Clustering Analysis

Man-Kwan Shan
Dept. of Computer Science
National Cheng-Chi Univ.

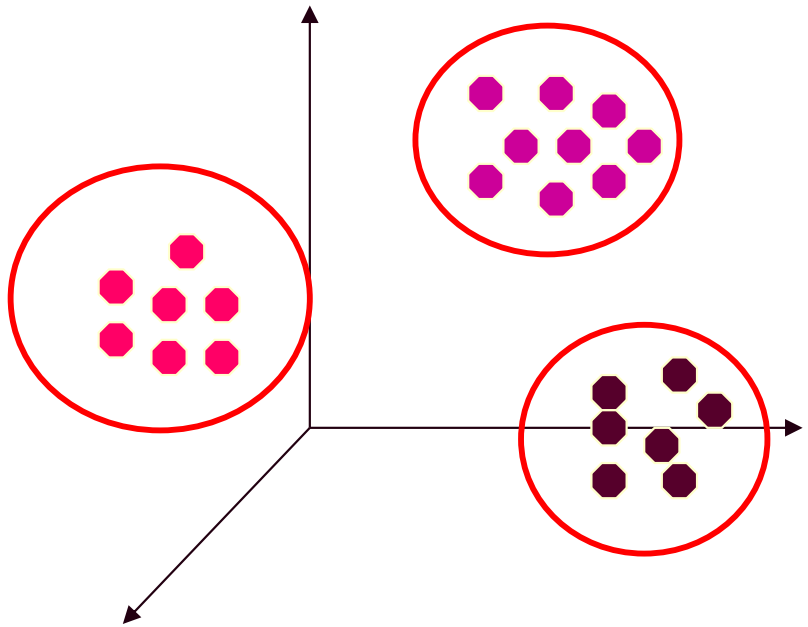
Clustering Analysis

Man-Kwan Shan
Dept. of Computer Science
National Cheng-Chi Univ.

Overview

Clustering

- Clustering: process of grouping a set of objects into clusters of similar objects
- Cluster: a collection of data objects that
  - are similar to one another within the same cluster
  - are dissimilar to the objects in other clusters



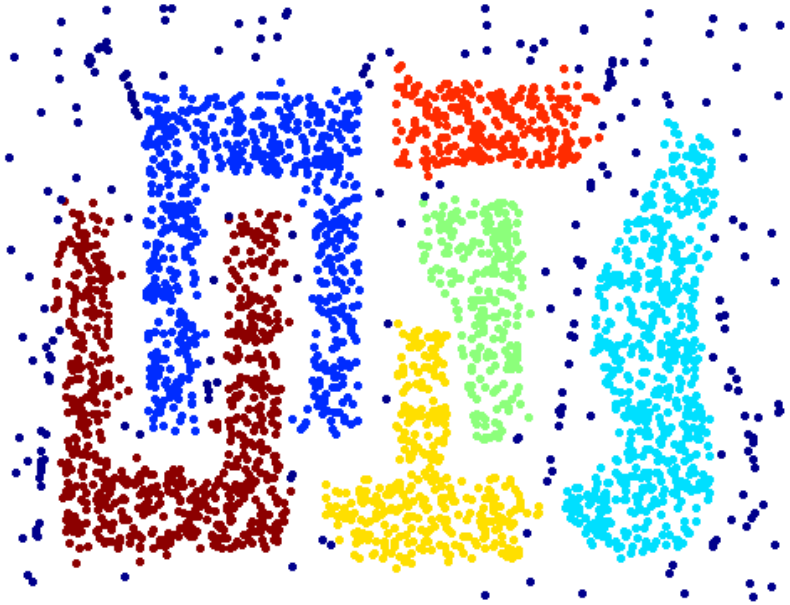
Good Clustering

- Good clustering (produce high quality clusters)
  - intra-cluster similarity is high
  - inter-cluster class similarity is low
- Quality factors
  - similarity measure and its implementation
  - definition and representation of cluster chosen
  - clustering algorithm

Requirements of Clustering

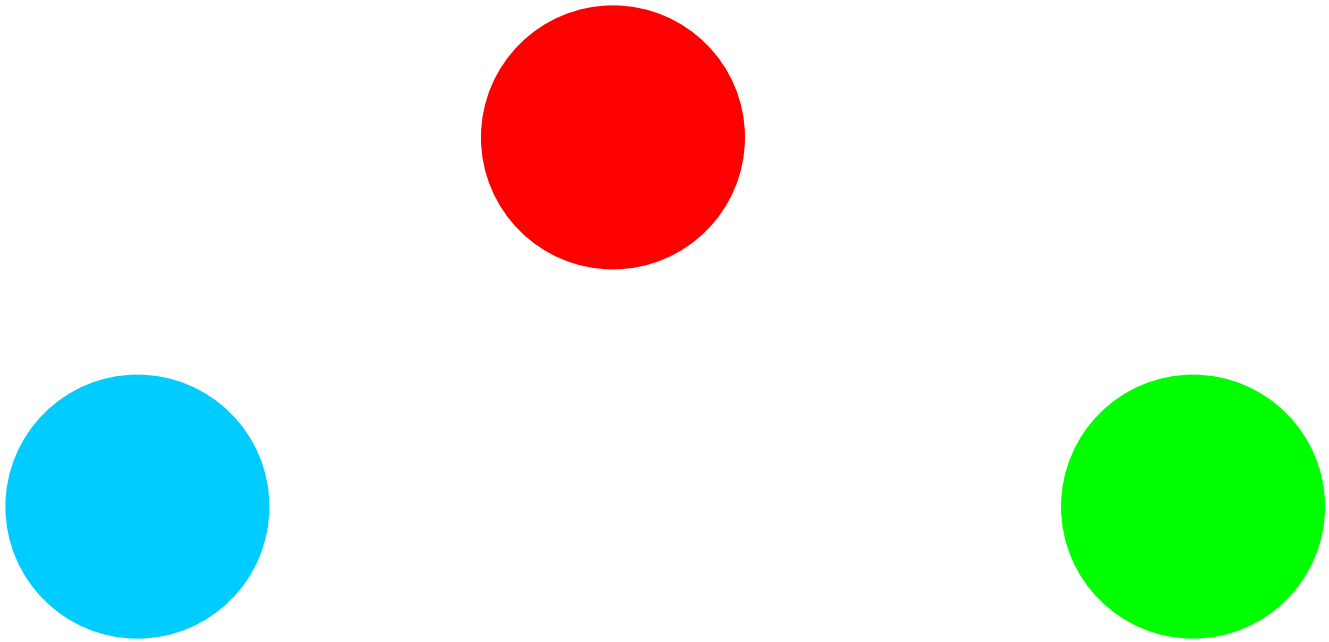
- Dealing with different types of attributes (not only numerical data)
- Discovery of clusters with arbitrary shape (not only sphere)
- Minimal requirements for domain knowledge to input design parameters
- Ability to deal with noisy data
- Insensitivity to order of input records
- High dimensionality
- Scalability
- Constraint-based clustering
- Interpretability and usability

Shape of Clusters



Types of Clusters: Well-Separated

- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.

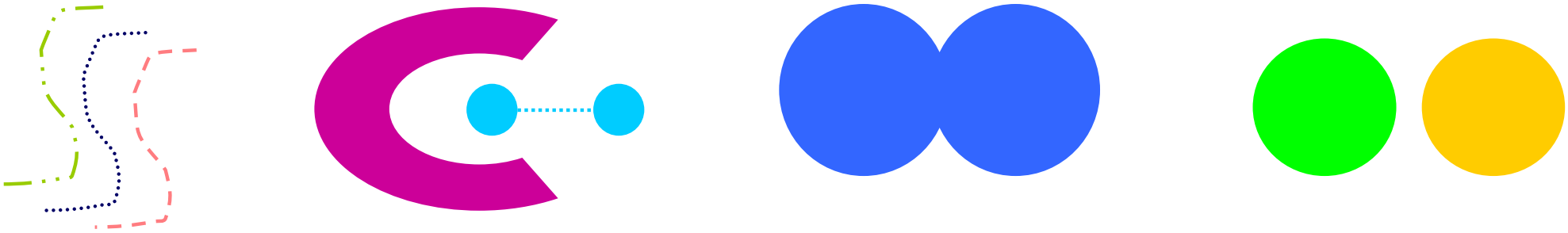


Types of Clusters: Center-Based

- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the center of a cluster, than to the center of any other cluster
  - The center of a cluster is often
    - a centroid, the average of all the points in the cluster, or
    - a medoid, the most representative point of a cluster

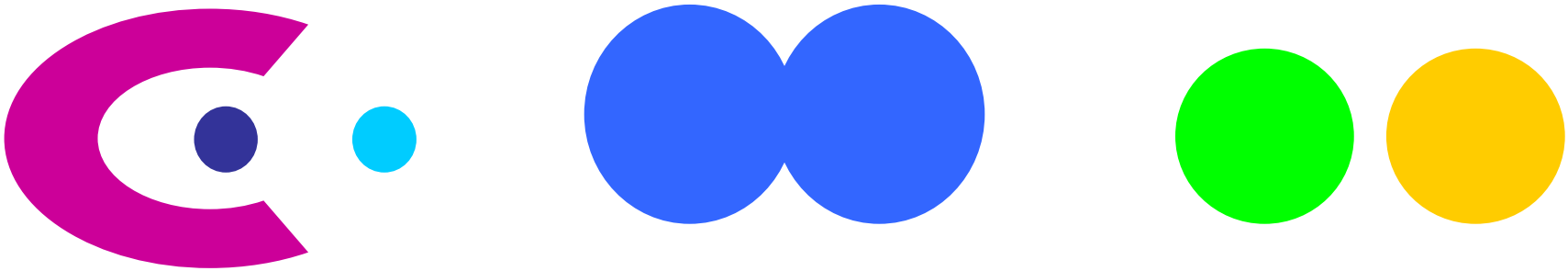
Types of Clusters: Contiguity-Based

- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



Types of Clusters: Density-Based

- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.



Approaches of Clustering Algorithms

Approaches of Clustering Algorithms

- Partition-based
  - Construct various partitions and then evaluate them by some criterion.
- Hierarchical
  - Create a hierarchical decomposition of objects using some criterion.
- Density-based
  - based on connectivity and density functions
- Model-based
  - A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other.

Partition-based Clustering

Partitioning-based Algorithms

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters.
  - e.g. partition of 4 objects {a, b, c, d} into 2 clusters

Cluster 1	Cluster 2
{a}	{b, c, d}
{b}	{a, c, d}
{c}	{a, b, d}
{d}	{a, b, c}
{a, b}	{c, d}
{a, c}	{b, d}
{a, d}	{b, c}

7 possible partitions

Partitioning-based Algorithms

- Given a *k*, find a partition of *k* clusters that optimizes the chosen partitioning criterion.
  - Global optimal: exhaustively enumerate all possible partitions.
  - Heuristic methods: sub-optimal
    - k-means: each cluster is represented by the center of the cluster
    - k-medoids: each cluster is represented by one of the objects in the cluster.

