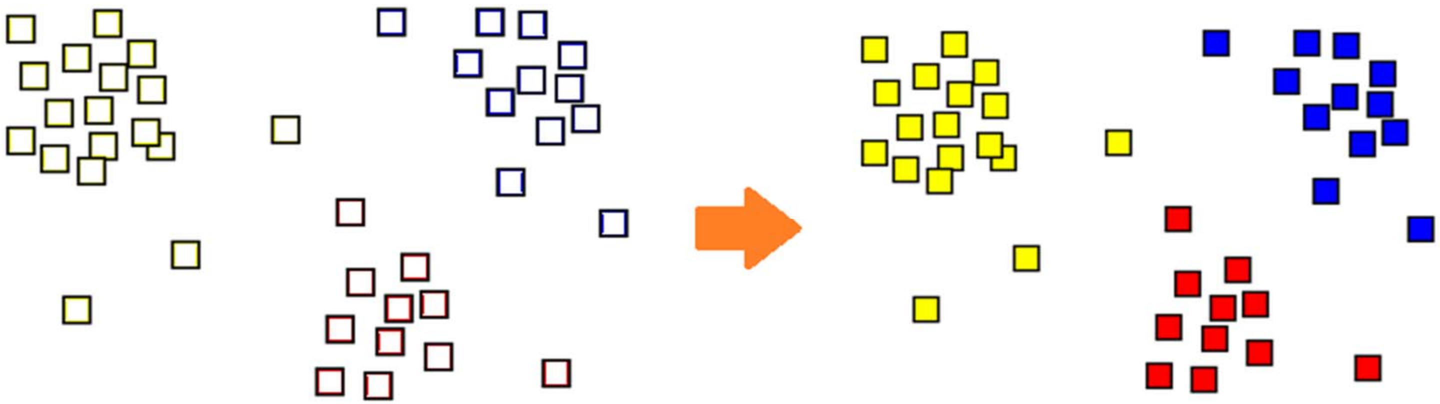
Contents

- Problem definition
- Applications
  - Scene clustering / Segmentation

- Major & Trends clustering algorithms
  - Kmeans clustering
  - Hierarchical clustering
  - Ensemble clustering
  - Large scale clustering (→ accelerated clustering)
    - Tree indexed kmeans
    - Elkan kmeans
    - Kmeans++

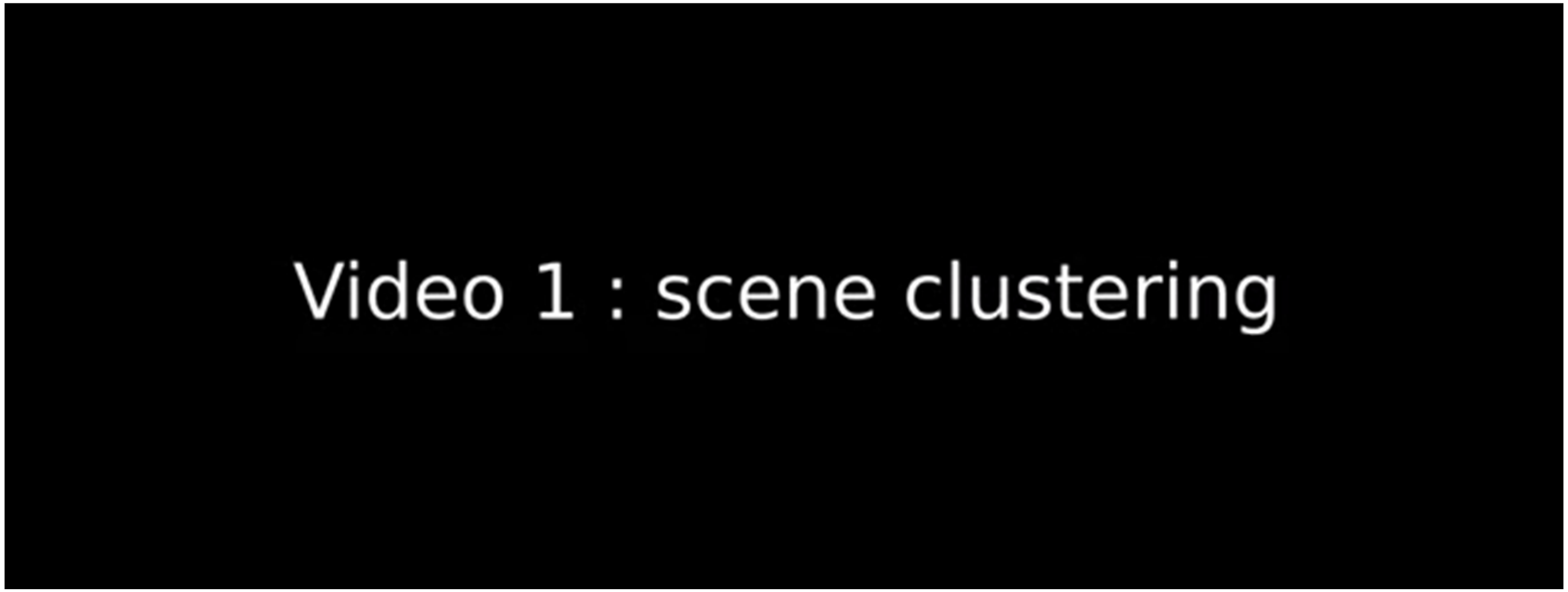
Clustering

- **Cluster analysis** or **clustering** is the task of assigning a set of objects into groups (called **clusters**) *- from Wikipedia-*



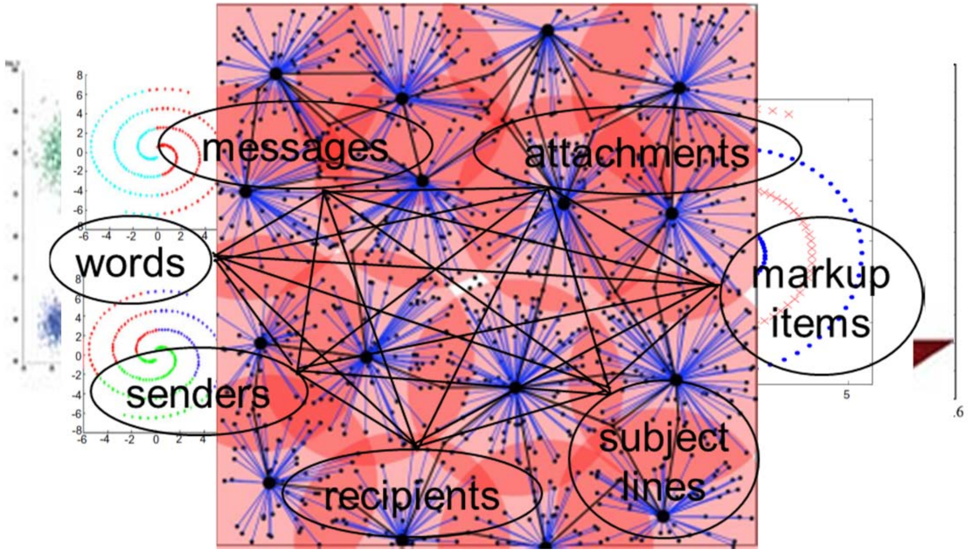
The result of a cluster analysis shown as the coloring of the squares into three clusters

Applications



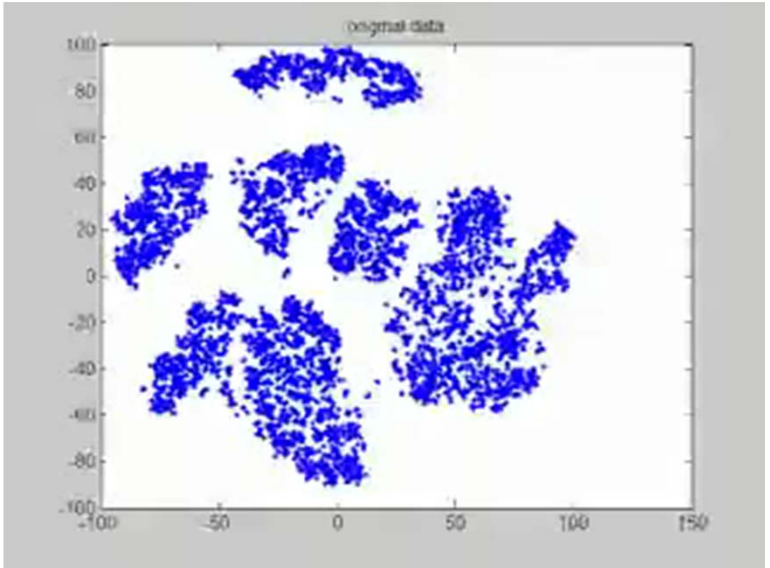
Major approaches to clustering

- K-means and its variants
- Hierarchical clustering
- Density-based clustering
- Clustering ensembles
- Large scale clustering
- Multi-way clustering



K-means

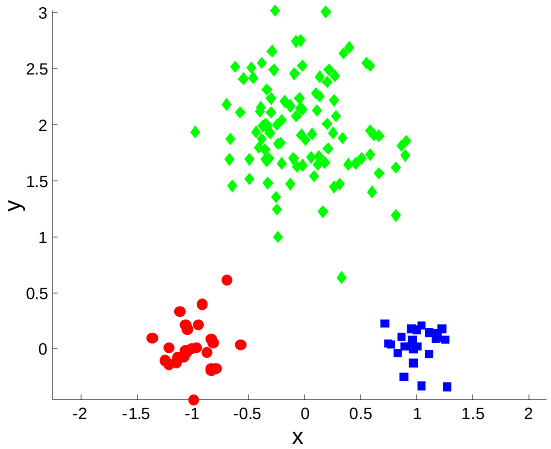
$$J_1 = \sum_{i}^{N} \sum_{j}^{K} r_{ij} \| x_i - c_j \|^2$$



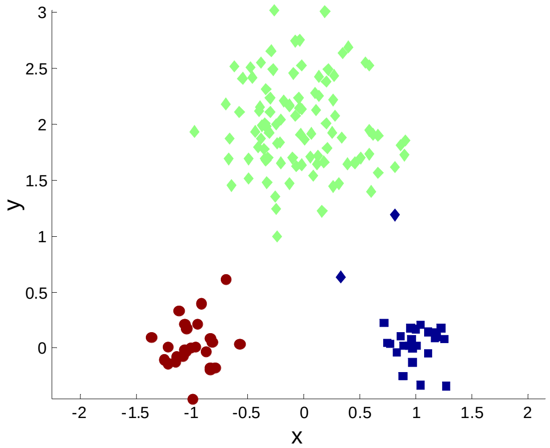
Issues and Limitations for K-means

How to choose initial centers?
How to choose K?

