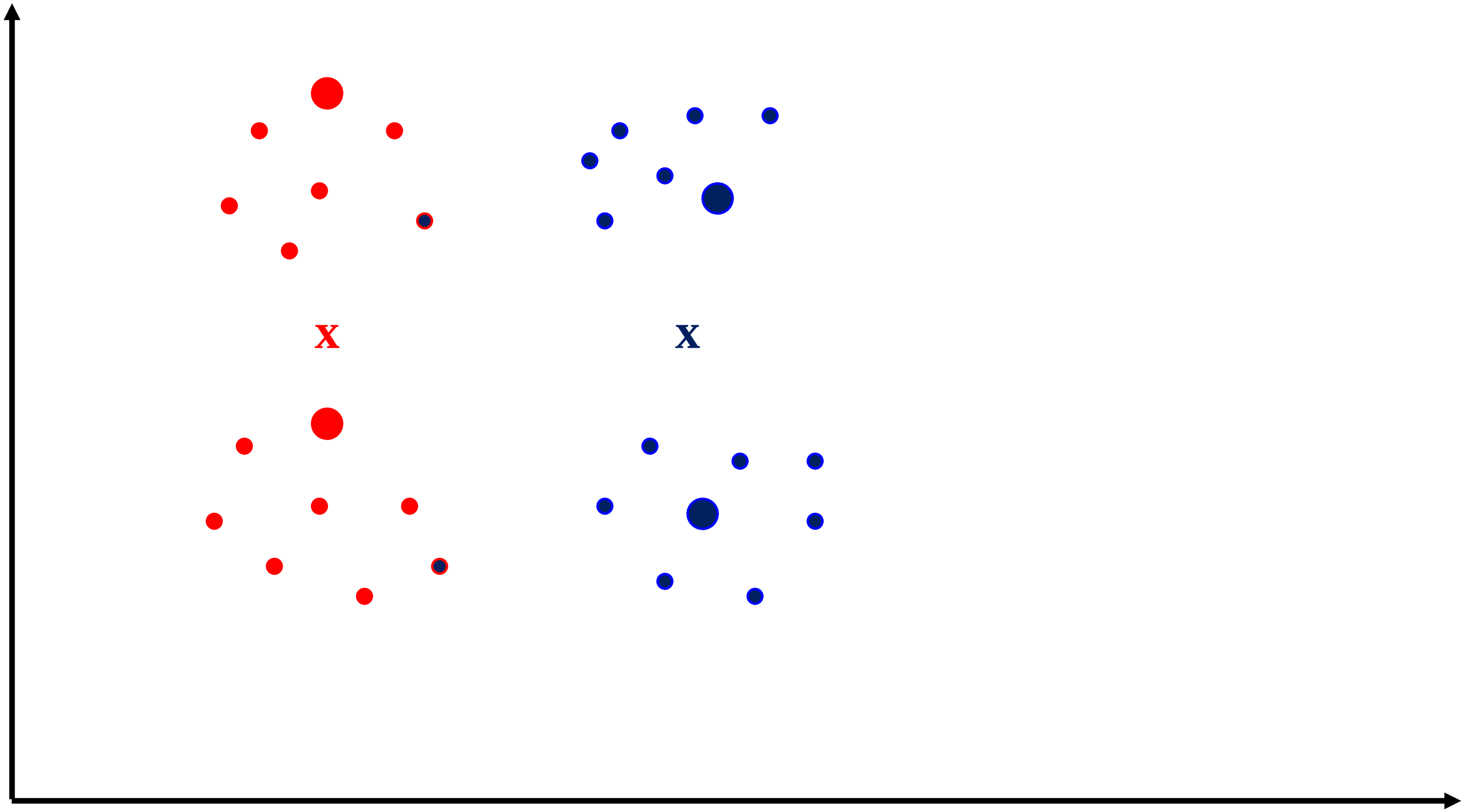
Semi-Supervised K-Means Example :