K-Means Clustering Method

- One of the oldest & most widely used clustering algorithms
- Given *k*, the *k-means* algorithm:

Step 1: Initial Partition

arbitrary partition objects into *k* nonempty subsets

Step 2: Update Centroids

compute mean as the centroids of the clusters of the current partition

Step 3: Relocation

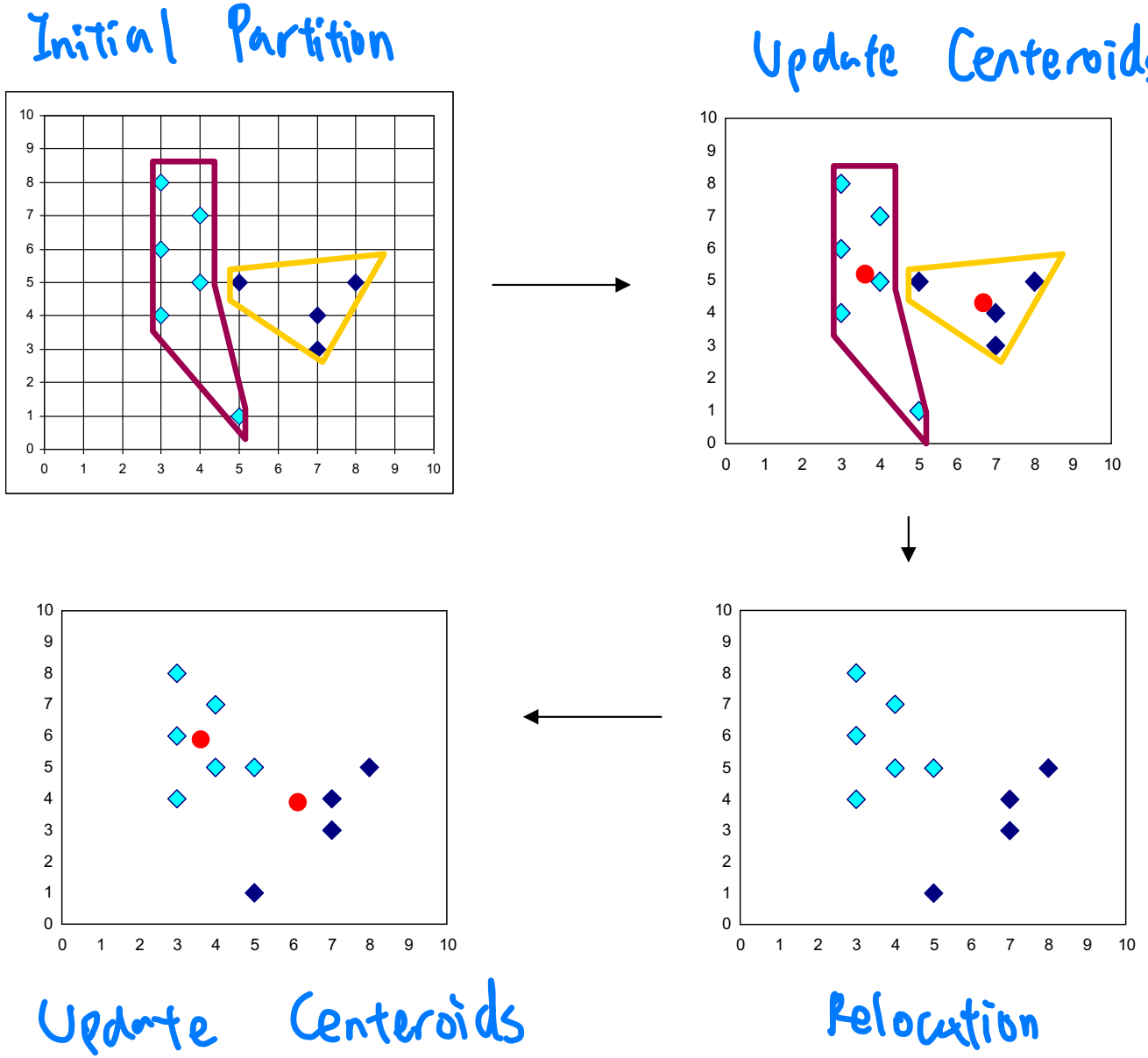
assign each object to the nearest cluster

Step 4: Go back to Step 2, stop when no more new relocation

convergence

The *K*-Means Clustering Method (cont.)

$K = 2$



K-Means Clustering Method (cont.)

- Given *k*, the *k-means* algorithm:

  Step 1: Initial centroids

  select *k* initial centroids

  Step 2: Relocation

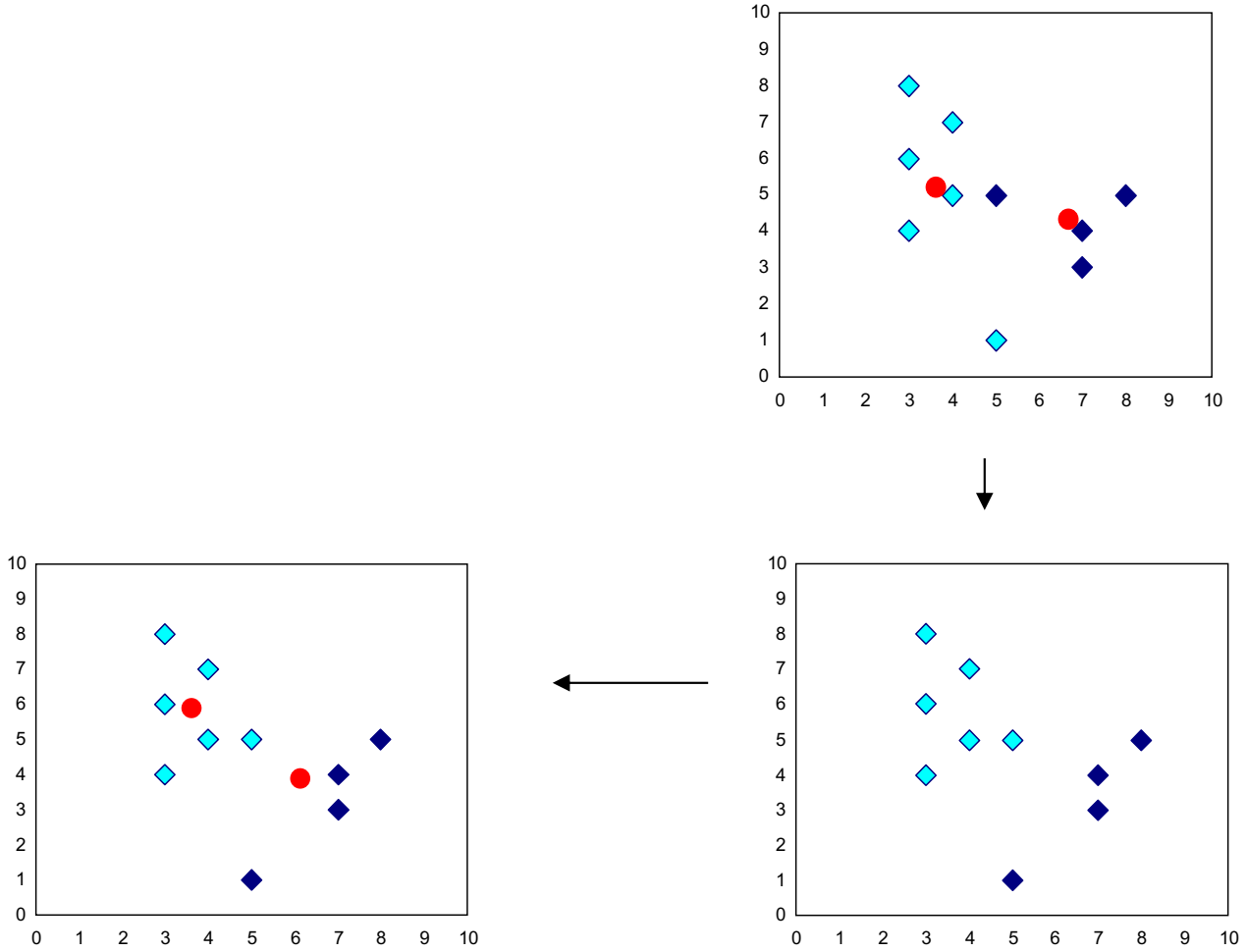
  Note assign each object to the nearest cluster

  Step 3: Update Centroids

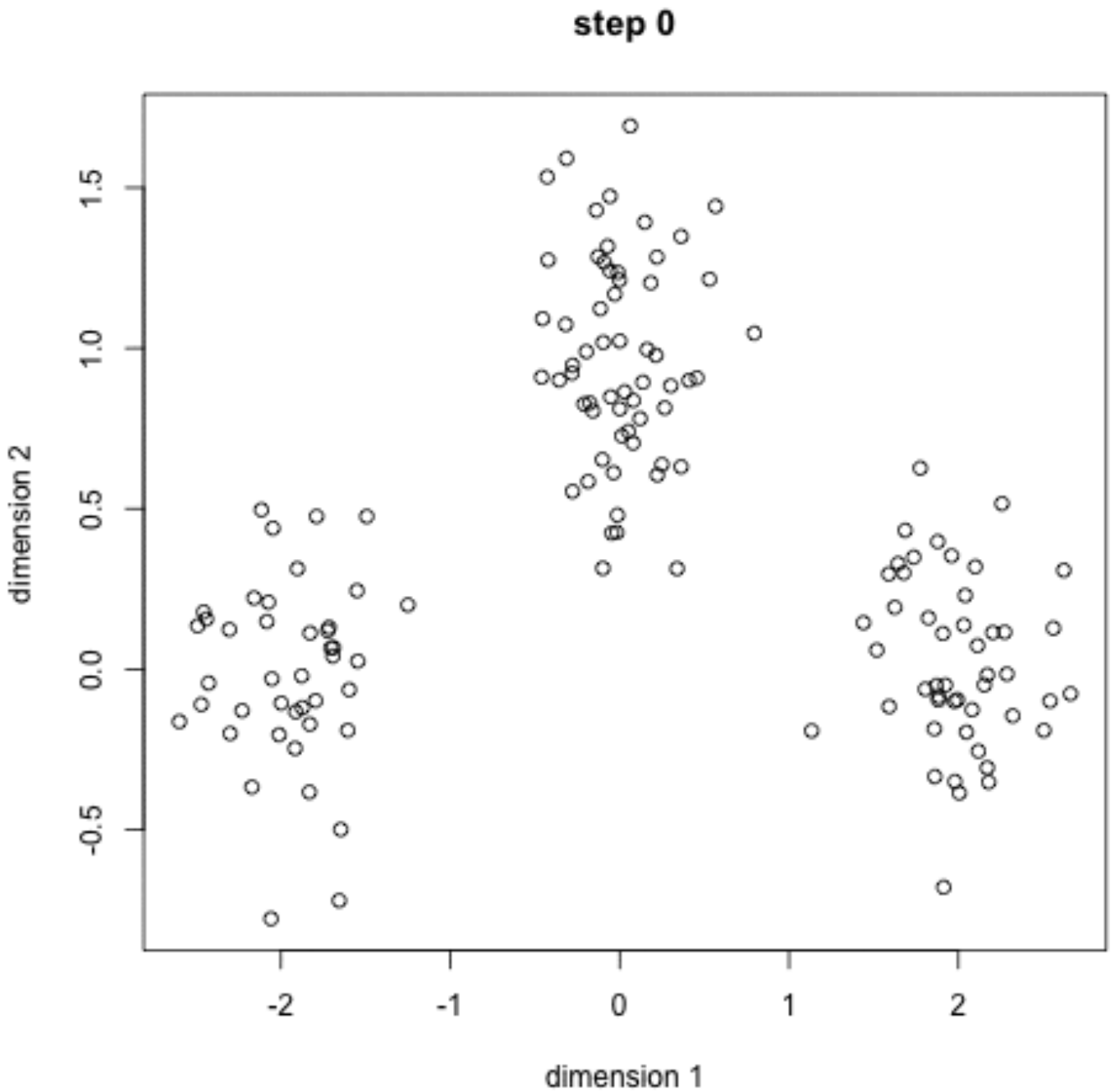
  compute mean as the centroids of the clusters of the current partition

  Step 4: Go back to Step 2, stop when no more new relocation

The *K*-Means Clustering Method (cont.)