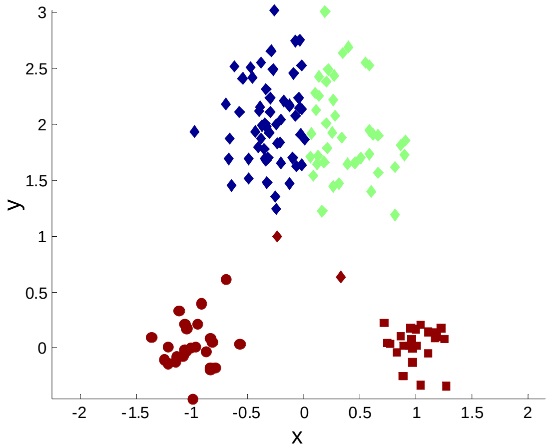
Two different K-means Clustering



Original Points



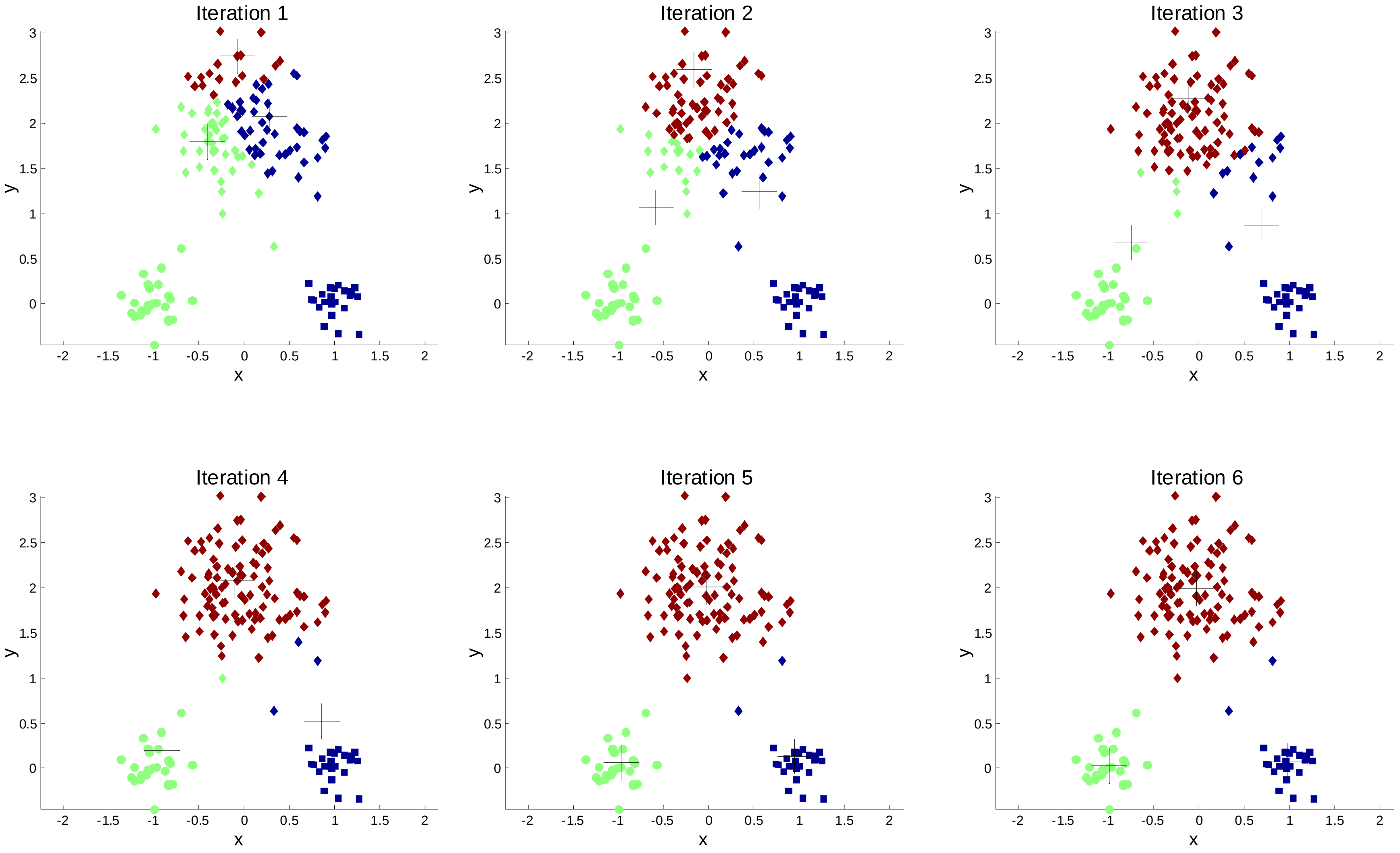
Optimal Clustering



Sub-optimal Clustering

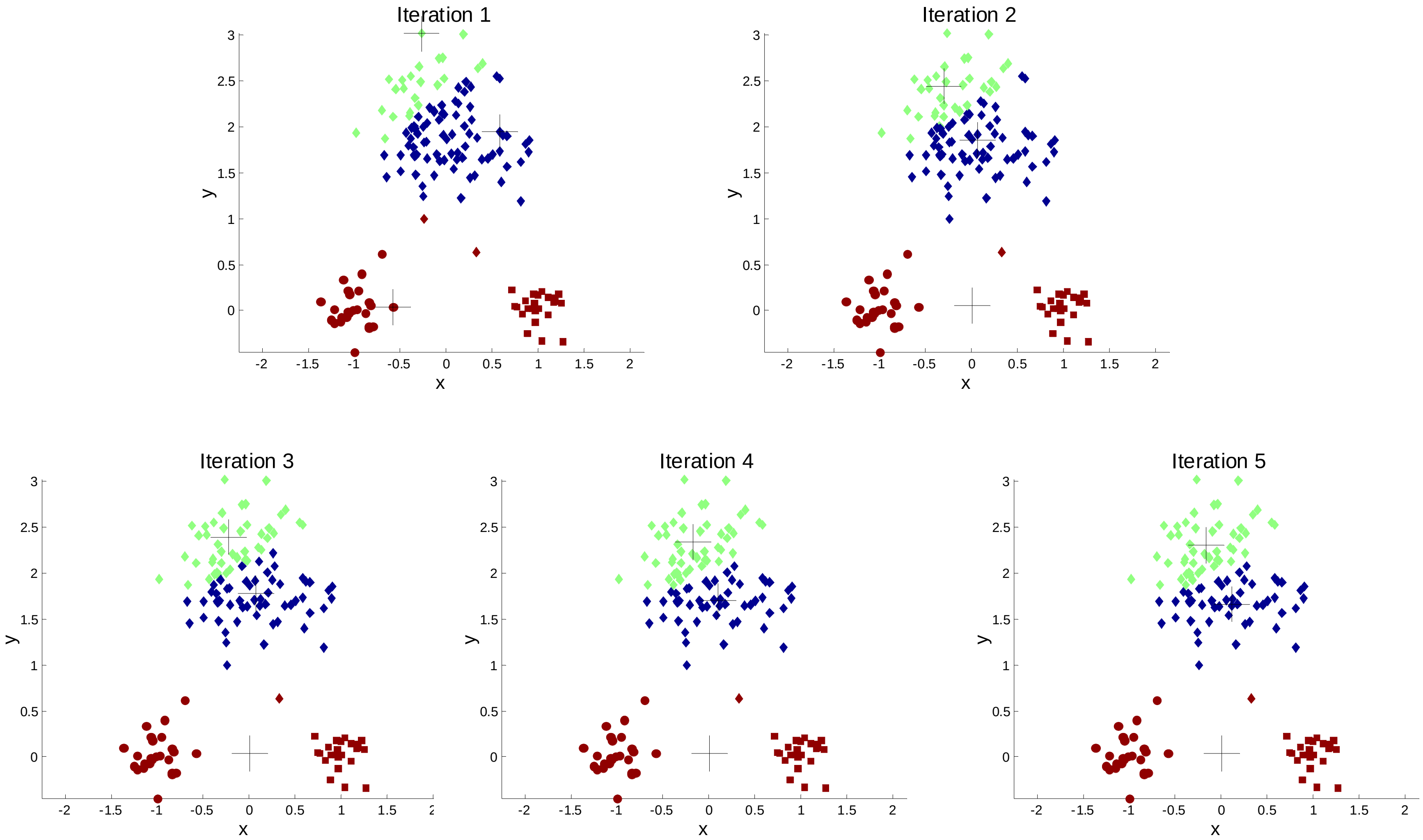
Importance of Choosing Initial Centroids