ASSIGN INSTANCES TO CLUSTERS



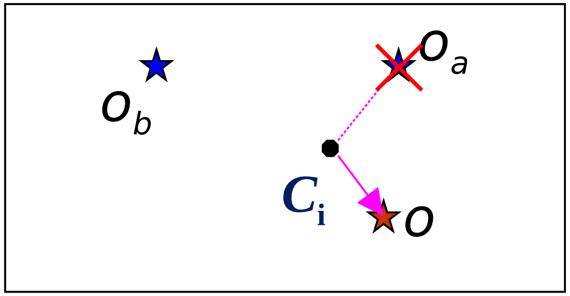
Semi-Supervised K-Means Example :

RE-ESTIMATE MEANS & ITERATE



K-medoids : ***A variant from K-means algorithm***

Idea*: *Avoid convergence problems by restricting centroids to coincide with the instances (Cluster C_i represented by representative instance o_i, the medoid)

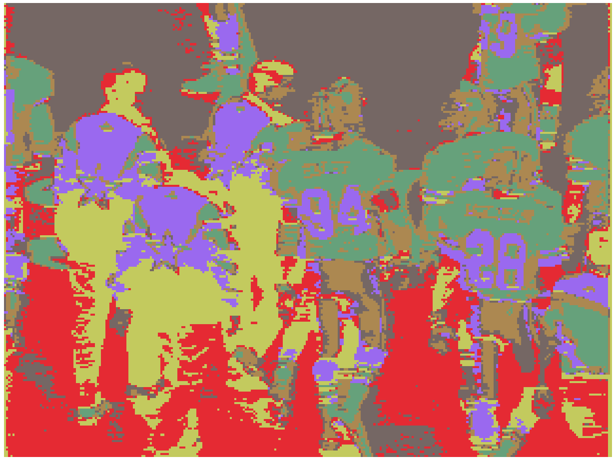


C_i reassigned to *o*

ISODATA algorithm

1. ***Select several cluster means and form clusters***
2. ***Split any cluster whose variance is too large***
3. ***Group together clusters that are too small***
4. ***Recompute clusters' means***
5. ***Repeat till 2 and 3 cannot be applied***