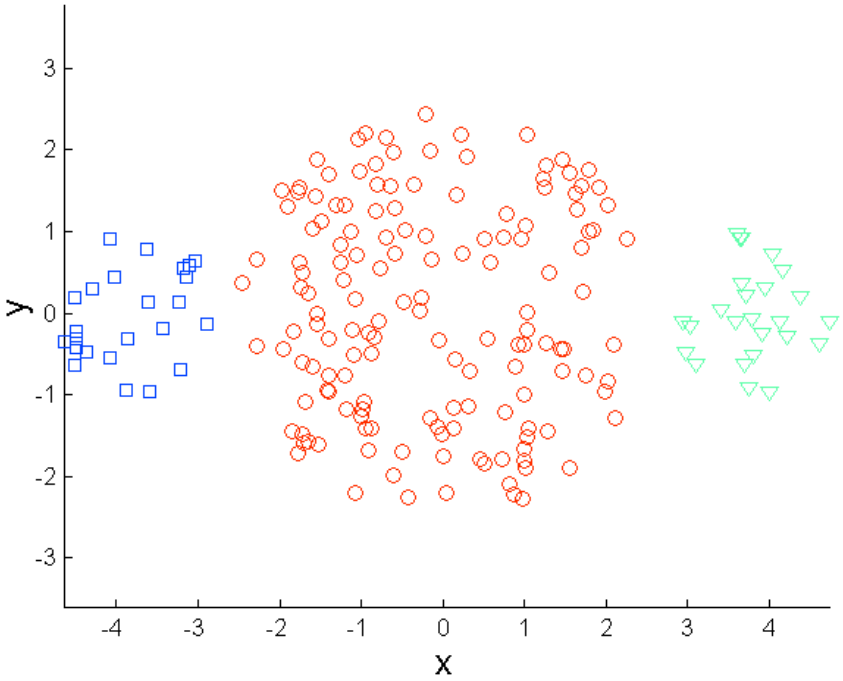
K-means



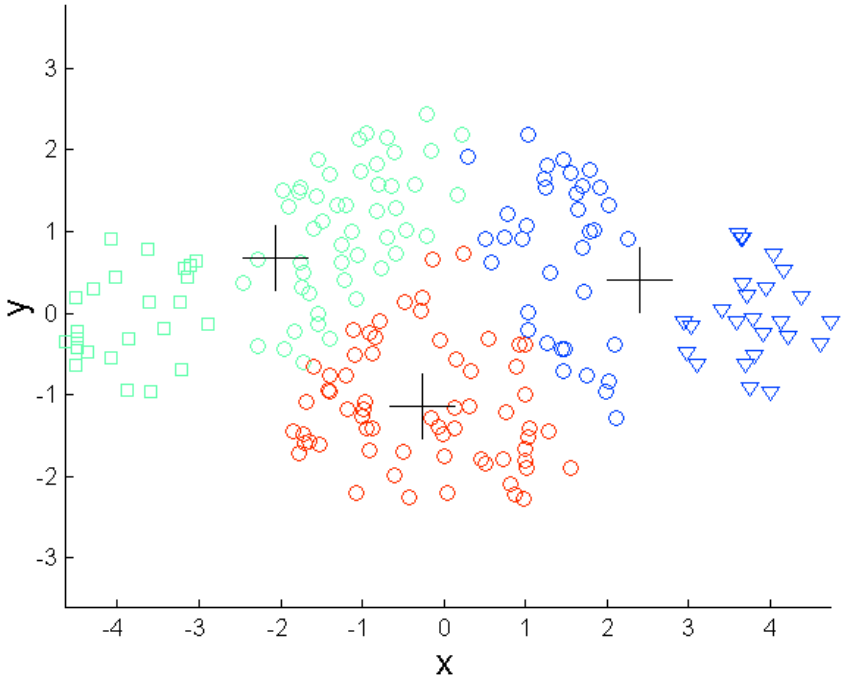
Comments on the *K-Means* Method

- Strength
  - Relatively efficient: $O(tkn)$, where n is # of objects, k is # of clusters, and t is # of iterations. Normally, k, $t << n$.
  - Often terminates at a *local optimum*.
- Weakness
  - Applicable only when *mean* is defined (categorical data?)
  - Need to specify *k,* the *number* of clusters, in advance.
  - Unable to handle noisy data and *outliers.*
  - Cannot handle clusters of different sizes & densities
  - Not suitable to discover clusters with *non-convex shapes.*
  - Empty cluster

Limitations of K-means: Differing Sizes

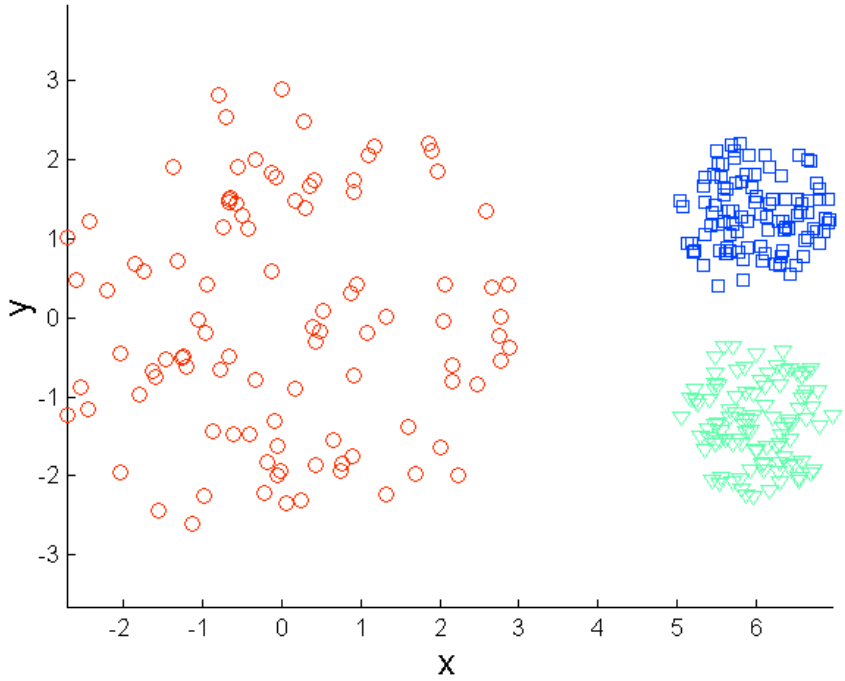


Original Points

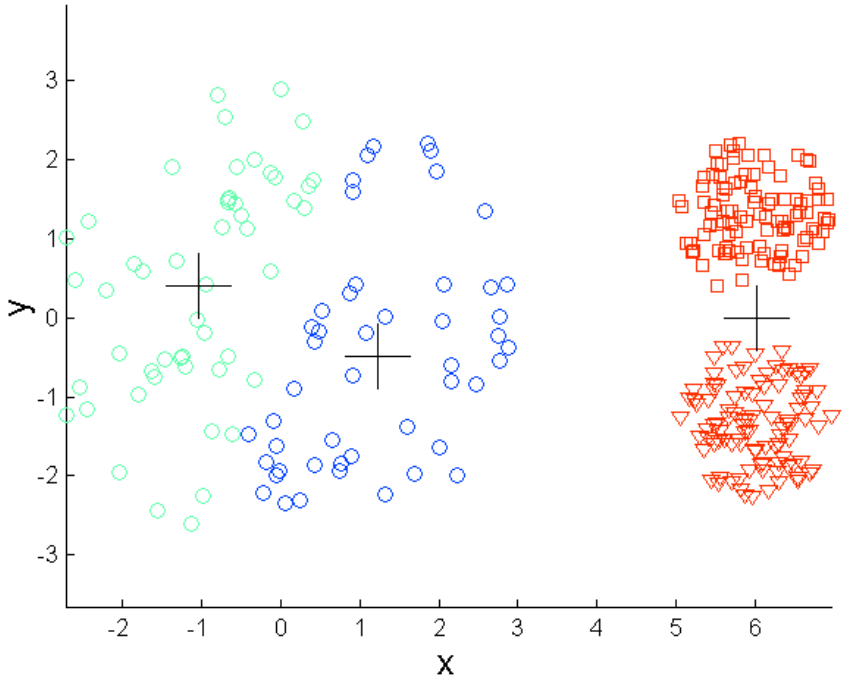


K-means (3 Clusters)