Good initial centroids



Importance of Choosing Initial Centroids

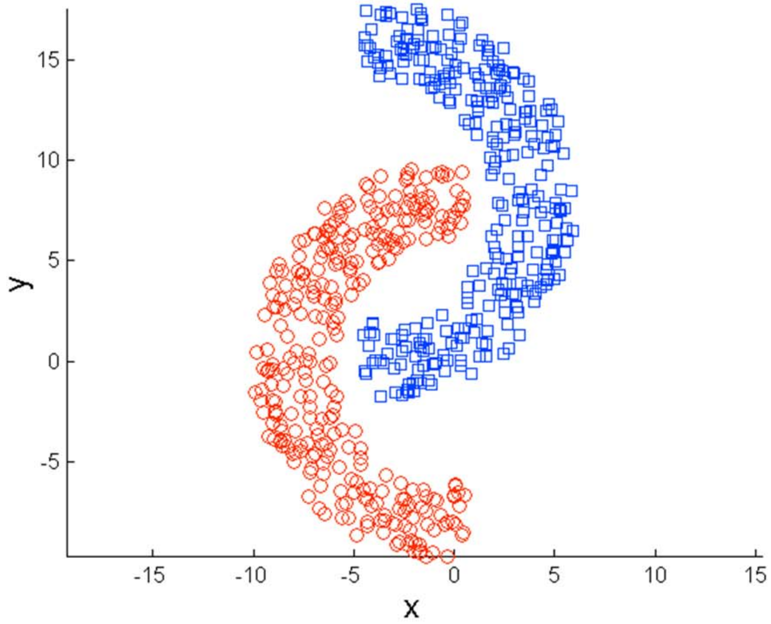
Bad initial centroids



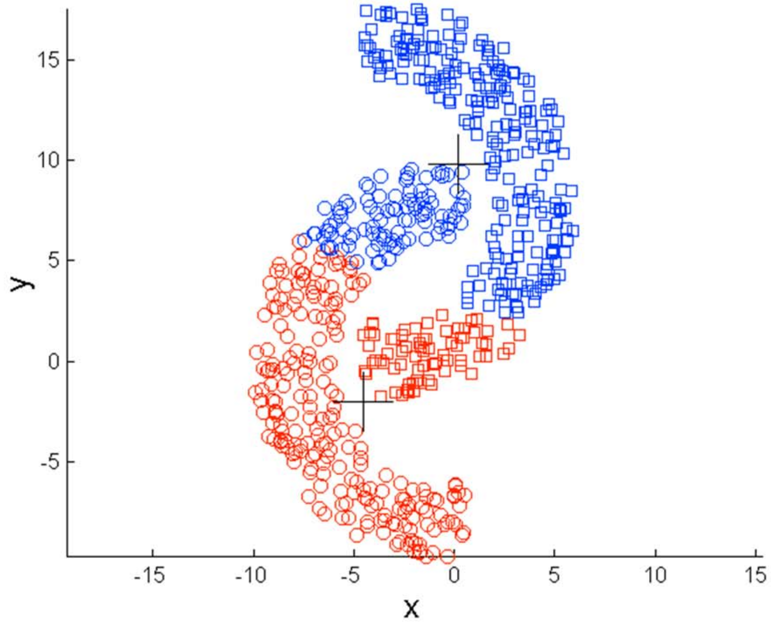
Solutions to Initial Centroids Problem

- Multiple runs
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering
- Select most widely separated

Limitation of kmeans: structured data



Original Points

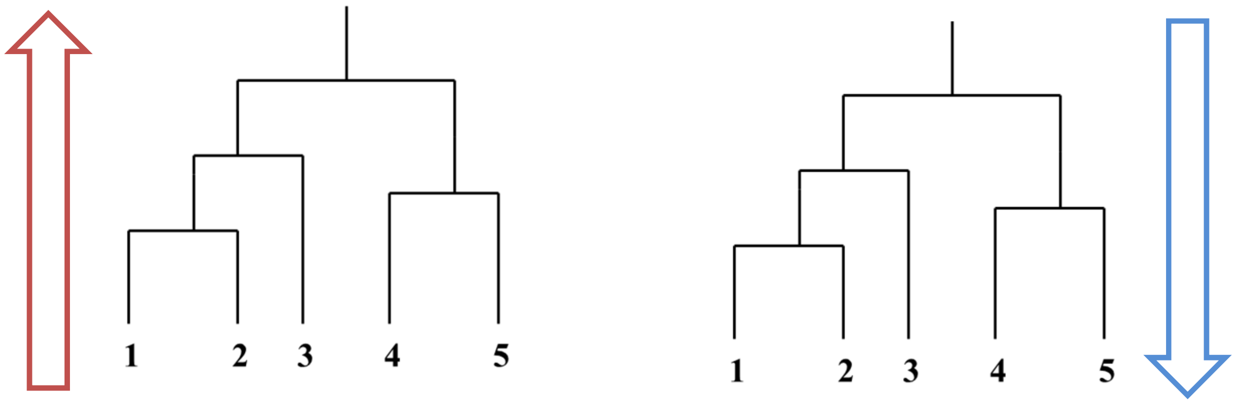


K-means (2 Clusters)

Hierarchical Clustering

- **Agglomerative**:
  - Start with the points as individual clusters
  - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left

- **Divisive**:
  - Start with one, all-inclusive cluster
  - At each step, split a cluster until each cluster contains a point (or there are k clusters)



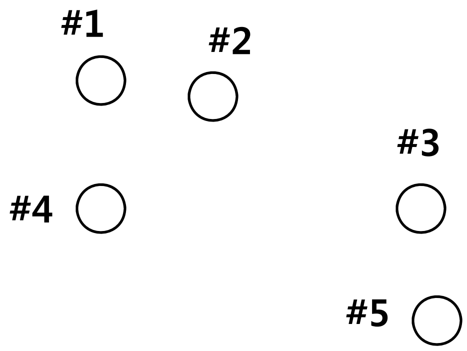
Agglomerative Clustering Algorithm

1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. **Merge** the two closest clusters
5. **Update** the **proximity matrix**
6. **Until** only a single cluster remains

Proximity matrix

Proximity matrix (5x5)

$$\begin{bmatrix} 0 & d(1,2) & d(1,3) & d(1,4) & d(1,5) \\ d(2,1) & 0 & d(2,3) & d(2,4) & d(2,5) \\ d(3,1) & d(3,2) & 0 & d(3,4) & d(3,5) \\ d(4,1) & d(4,2) & d(4,3) & 0 & d(4,5) \\ d(5,1) & d(5,2) & d(5,3) & d(5,4) & 0 \end{bmatrix}$$



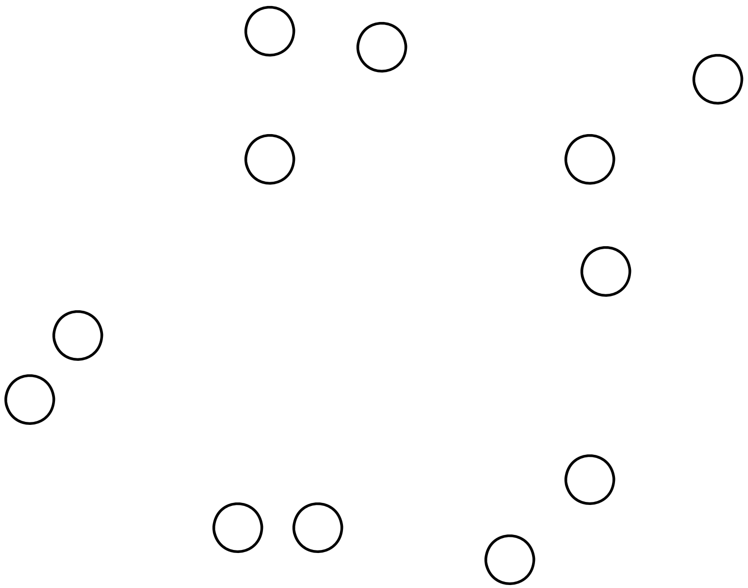
d(i,j)=difference/dissimilarity between i and j

Different Proximity measures

Distance metric

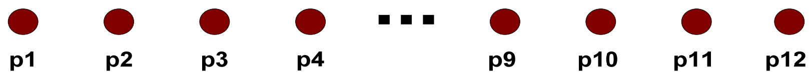
Example) Euclidean distance, Manhattan distance, etc