ISODATA algorithm



K-*means*, K=6

Original

Isodata,
K *became* 5

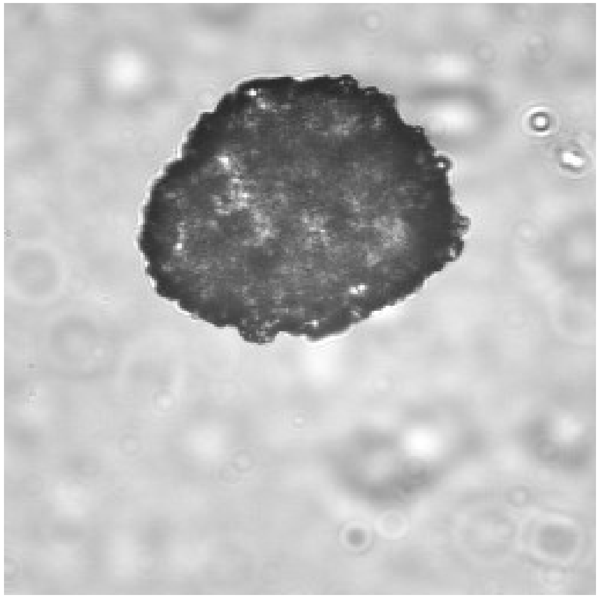
5. K-Means algorithm applications

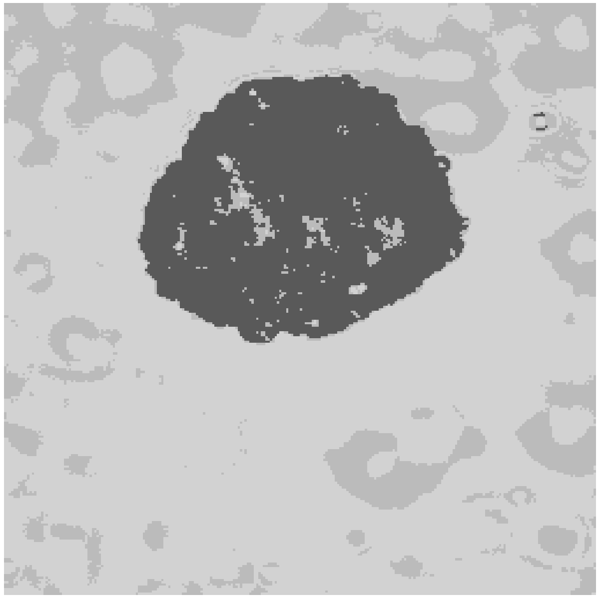
Image Segmentation

Breaking up the image into meaningful or perceptually similar regions

Image Segmentation

X : Pixel's Grey level


Original Image


K=3

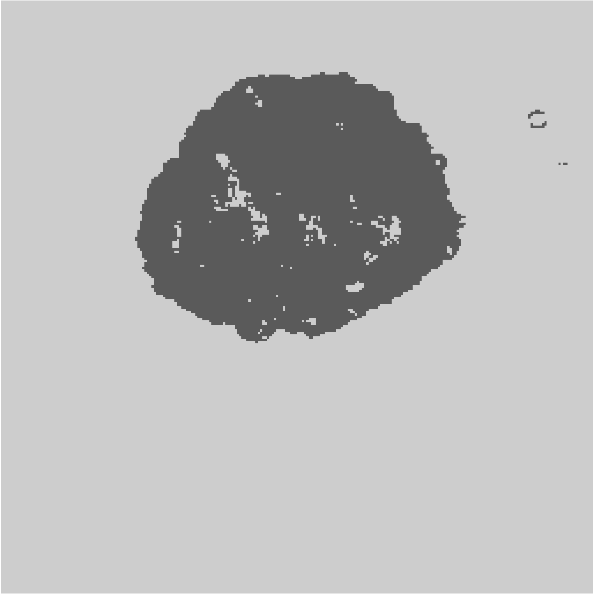
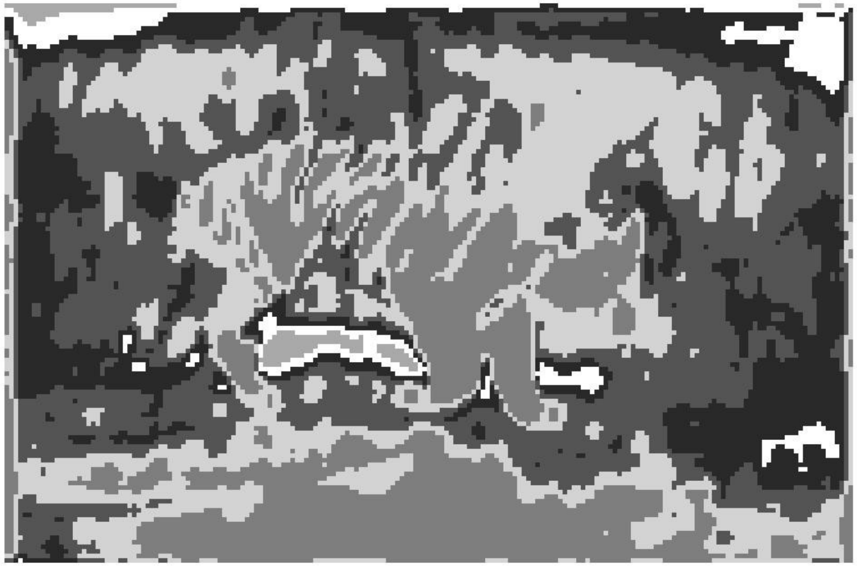

K=2

Image Segmentation

X : Pixel's color level (i.e., 3 grey level features)

Original Image



K=5

Image Segmentation

X : Pixel's color level (i.e., 3 grey level features)