Limitations of K-means: Differing Density

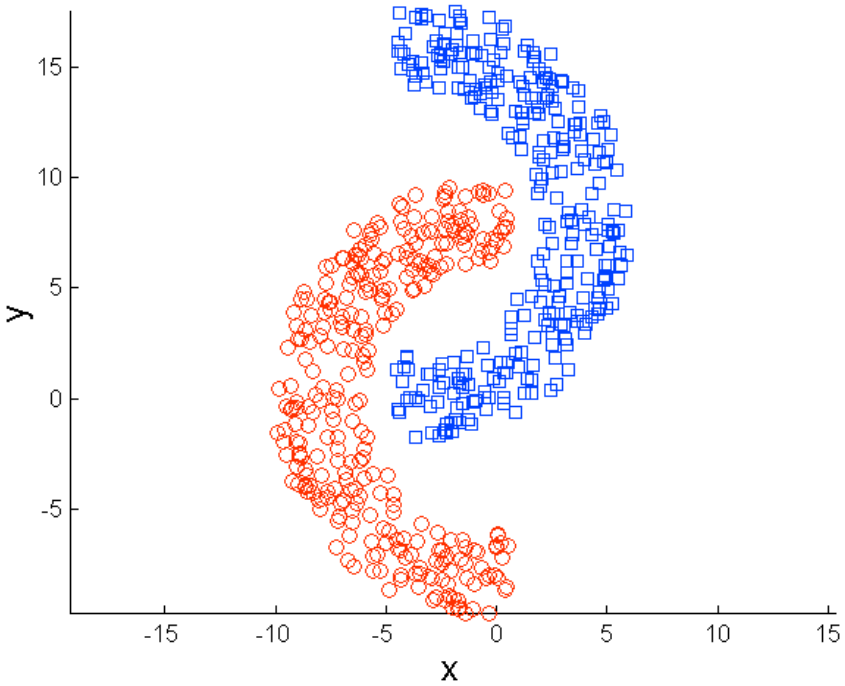


Original Points

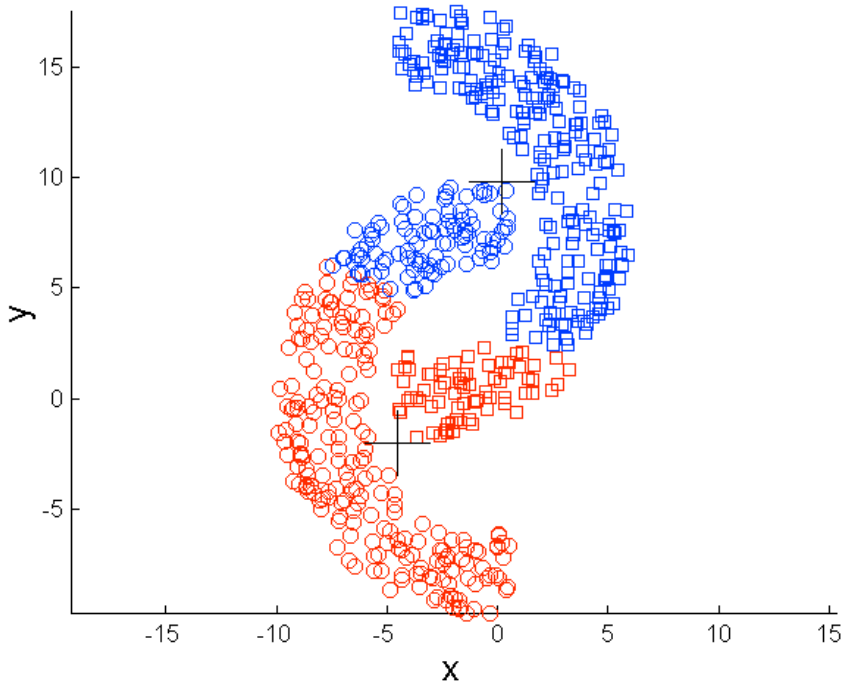


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points



K-means (2 Clusters)

Variations of the K-means Method

- Variants of the *k*-means
  - Selection of the initial *k* means.
  - Dissimilarity calculations. (L_1, L_2, Cosine, …)
  - Strategies to calculate cluster means.
  - Bisecting K-means
- Handling categorical data: *k-modes*
  - Replacing means of clusters with modes

* k-means is a special variant of the EM-algorithm with the assumption that clusters are spherical.

The *K-Medoids* Clustering Algorithms

- Medoid: representative objects in clusters
  - PAM (Partitioning Around Medoids, 1987)
  - CLARA (Clustering LARge Application, 1990)
  - CLARANS (a CLustering Algorithm based on RANdomized Search, 1994)

PAM (Partitioning Around Medoids)

- PAM built in S+.
- Use object to represent the cluster.

Select ***k*** medois arbitrarily

Repeat
  assign each remaining object to the cluster
    with nearest medoid
  calculate the objective function for clustering quality
  for each cluster C
    swap the medoid if swap reduces
      the objective function
Until there is no swap

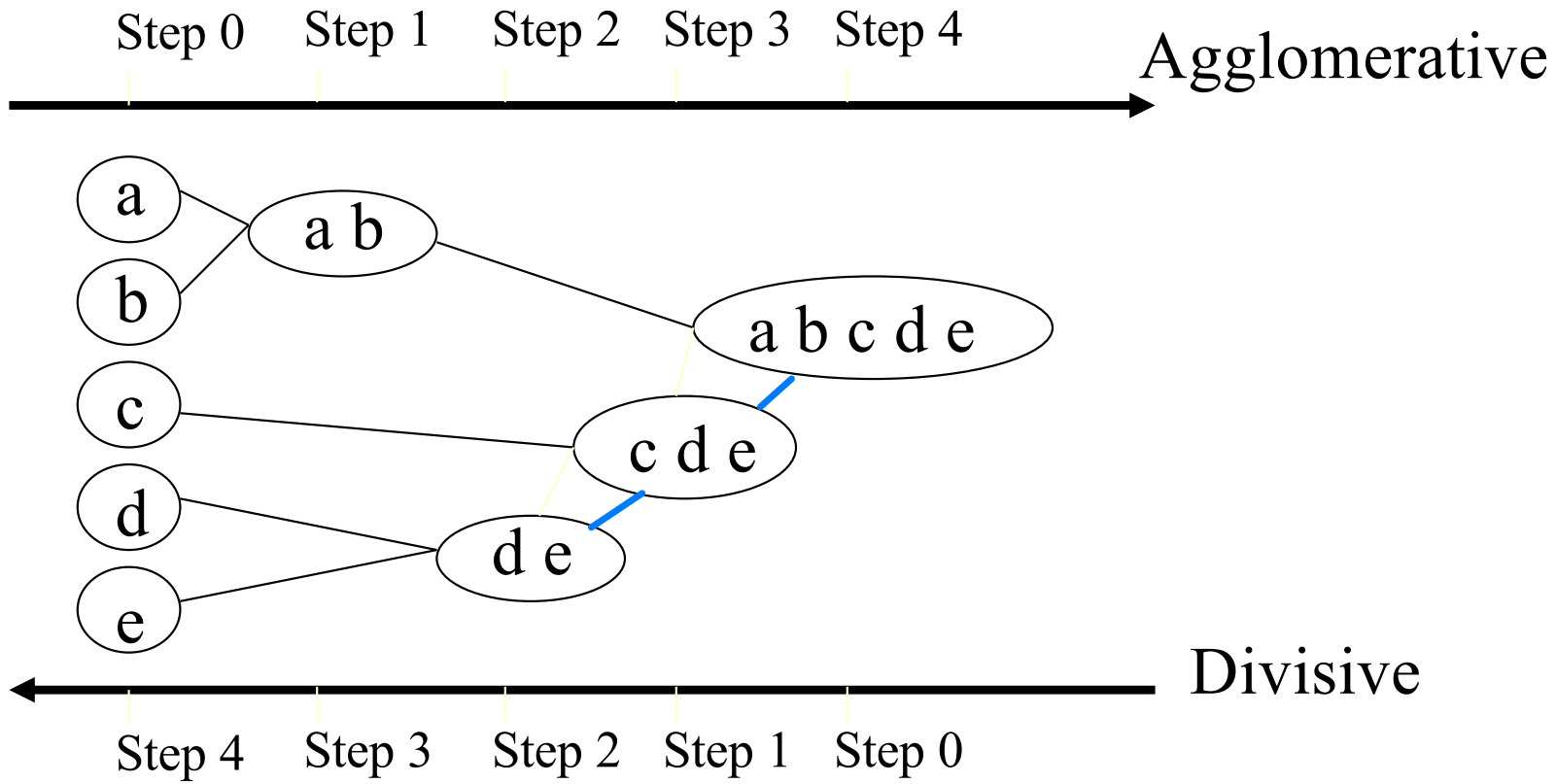
Hierarchical Clustering

Hierarchical Clustering

- A 2nd important category of clustering algorithms
- Grouping data into a tree of clusters
- Helpful for taxonomy construction
- Number of clusters is not required, but needs a termination condition.
- Approaches
  - bottom up: agglomerative
  - top down: divisive

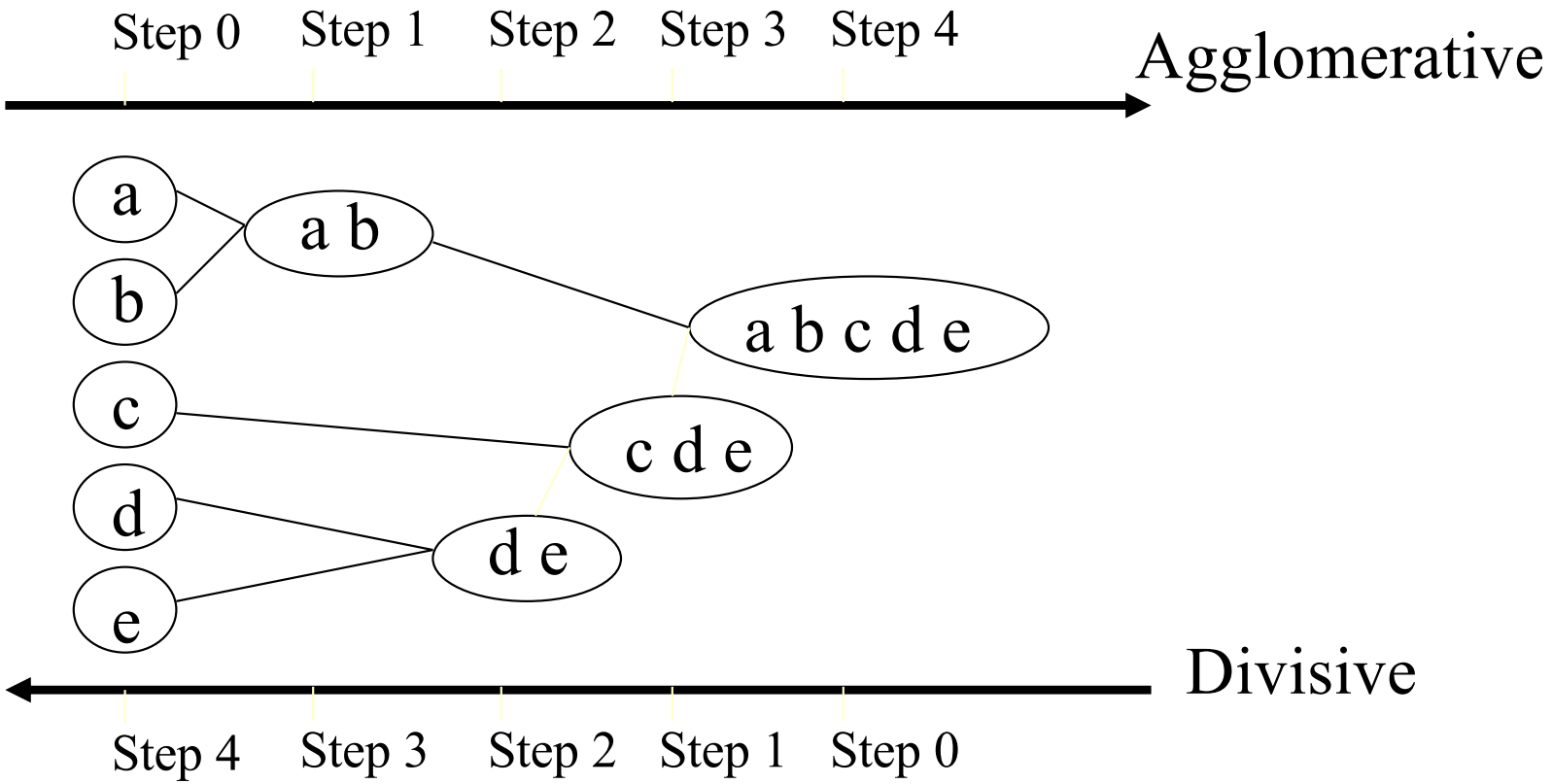
Hierarchical Clustering (cont.)