Input/ Initial setting

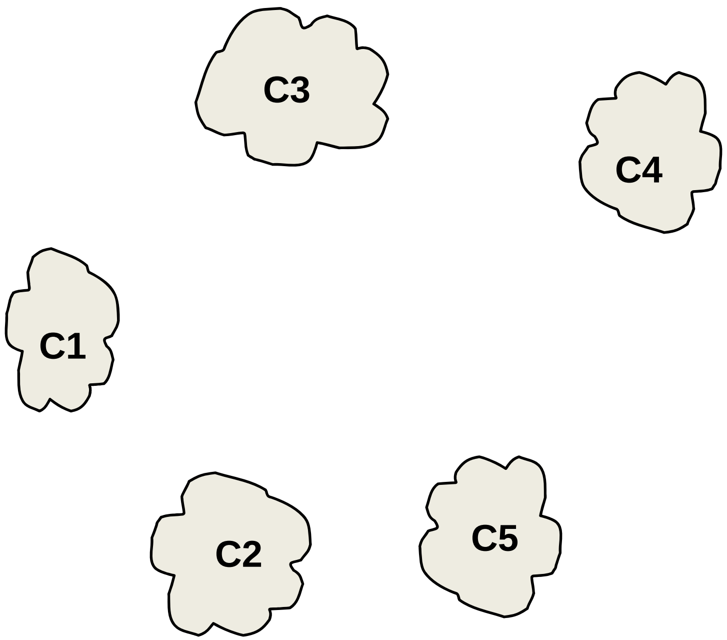


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Distance/Proximity Matrix

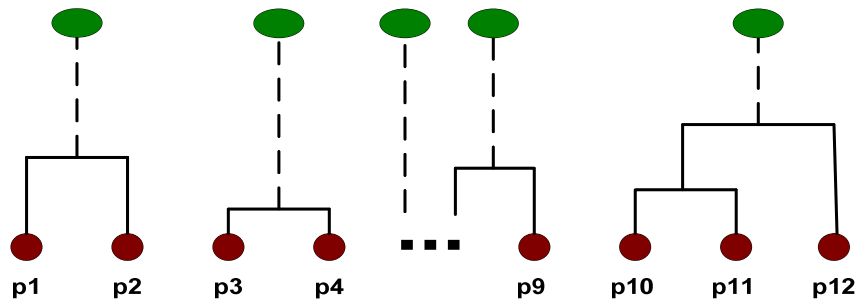


Intermediate State

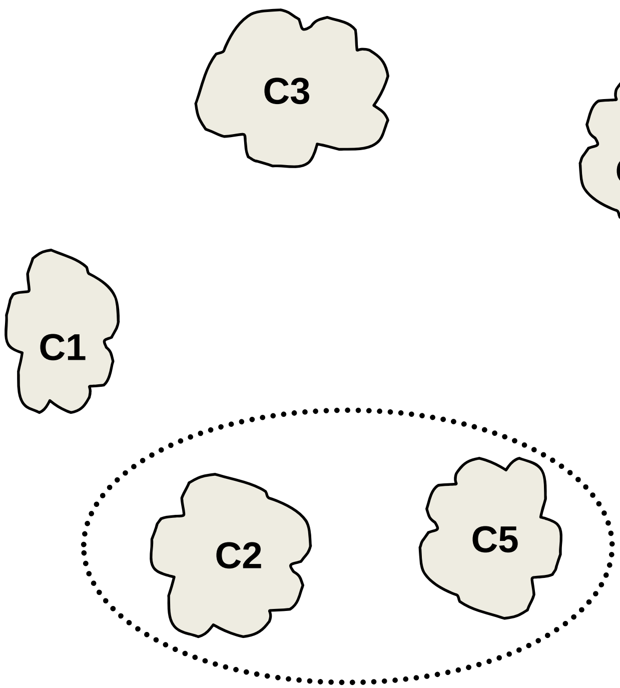


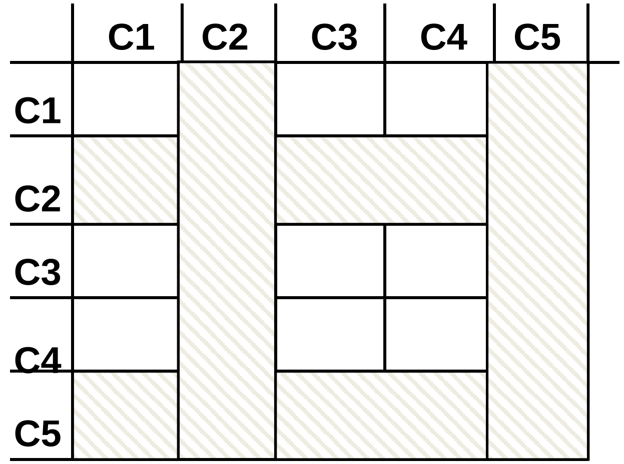
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix

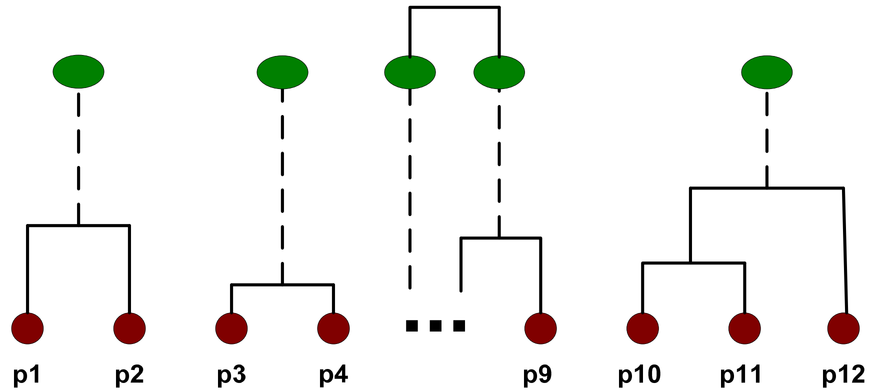


Intermediate State

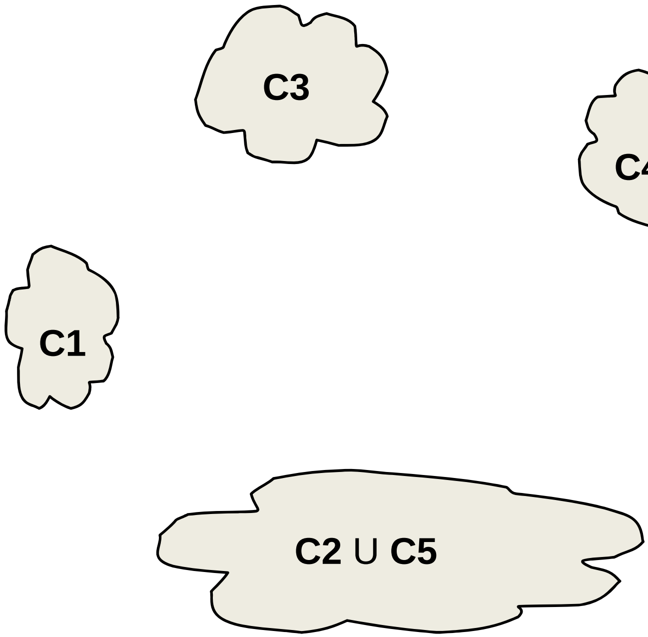




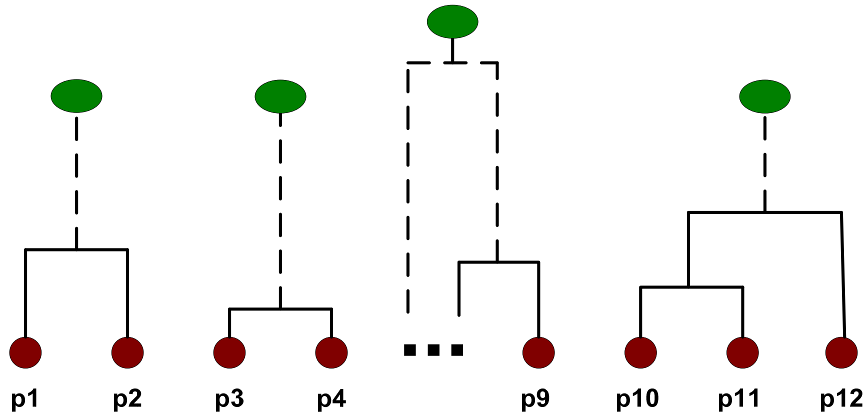
Distance/Proximity Matrix



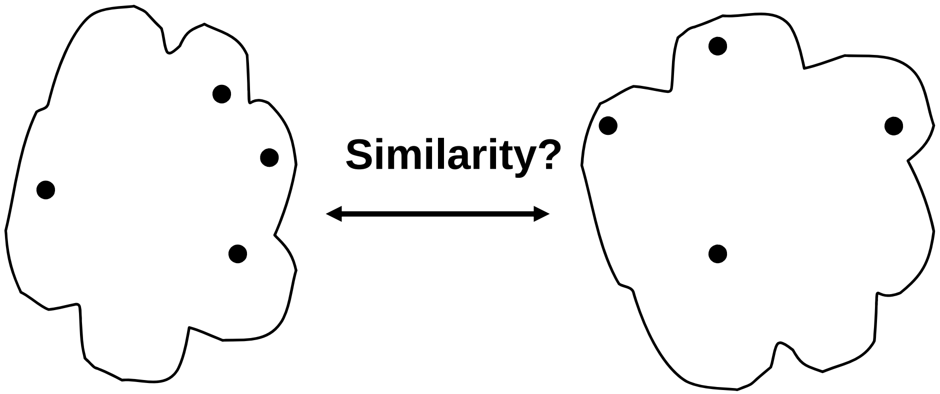
After Merging



	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

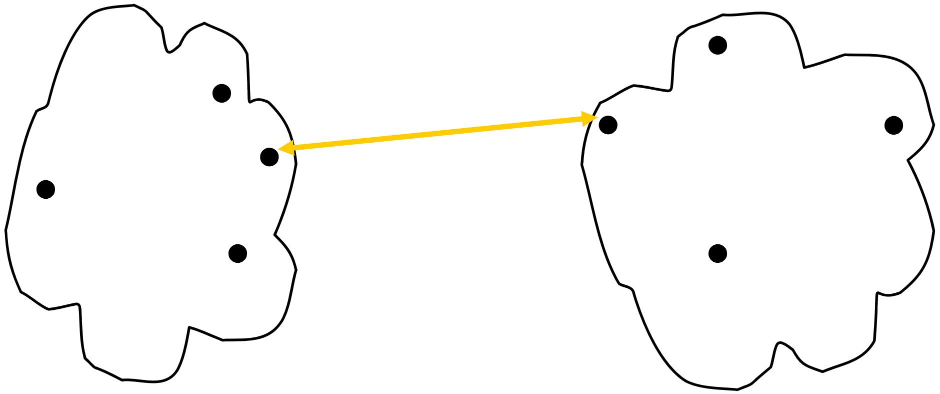


How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids

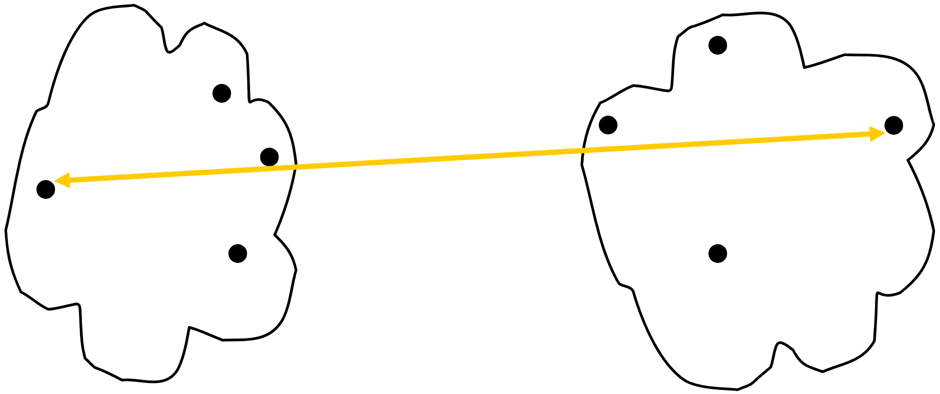
How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} \{ d(x, y) \}$$