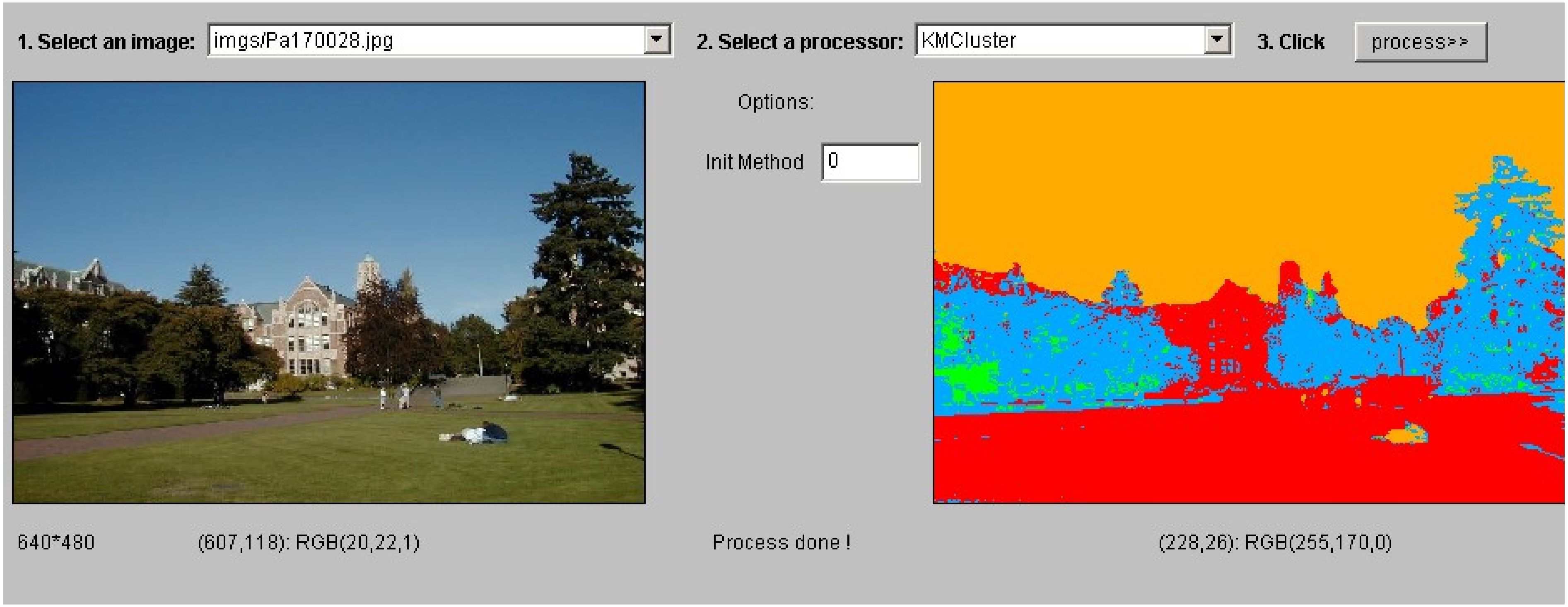
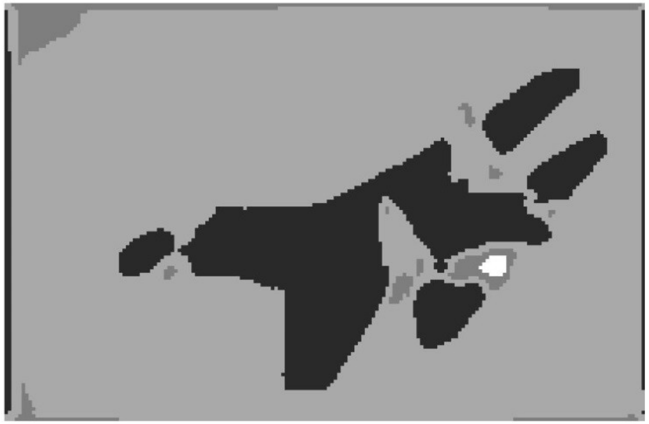


Image Segmentation

X : Feature vector computed on L x L *image sub-blocks*



Original Image | **5 x 5 *image sub-blocks*** | **10x10 *image sub-blocks***

K-means algorithm Applications

Image Segmentation

50x50

Image Segmentation

Image Compression

Clustering is related to vector quantization

- *Dictionary of vectors (the cluster centers)*
- *Each original instance represented using a dictionary index*
- *Each center "claims" a nearby region (Voronoi region)*