- Approaches
  - bottom up: agglomerative
  - top down: divisive



Agglomerative Hierarchical Clustering

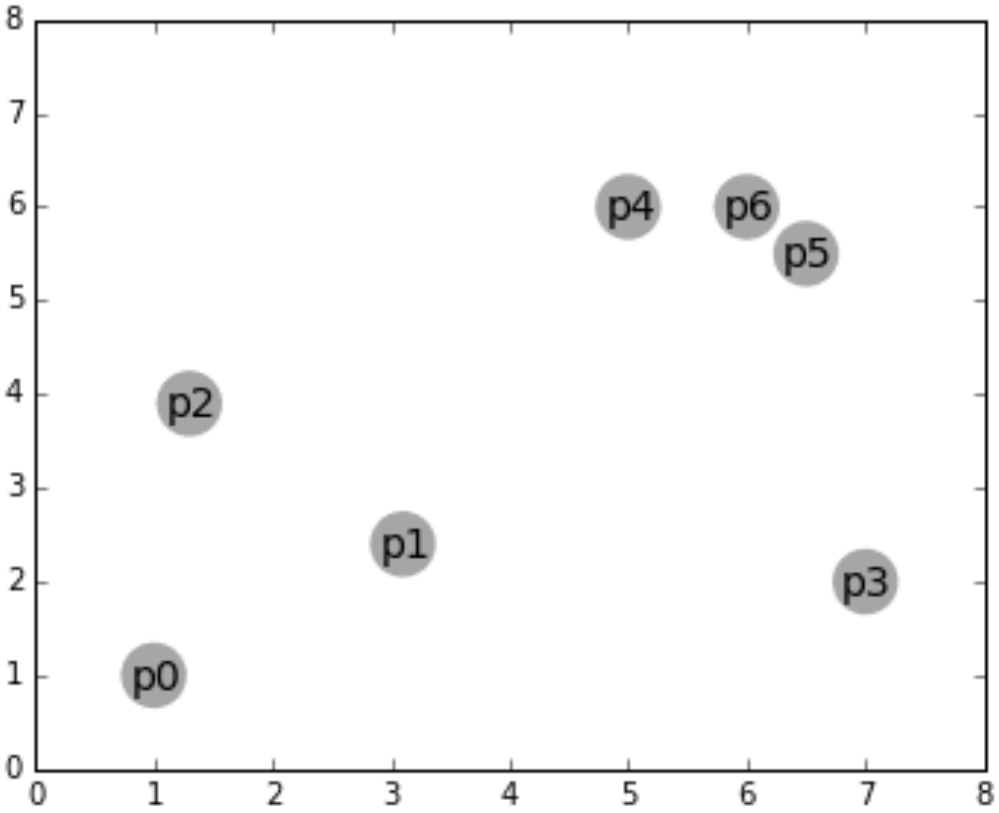
- More common than divisive approach.
- Start with the objects as individual clusters &
  at each step merge the closest pair of clusters.
- Definition of cluster proximity is required.

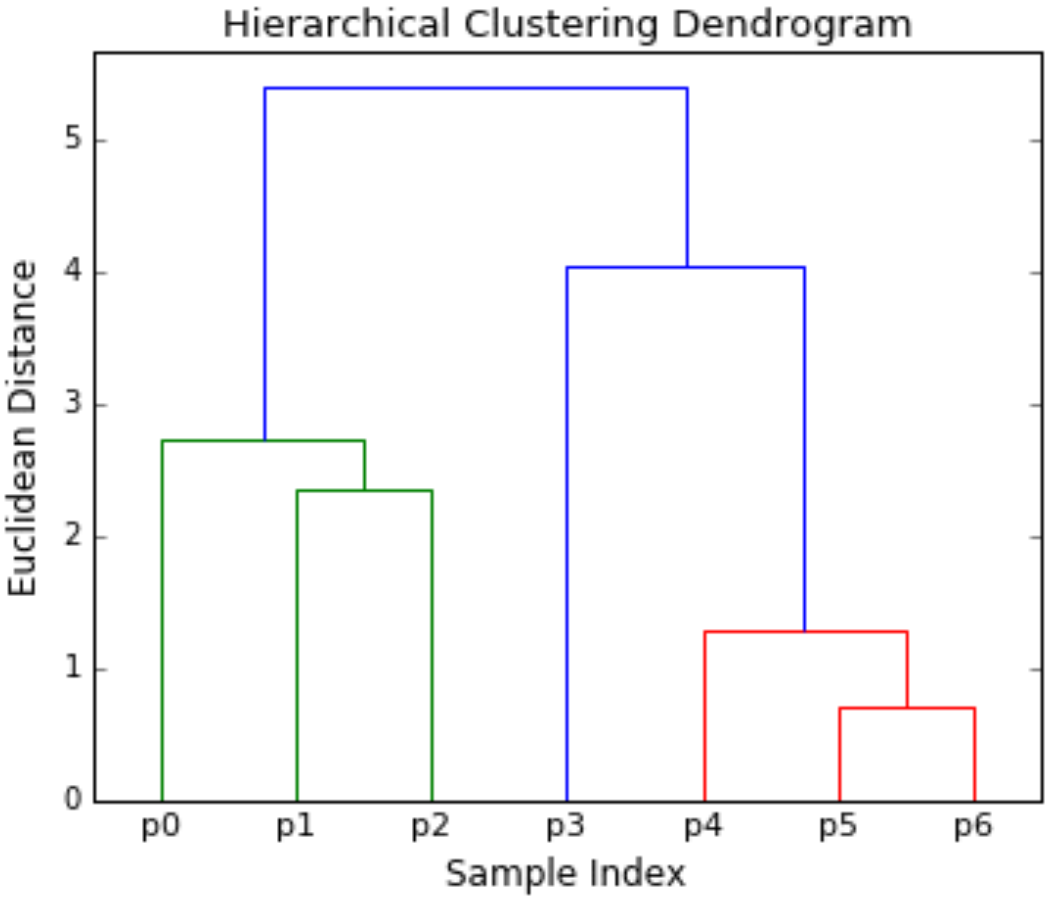


Agglomerative Hierarchical Clustering

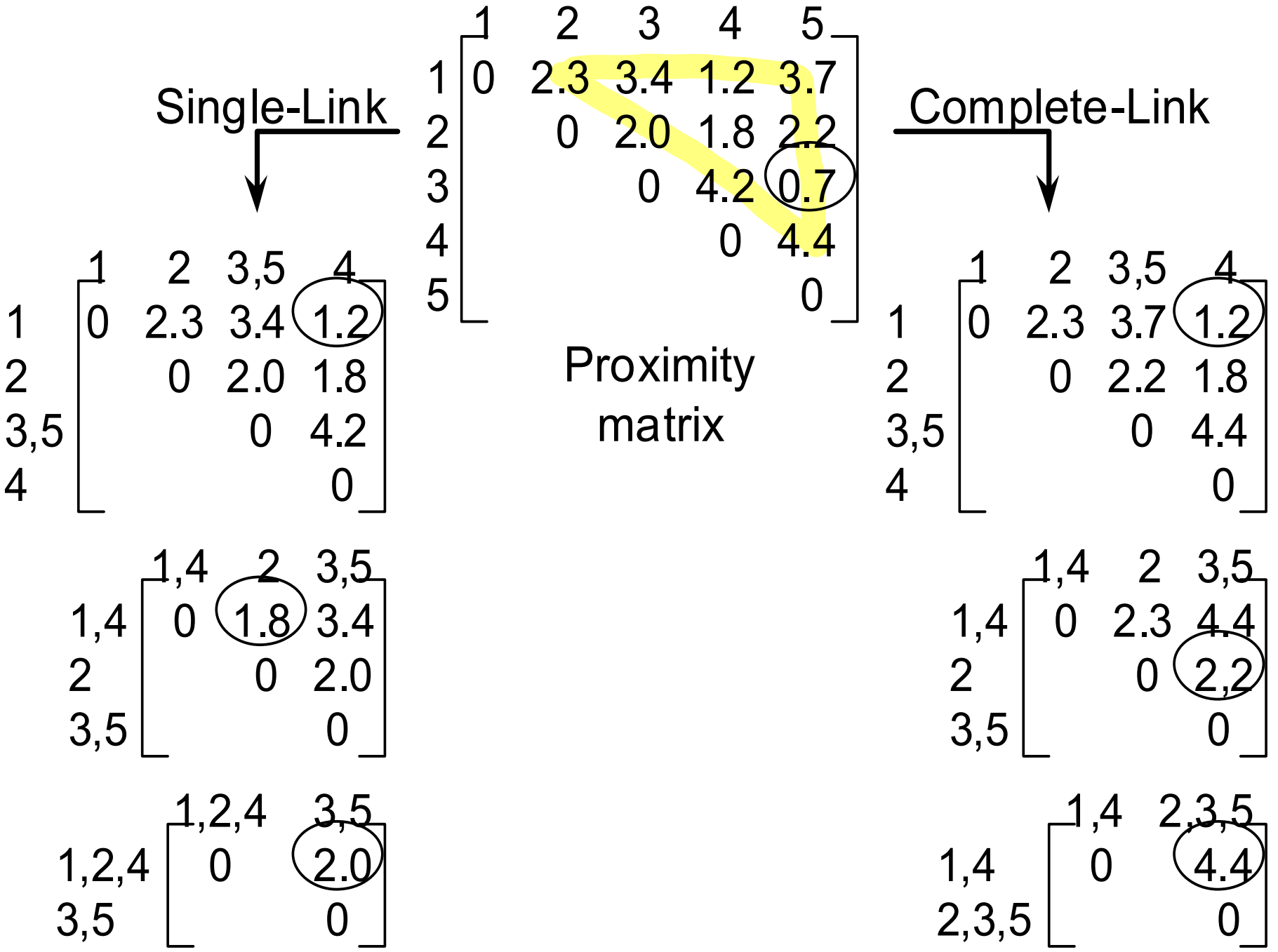
- Algorithm
  1. Compute the proximity matrix, if necessary
  2. Repeat
  3. Merge the closest two clusters
  4. Update the proximity matrix to reflect the proximity between the new cluster & original clusters
  5. Until only one cluster remains