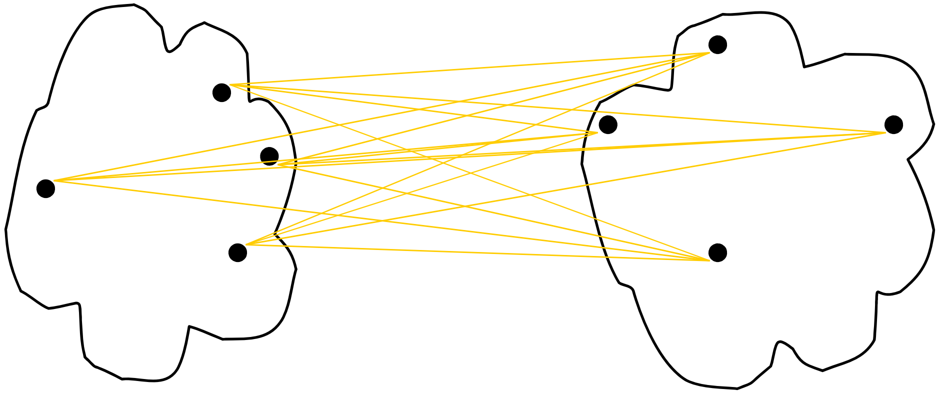
How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids

$$d(C_i, C_j) = \max_{x \in C_i, y \in C_j} \{ d(x, y) \}$$

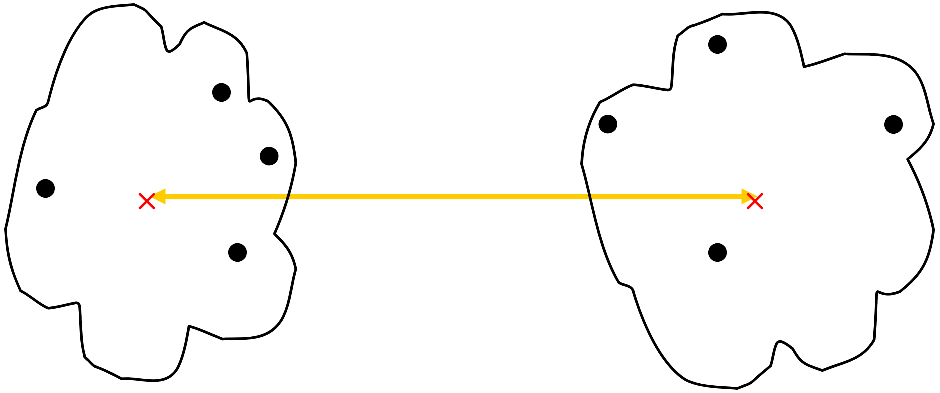
How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$$

How to Define Inter-Cluster Similarity

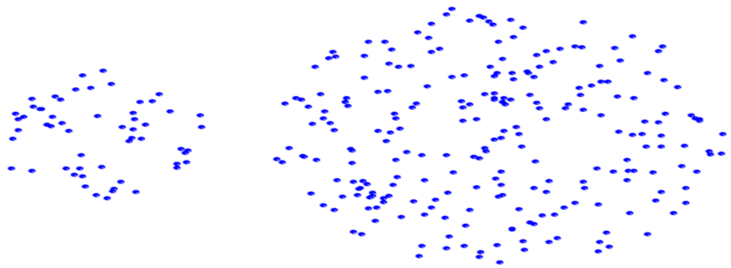


- MIN
- MAX
- Group Average
- Distance Between Centroids

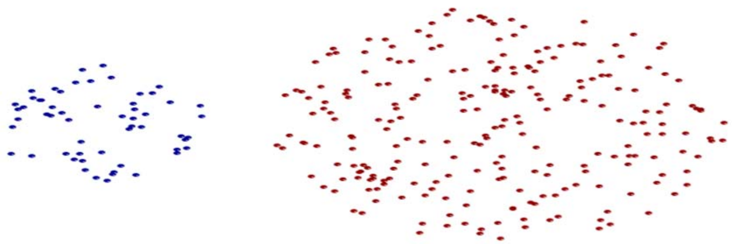
$$d(C_i, C_j) = d(c_i, c_j)$$

$$c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \qquad c_j = \frac{1}{|C_j|} \sum_{x \in C_j} x$$

Strength/Limitations of MIN

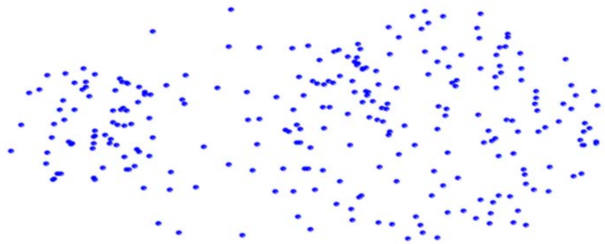


Original Points

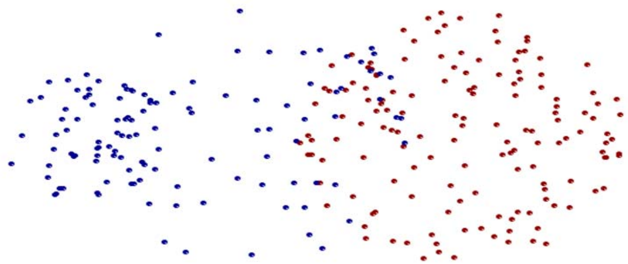


Two Clusters

• Can handle non-elliptical shapes



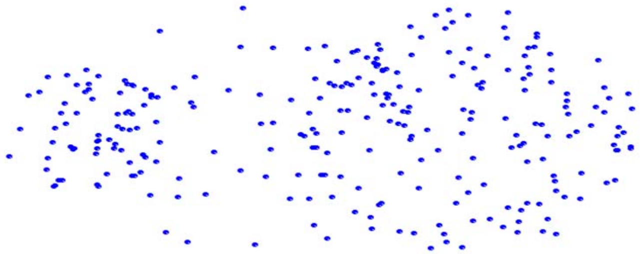
Original Points



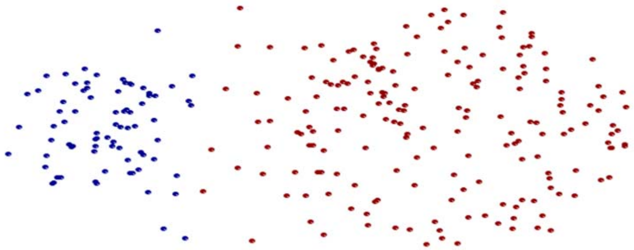
Two Clusters

• Sensitive to noise and outliers

Strength/Limitations of MAX

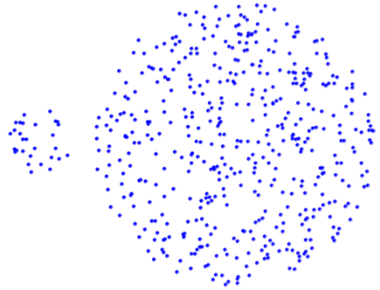


Original Points

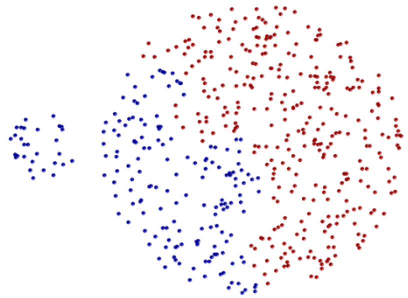


Two Clusters

- Less susceptible to noise and outliers



Original Points



Two Clusters

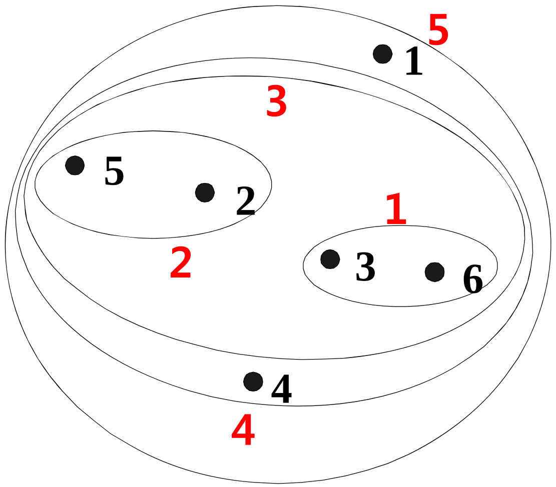
- Tends to break large clusters
- Biased towards globular clusters

(Cannot handle non-elliptical shapes)

Strength/Limitations of average

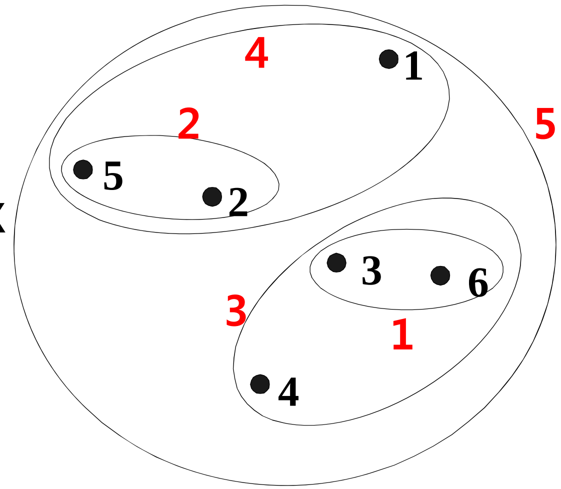
- Compromise between Single and Complete Link
- **Strengths**
  - Less susceptible to noise and outliers
- **Limitations**
  - Biased towards globular clusters