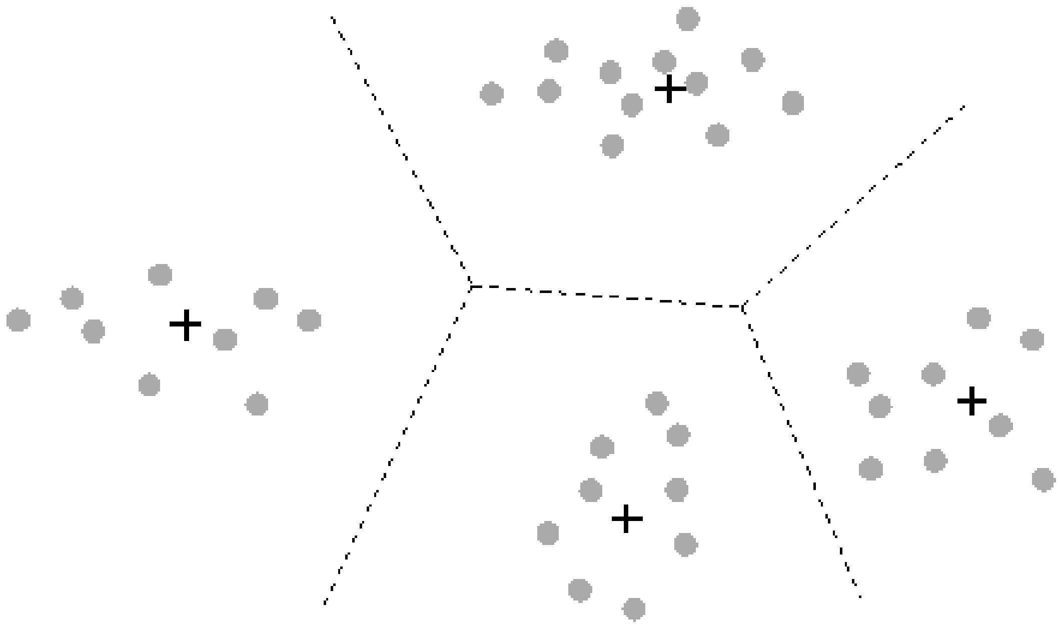
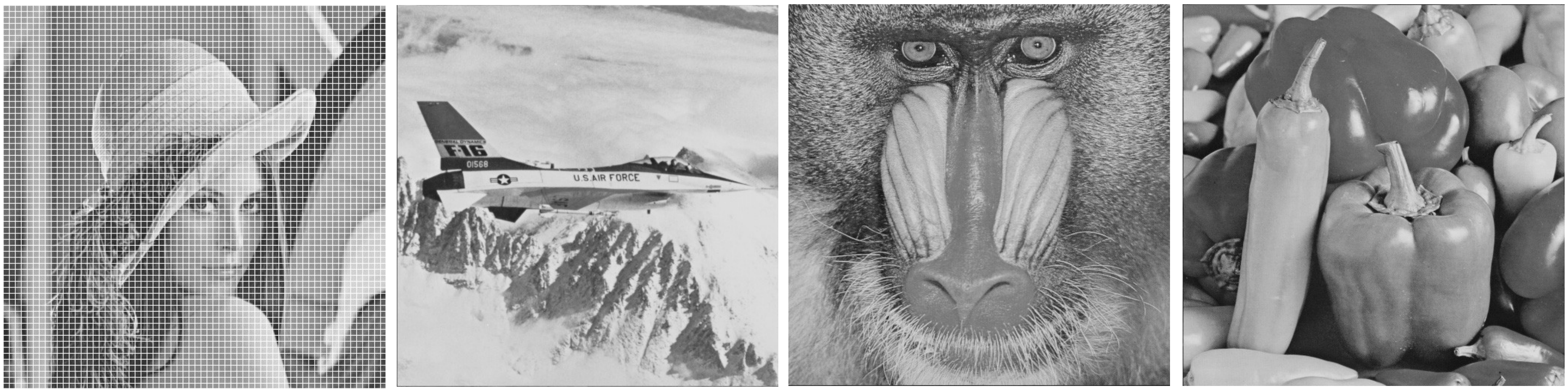
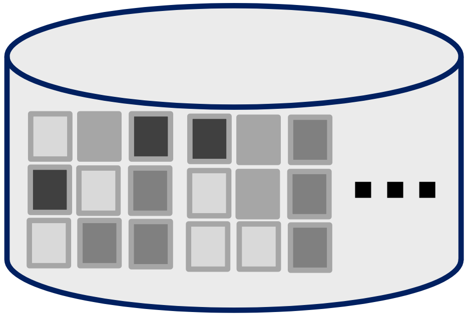