Hierarchical Clustering

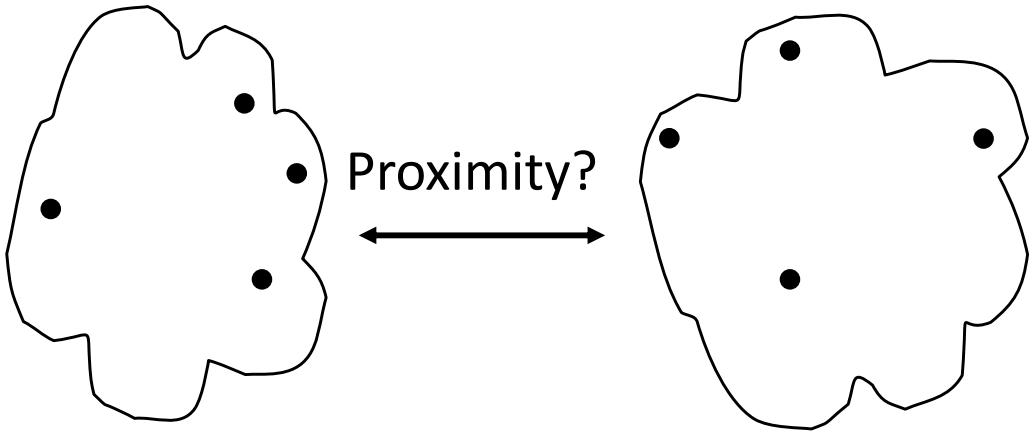




upper triangular matrix : stored in memory using an array



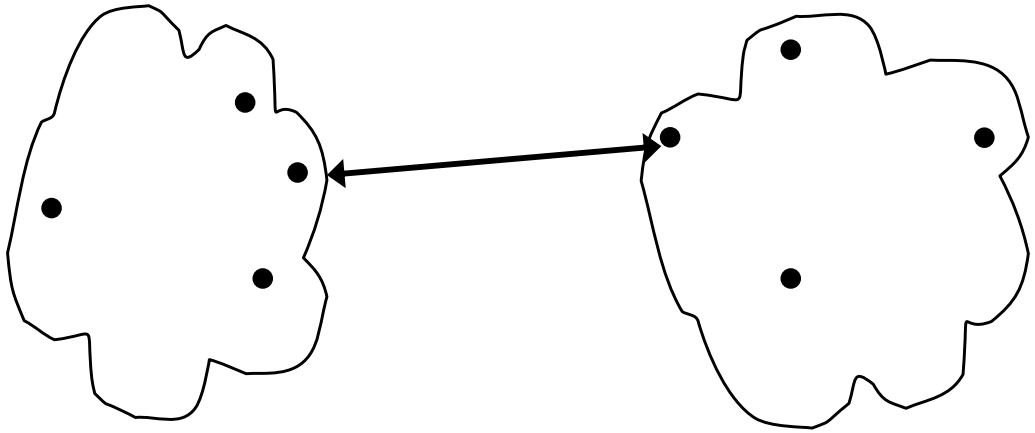
Proximity Between Two Clusters



- MIN
- MAX
- Average
- Distance Between Centroids

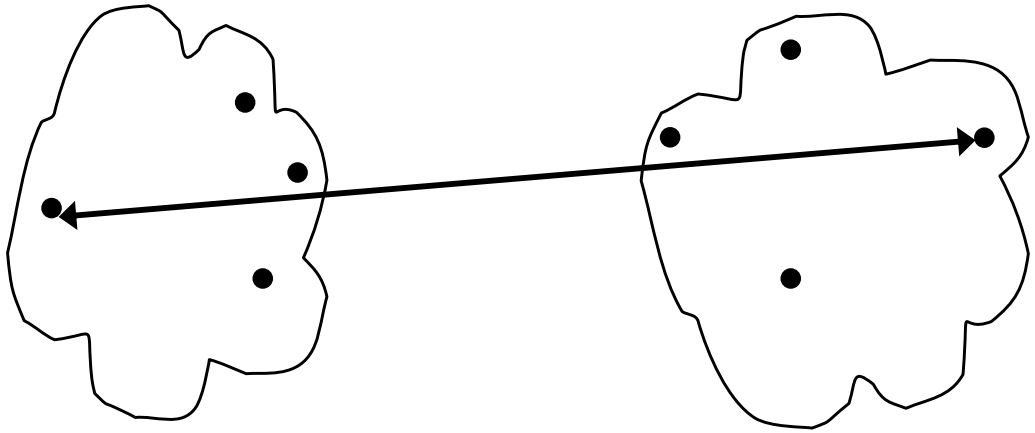
Proximity (Distance) between Clusters

- Single link: minimum distance



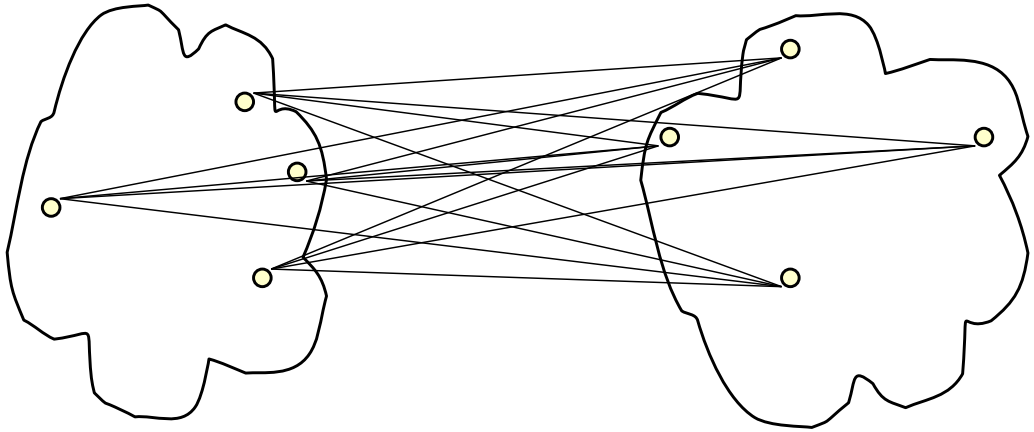
Proximity between Clusters (cont.)

- Complete link: maximum distance



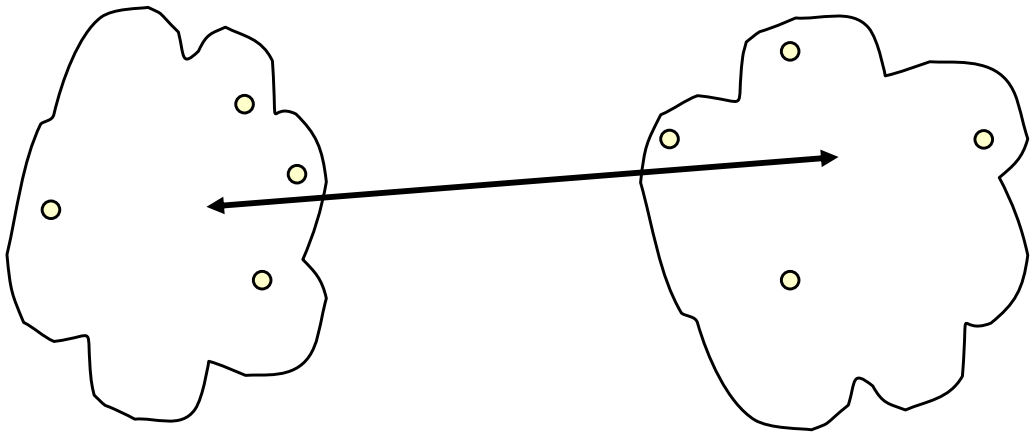
Proximity between Clusters (cont.)

- Average link: average distance

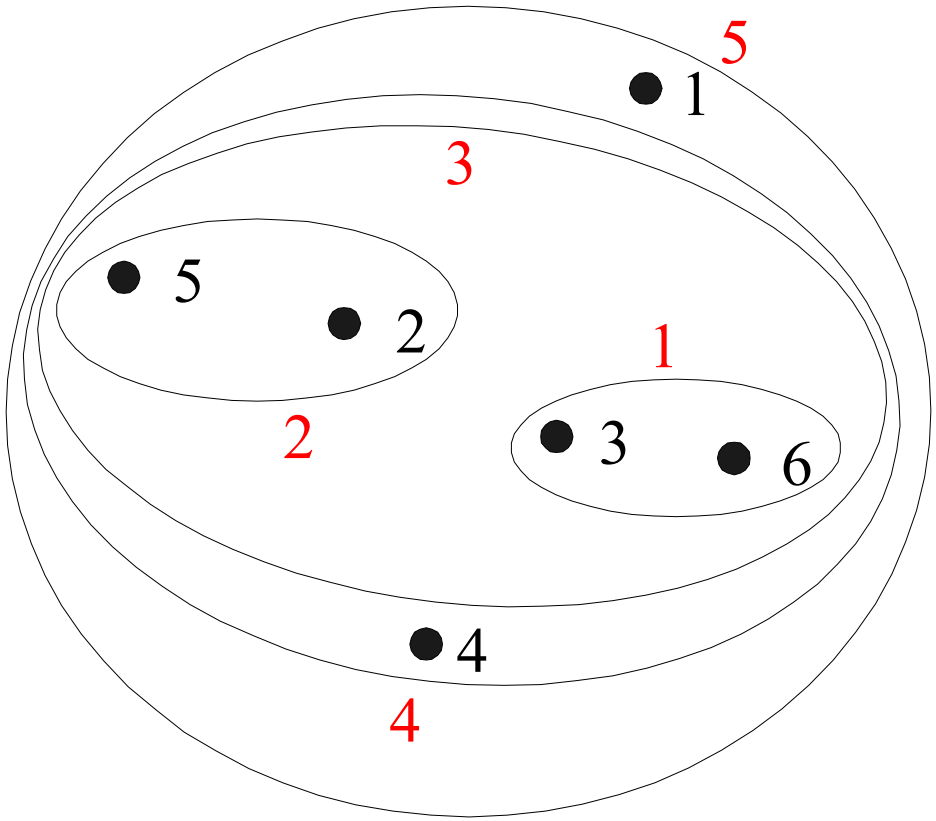


Proximity between Clusters (cont.)

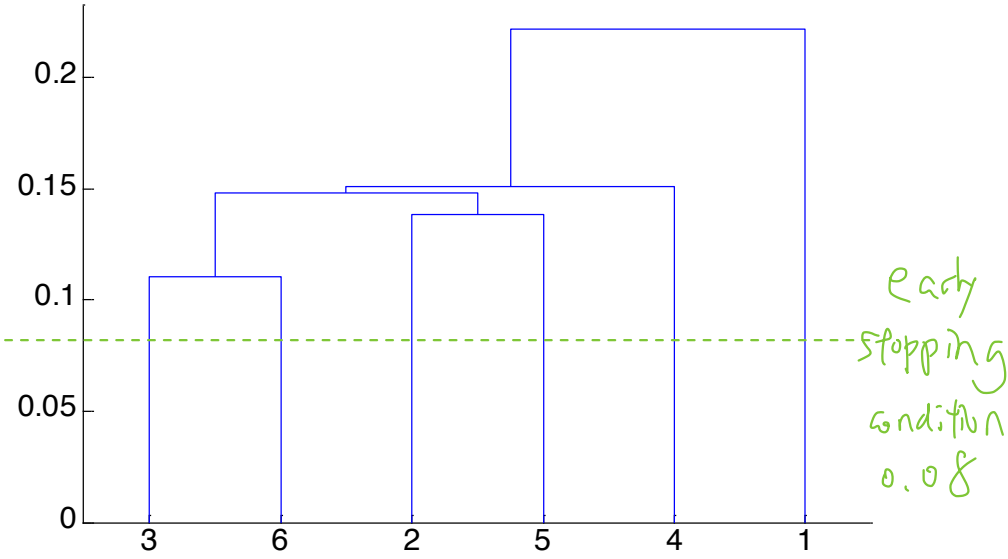
- Mean distance:



Hierarchical Clustering: Single Link (Min)