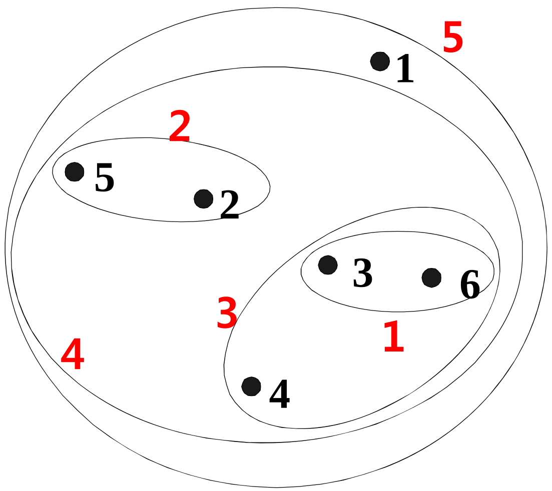
Hierarchical clustering: comparison



MIN

MAX





Group Average

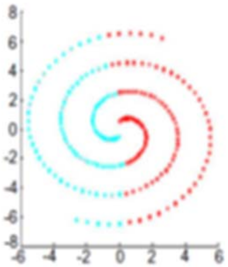
Hierarchical Clustering: Problems and Limitations

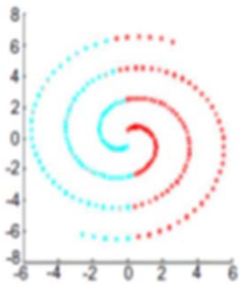
- **Advantages**
  - No objective function is directly minimized

- **Limitations**
  - **$O(N^2)$ space** since it uses the proximity matrix.
  - **$O(N^3)$ time** in many cases
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters
  - Difficulty handling different convex shapes
  - Breaking large clusters

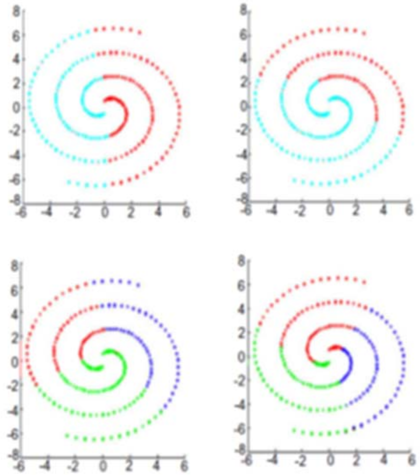
Clustering ensembles

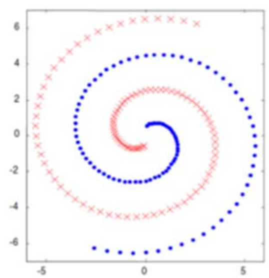
Single clustering





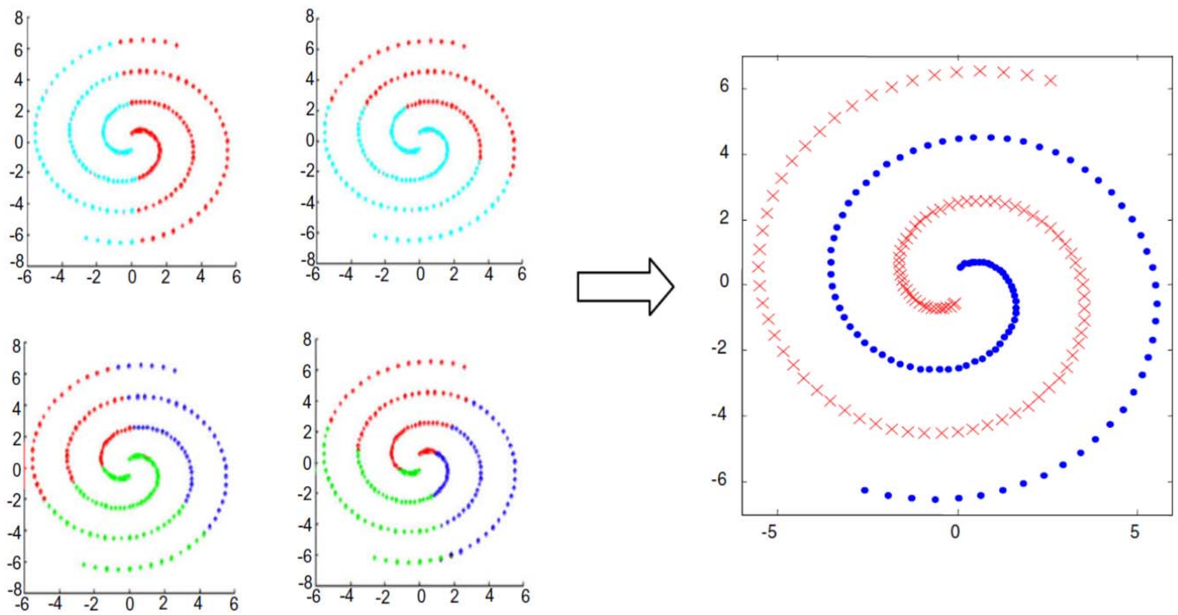
Ensemble clustering





Clustering ensembles

Fred and Jain(2002)



Ensembles

Different K
Different initialization

Combination

Co-occurrence matrix + Single Link method

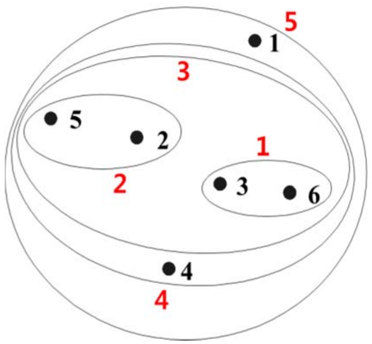
Clustering ensembles

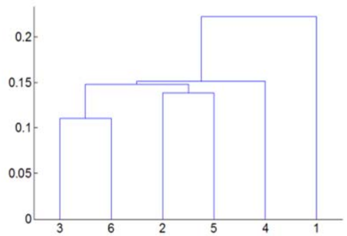
Similarity measure

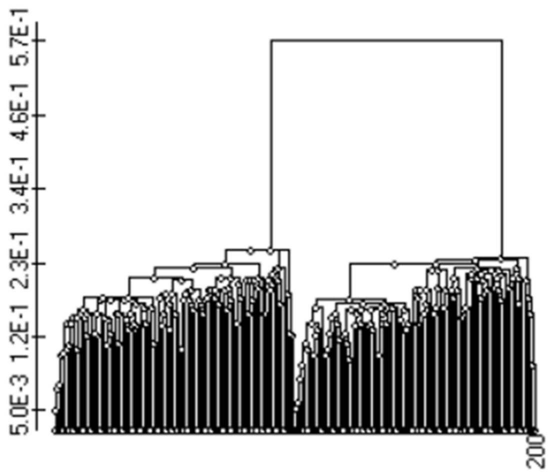
$$co\_assoc(i,j) = \frac{votes_{ij}}{N},$$

Where N is the number of clustering and $votes_{ij}$ is the number of times the pattern pair(i,j) is assigned to the same cluster among the N clustering.

Combining : Single Linkage

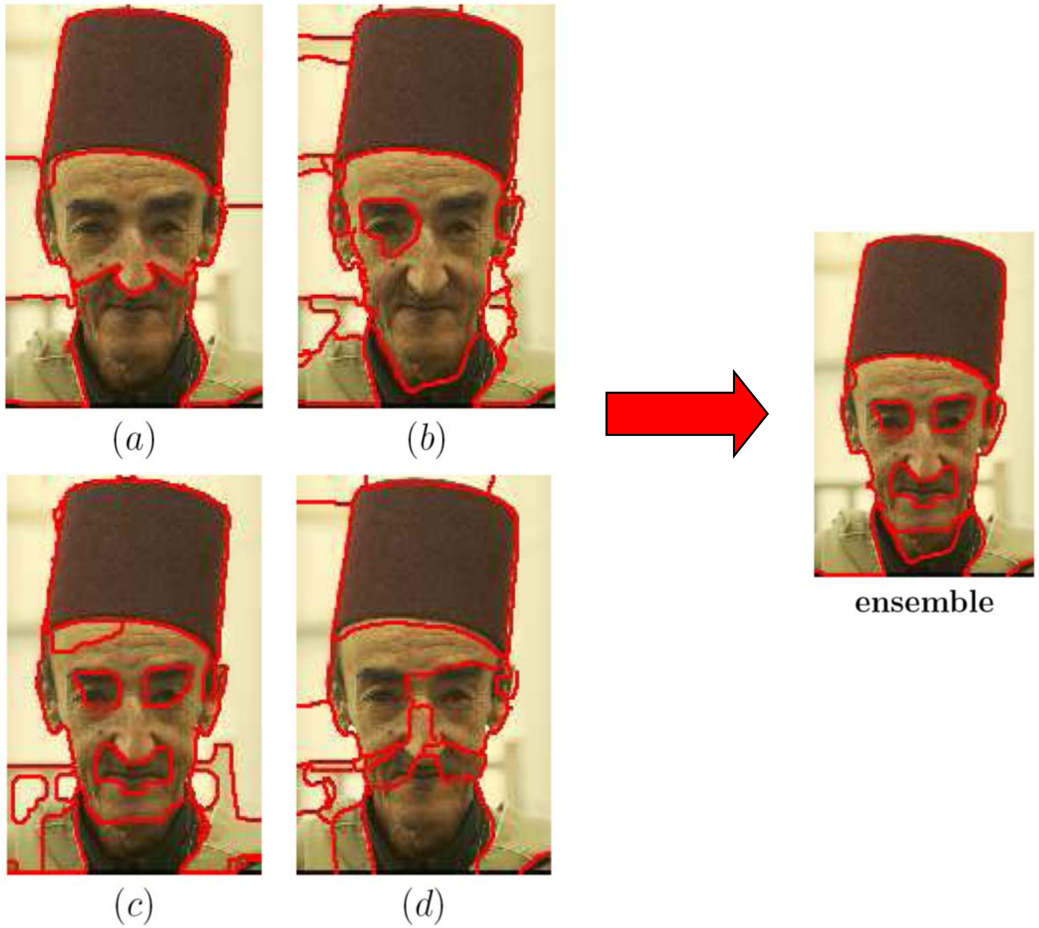






Clustering ensembles

Example



Large scale clustering

What is a large scale data?