Image Compression

Training Data : Set of L x L *sub-blocks from 4 training images*





K-means algorithm Applications

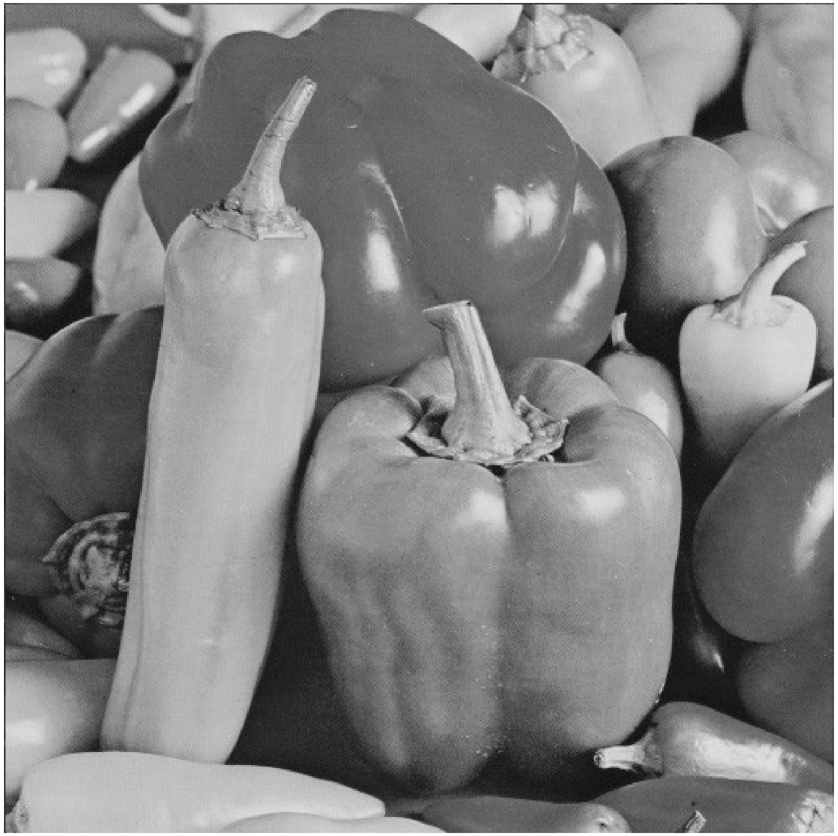
Image Compression

Original **Decoded image, psnr: 31.32**

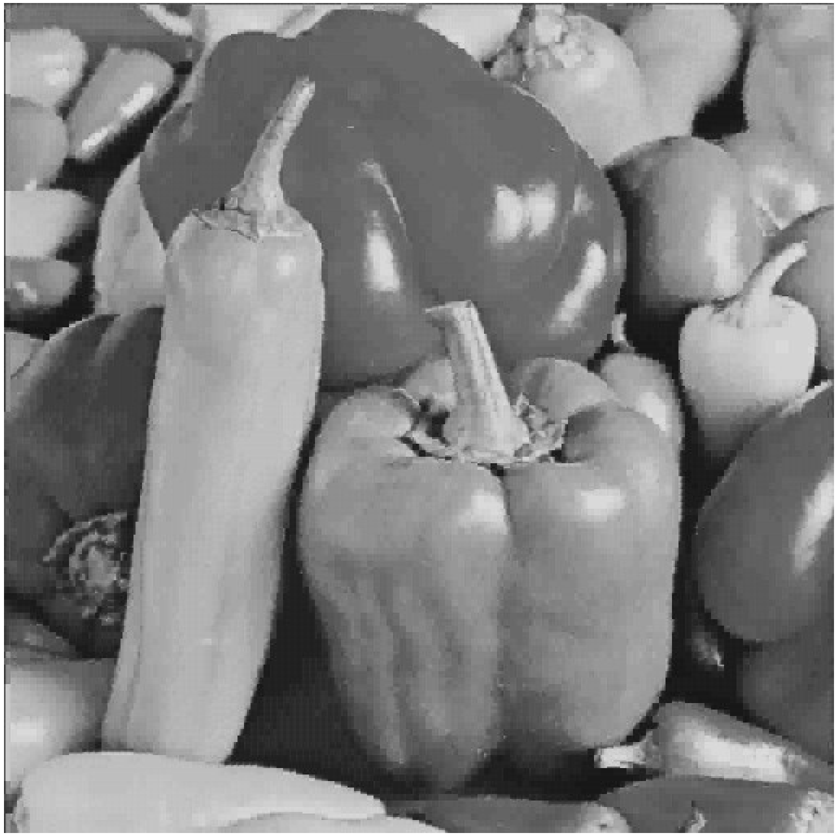
K-means algorithm Applications

Image Compression

Original



Decoded image, psnr: 30.86

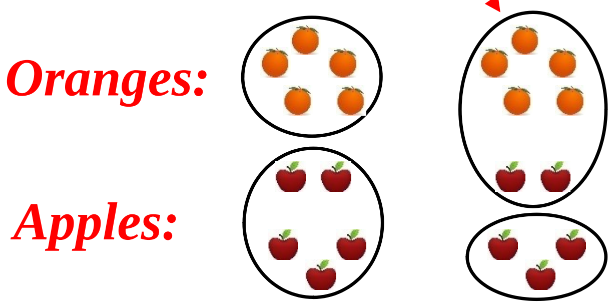


6. Quality of clustering

Quality of clustering

When training instances are labelled, (Class labels known for ground truth): several quality measures can be used: Accuracy, precision, recall…

Precision = 5/5 = 100%
Recall = 5/7 = 71%