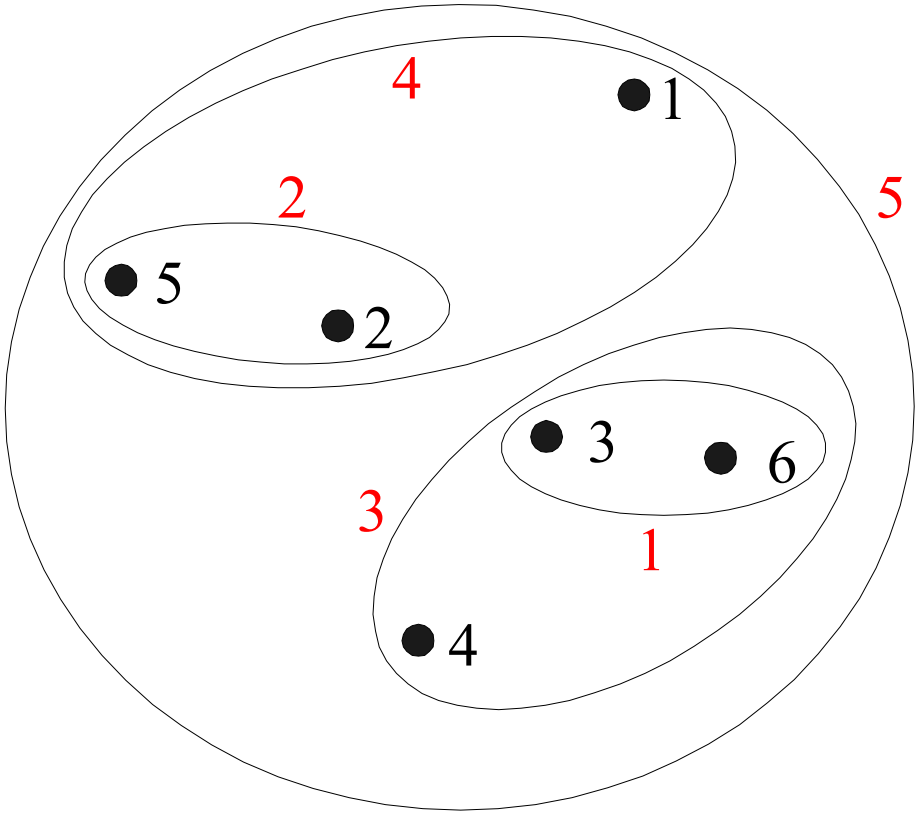


Nested Clusters

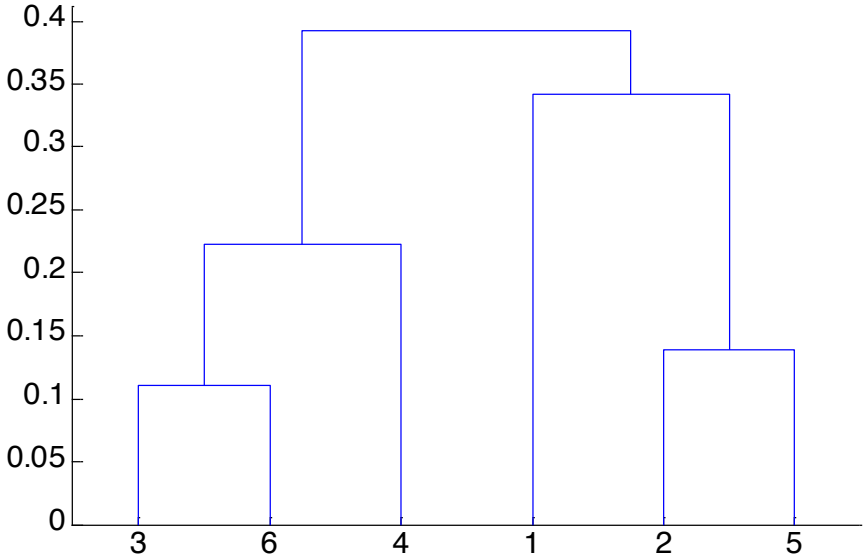


Dendrogram

Hierarchical Clustering: Complete Link (Max)

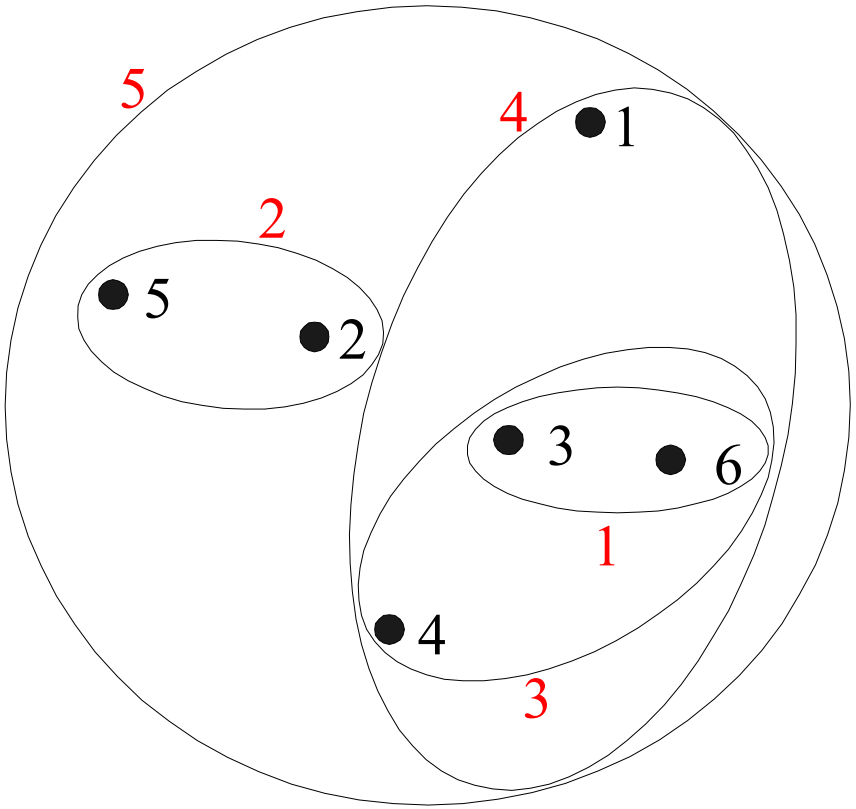


Nested Clusters

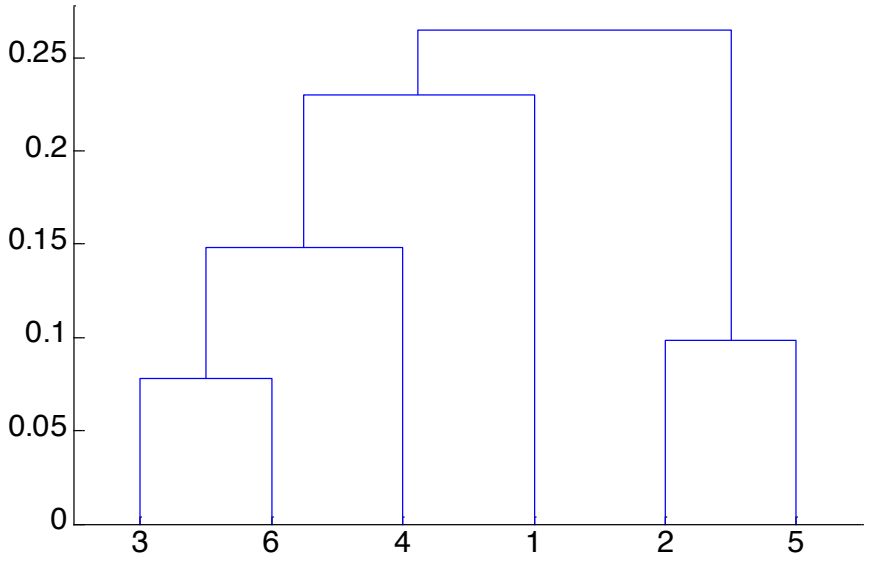


Dendrogram

Hierarchical Clustering: Average Link

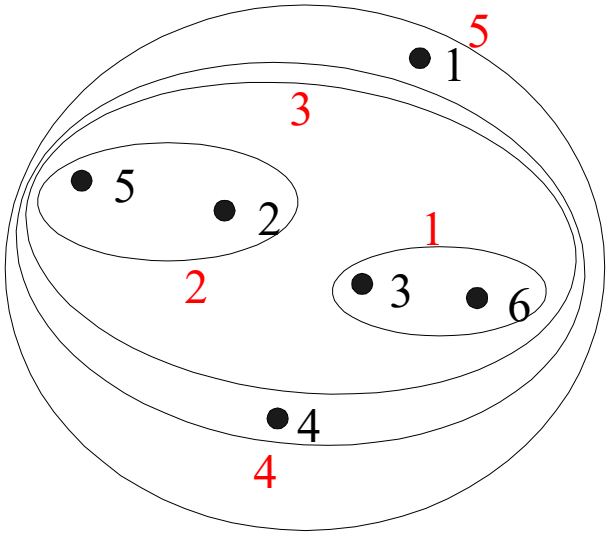


Nested Clusters



Dendrogram

Agglomerative Hierarchical Clustering: Comparison

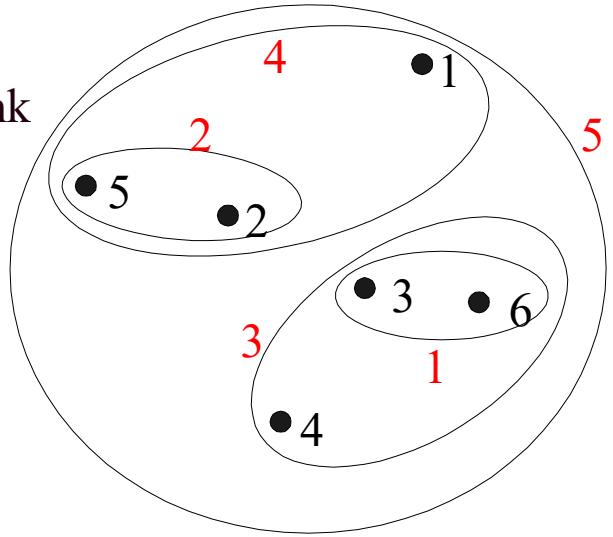


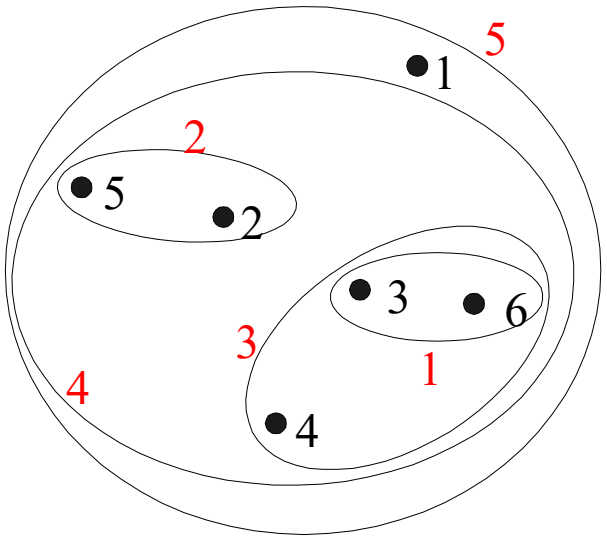
Single Link

(MIN)

Complete Link

(MAX)





Average Link

(Average)

Comments on Hierarchical Clustering

- Weakness of agglomerative clustering methods:
  - do not scale well: time complexity of at least $O(n^2 \log n)$, where n is the number of total objects
  - can never undo what was done previously.