Table 1
Example applications of large-scale data clustering.

Application	Description	# Objects	# Features
Document clustering	Group documents of similar topics (Andrews et al., 2007)	10^6	10^4
Gene clustering	Group genes with similar expression levels (Lukashin et al., 2003)	10^5	10^2
Content-based image retrieval	Quantize low-level image features (Philbin et al., 2007)	10^9	10^2
Clustering of earth science data	Derive climate indices (Steinbach et al., 2003)	10^5	10^2

Algorithm

- **efficient nearest neighbor(NN) search**
- Data summarization
- Distributed computing
- Incremental clustering
- Sampling based methods
- **Removing redundant calculations**

① Kdtree based kmeans
② Elkan kmeans
③ Hamerlay kmeans
④ Kmeans++

뒤에서 자세히 설명

Various algorithms are studied.

But…

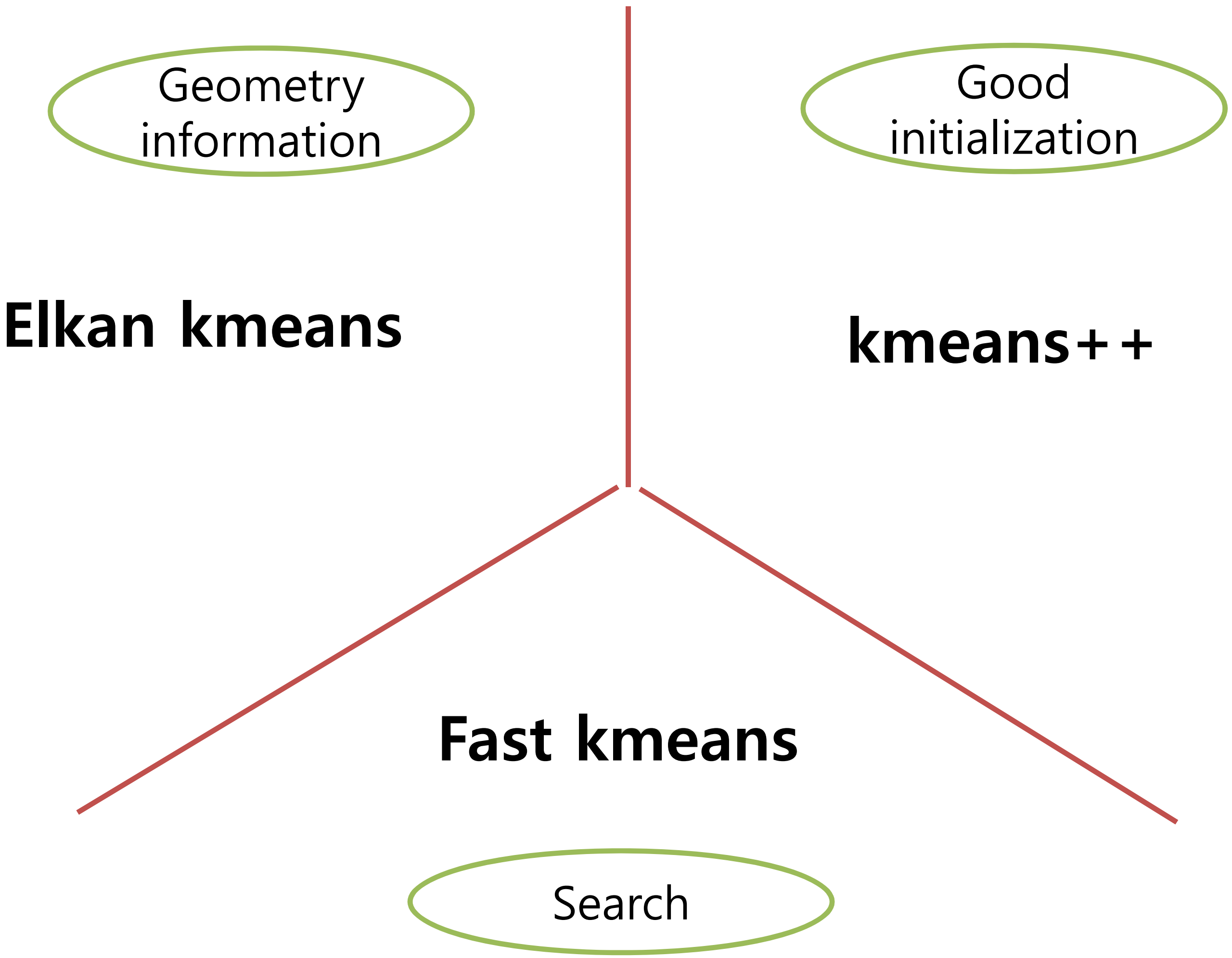
K-means clustering is the most popular algorithm.

Clustering ensembles

Large scale clustering

Multi-way clustering

Accelerated algorithms



Search

KD-TREE BASED FAST KMEANS

Inner-most loop in kmeans

$$J_1 = \sum_{i}^{N} \sum_{j}^{K} r_{ij} \boxed{\| x_i - c_j \|^2}$$

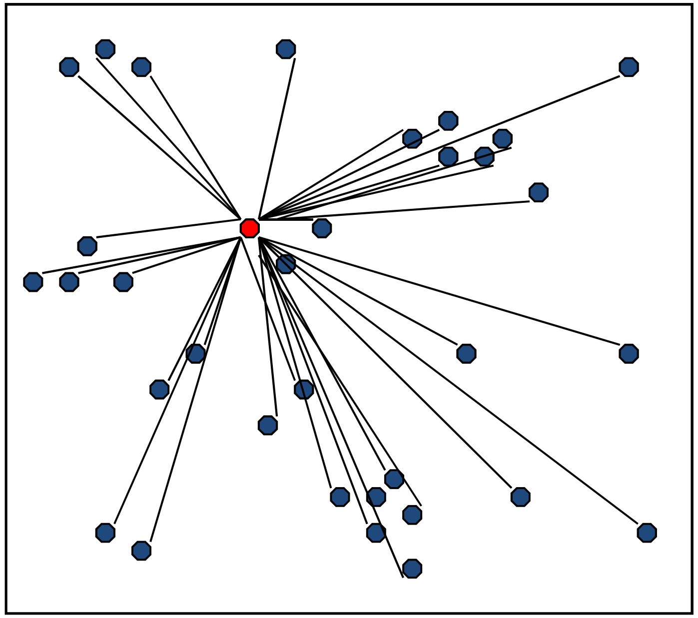
Repeat points-centers distance calculations

This is most time consuming part

How can we reduce these calculations?

Naïve Nearest Neighbor

For finding the closest cluster

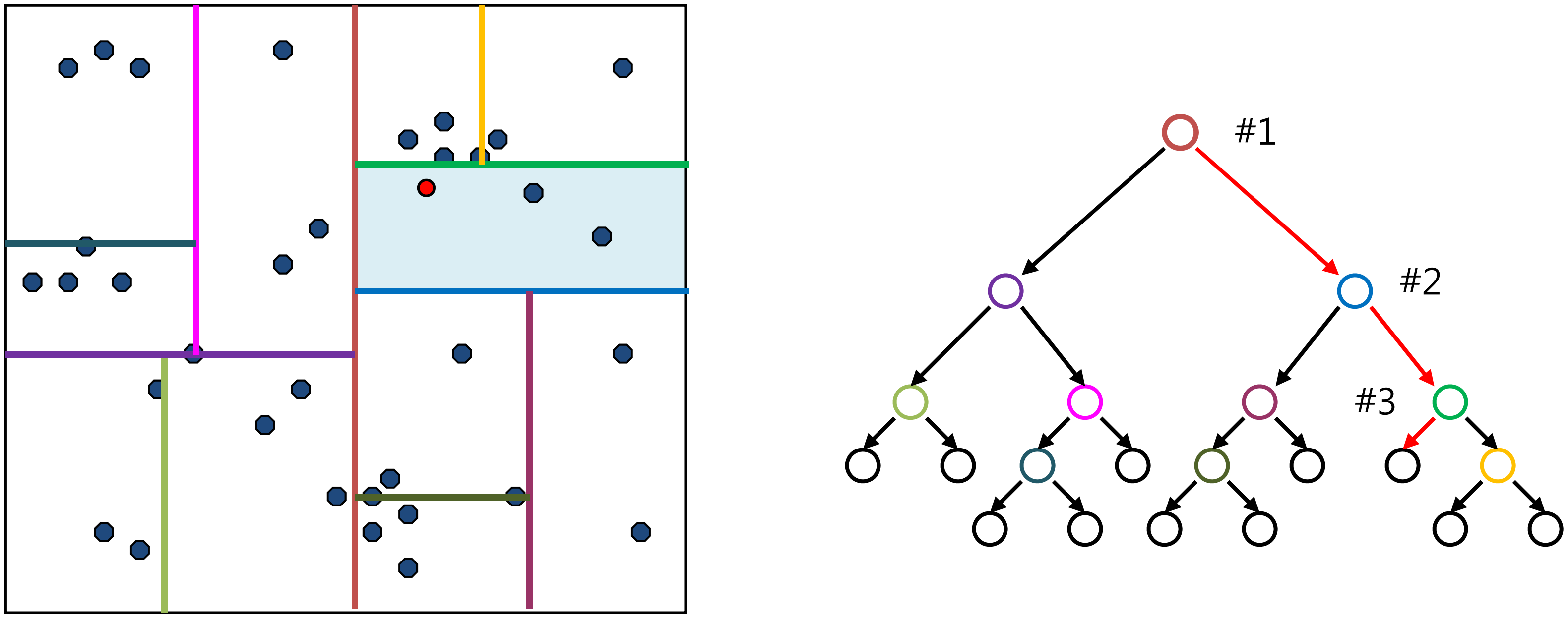


33 Distance Computations

$$J_1 = \sum_{i}^{N} \sum_{j}^{K} r_{ij} \| x_i - c_j \|^2$$

Speeding up Nearest Neighbor

Using KD-tree
- Examine nearby points first
- Ignore any points that are father than the nearest point