Precision = 3/5 = 60%
Recall = 3/3 = 100%

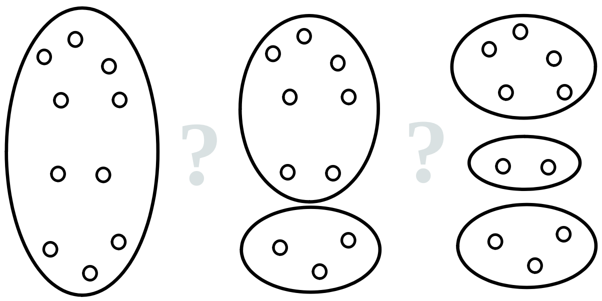
Quality of clustering

A <u>good clustering</u> method will produce high quality clusters :

- High <u>intra-class</u> similarity: cohesive within clusters
- Low <u>inter-class</u> similarity: distinctive between clusters

<u>Internal Measures</u>

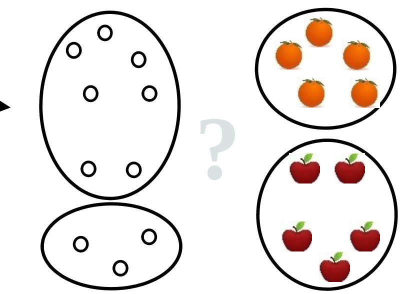
- *Validate without external info*
- *With different number of clusters*
- *Solve the number of clusters*



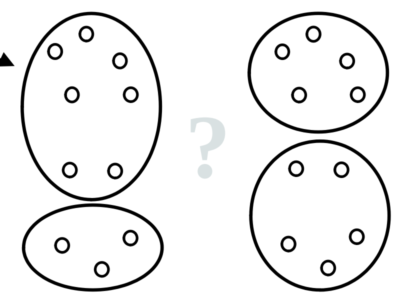
Quality of clustering

External Measures

- *Validate against ground truth*



- *Compare two clusters:*
  (how similar)