Go KD-TREES

Geometry information

ELKAN KMEANS

Elkan kmeans

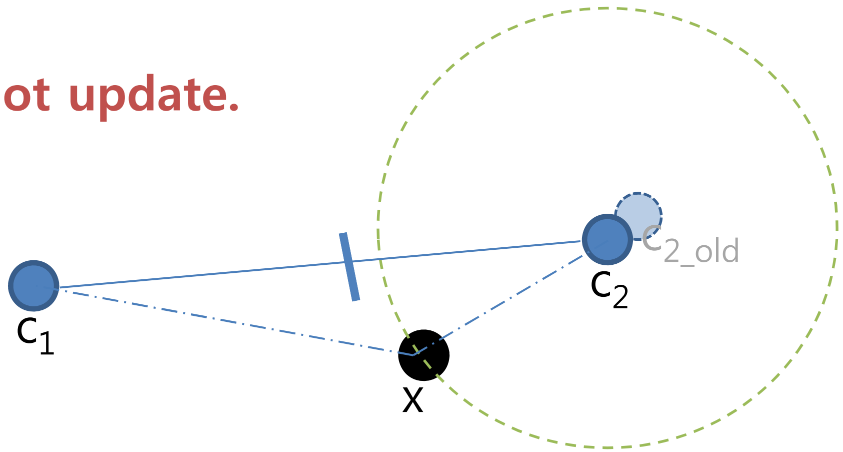
kmeans

Update clusters of all points

kmeans using Elkan distance bound

Update clusters of some points which are out of bound

$d(x,c_2)$ needs not update.

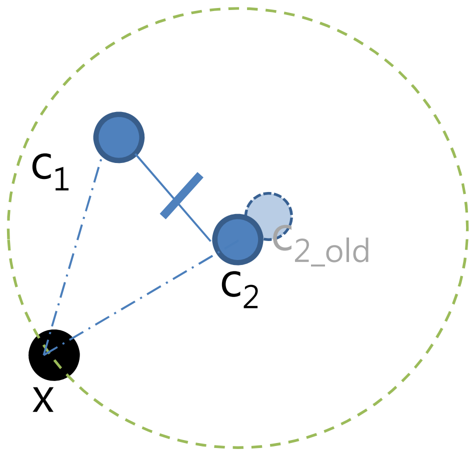


Assumption, $c_2 \fallingdotseq c_{2\_old}$

Lemma #1 If $d(c_2,c_1) >= 2d(x,c_{2\_old})$, Then $d(x,c_1) >= d(x,c_2)$

Elkan kmeans

$d(x,c_2)$ needs update.
And gets new lower bound



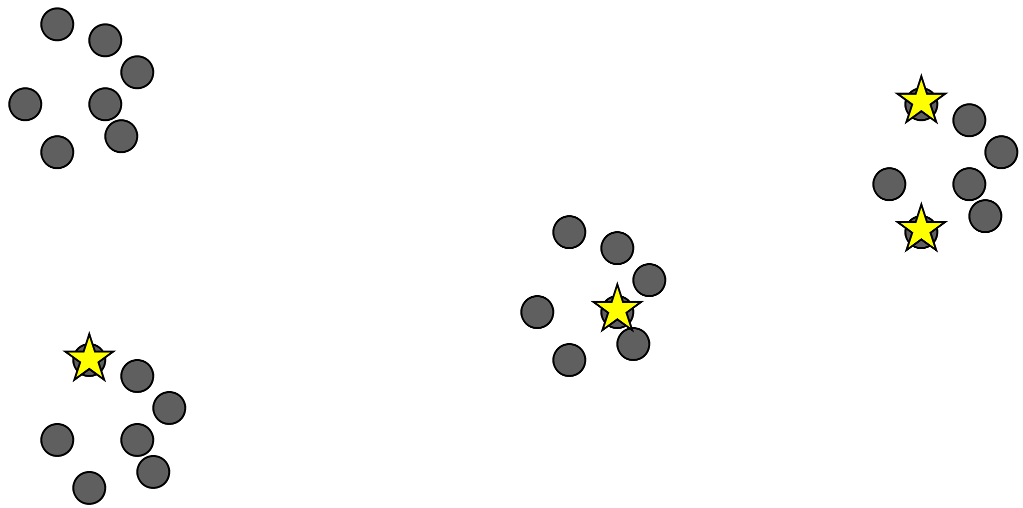
Assumption, $c_2 \fallingdotseq c_{2\_old}$

Lemma #2 $d(x,c_2) >= \max[\, 0,\ d(x,c_1)-d(c_1, c_{2\_old})]$
$= \max[0,\ d(x,c_{2\_old})-d(c_1, c_{2\_old})]$

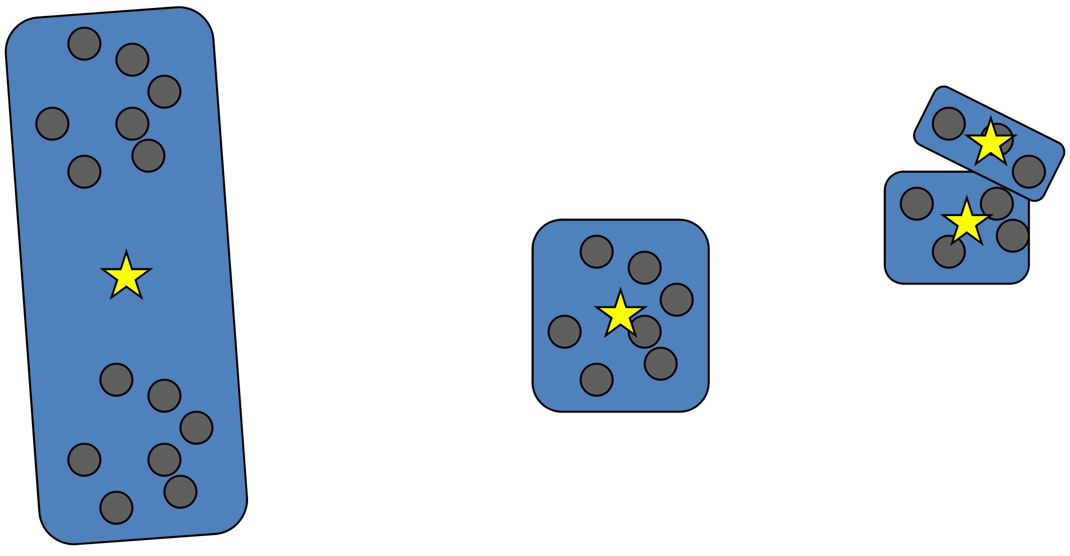
Good initialization

KMEANS++

Bad initialization



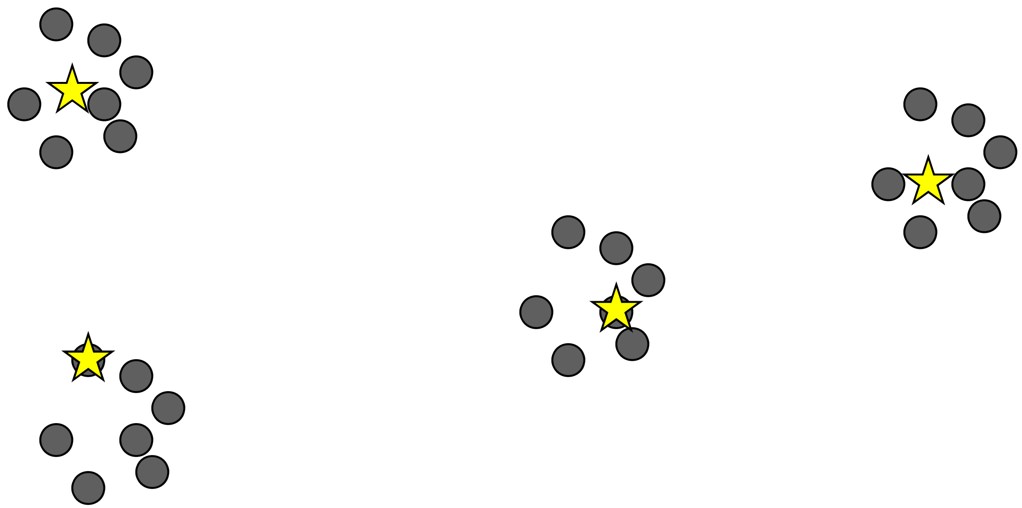
Trapped local minima



Kmeans++

- The k-means++ way:
  - Choose starting centers iteratively
  - Let $D(x)$ be distance from x to nearest existing center
  - Take x as new center with prob. proportional to $D(x)^2$

- Run standard Lloyd's method with these centers



Evaluating *k*-means++

- **Speed**:
  - Initialization similar to 1 iteration of Lloyd's method
  - In practice:
    - Uses fewer iterations than Lloyd's method
    - Runs *faster* than Lloyd's method
- Simplicity

Evaluating *k*-means++

- Speed
- **Simplicity:**
  - Only marginally harder to implement than Lloyd’s method
  - Easy to understand intuitively

Take home message

- Structured data clustering
  - Ensemble clustering can handle non-convex data.

- Unstructured data clustering
  - **KD-tree based method** is the fastest algorithm in low dimensional data. (up to 10 dimensions)
  - **Elkan kmeans** is the state of the art in high dimensional large data.
  - **Kmeans++** is almost used in the initial procedure of kmeans variations.