Quality of clustering

Cluster tightness (or homogeneity) measure:

$$Q = \sum_{k} \frac{1}{|\mathbf{B}_k|} \sum_{X \in \mathbf{B}_k} \|X - C_k\|^2$$

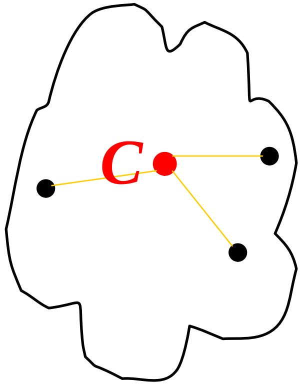
$|\mathbf{B}_k|$ *is the number of data instances in cluster* **k**

Q *will be small if (on average) the data instances in each cluster are close*

The Q measure takes into account homogeneity within clusters, but not separation between clusters

Silhouette coefficient

Cohesion: ***measures how closely related are objects in a cluster***

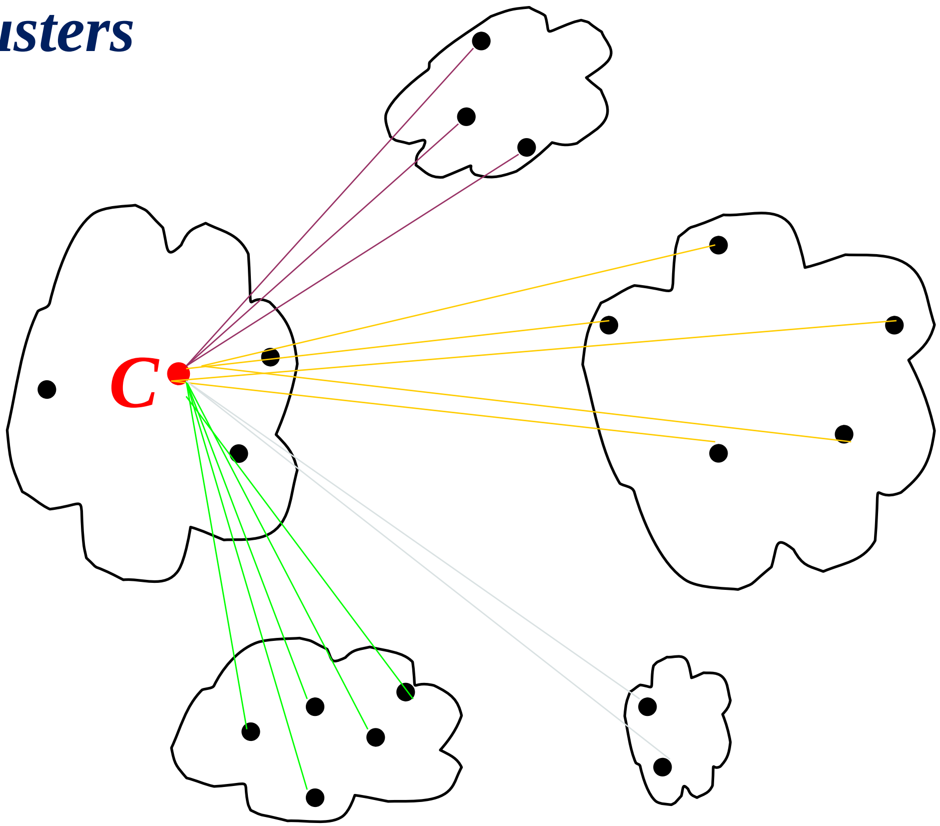


Cohesion

$a(C)$: average distance of C to all other vectors in the same cluster

Silhouette coefficient

- **Separation**: ***measure how distinct or well-separated a cluster C is from other clusters***



Silhouette coefficient

Silhouette S(C):

$$S(C) = \frac{b(C) - a(C)}{Max(a(C), b(C))}$$

Silhouette Coefficient S:

$$S = \frac{1}{K} \sum_{K=1}^{K} S(C_K)$$

$S(C)$, $S \in [-1, +1]$: -1=Bad, 0=Indifferent, 1=Good