① Distance Matrix Construction: $O\left(\frac{n(n-1)}{2}\right) = O(n^2)$

② Sorting Distances: O(sorting $O(n^2)$ distances) = $O(n^2 \log(n^2))$

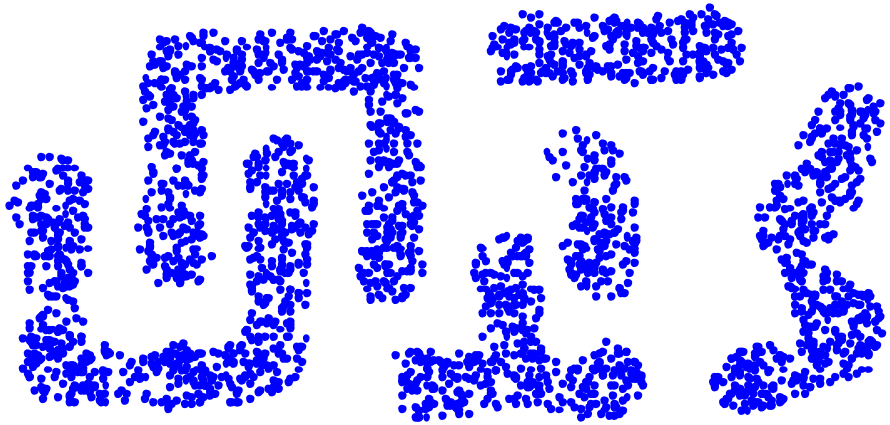
= $O(n^2 \log(n))$

③ Merging Clusters: O(updating distances in the matrix) = $O(n^2)$

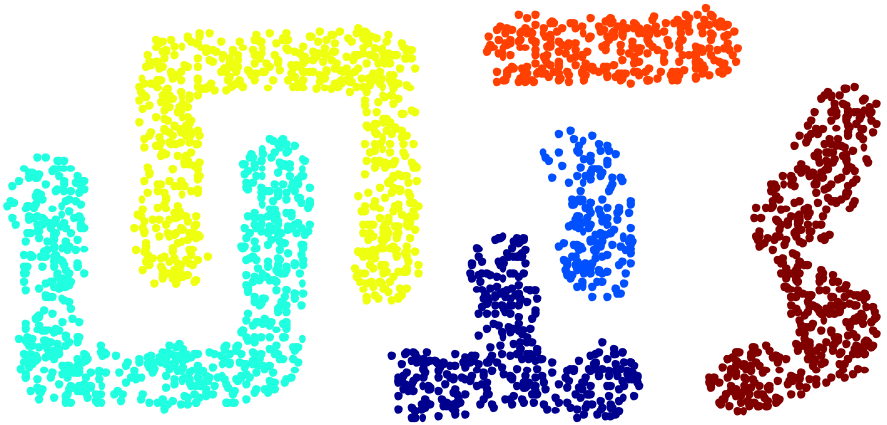
④ From ①, ②, and ③, the time complexity is $O(n^2 \log n)$

Strength of MIN

- **Can handle non-elliptical shapes**



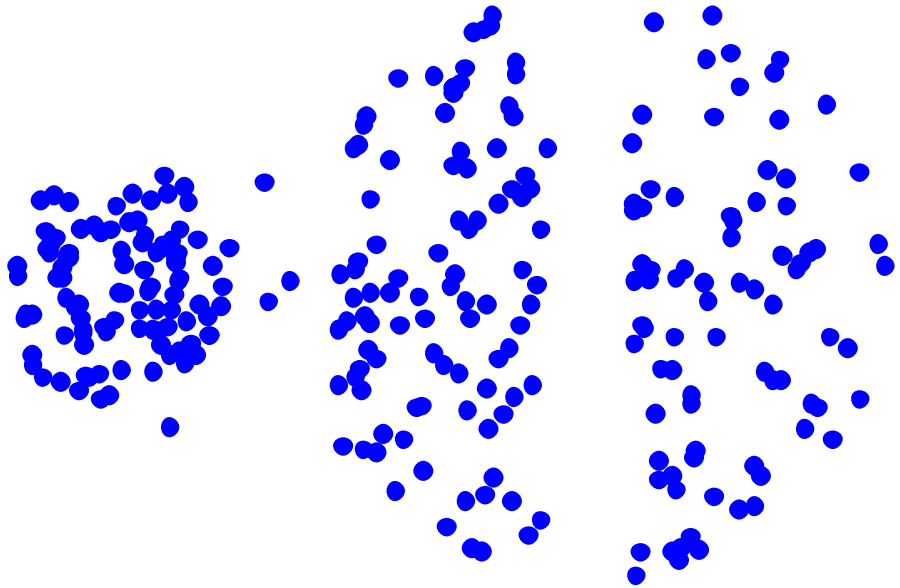
Original Points



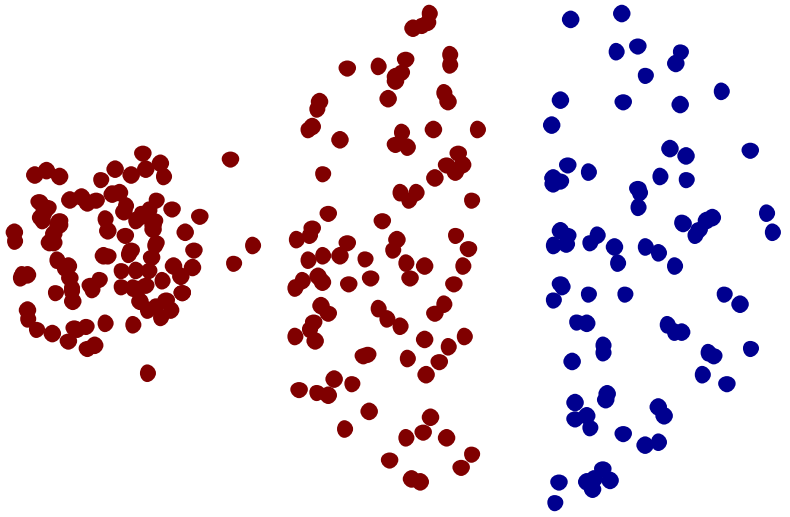
Six Clusters

Limitations of MIN

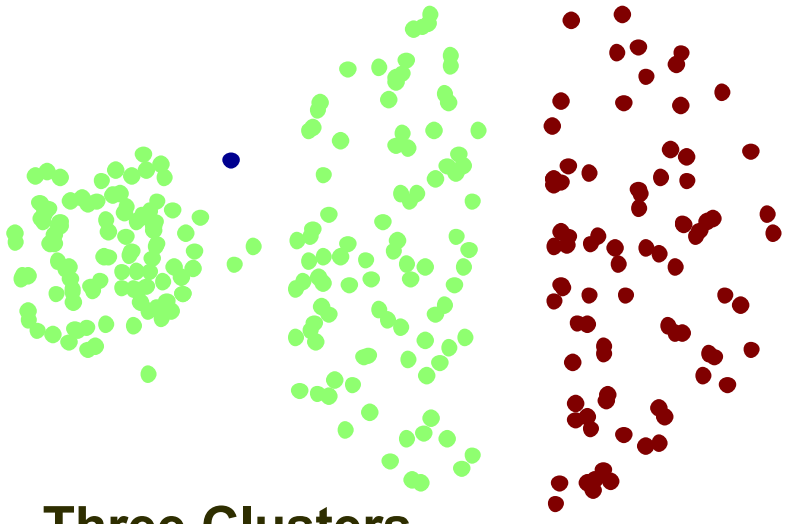
- **Sensitive to noise**



Original Points



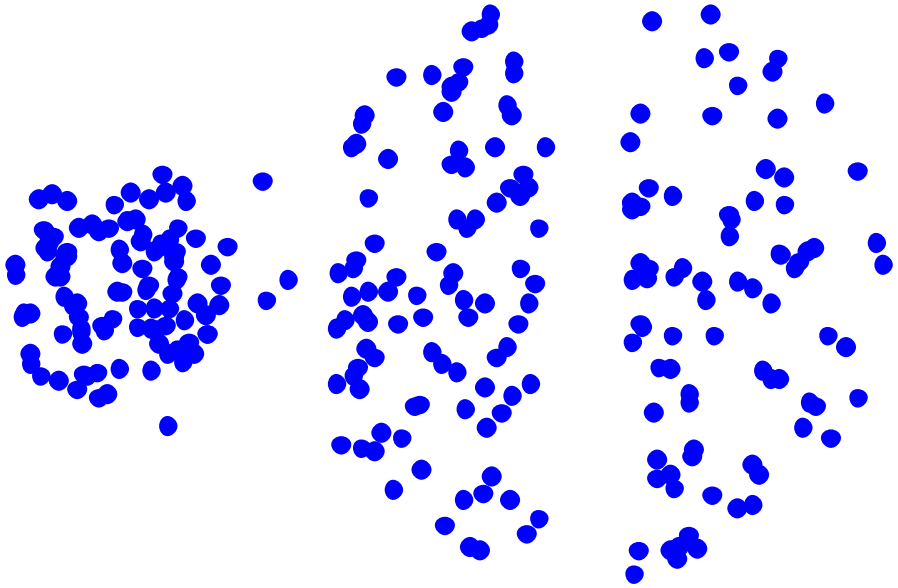
Two Clusters



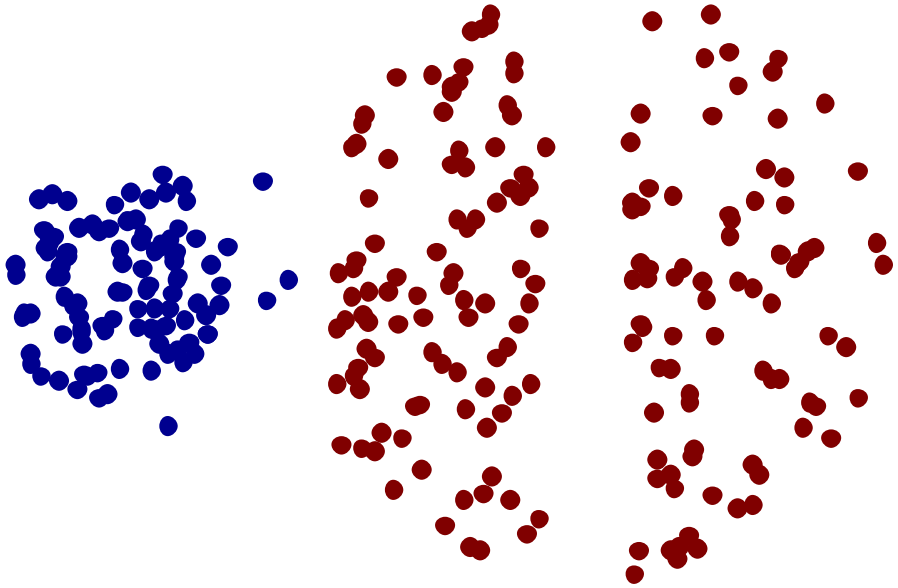
Three Clusters

Strength of MAX

- Less susceptible to noise



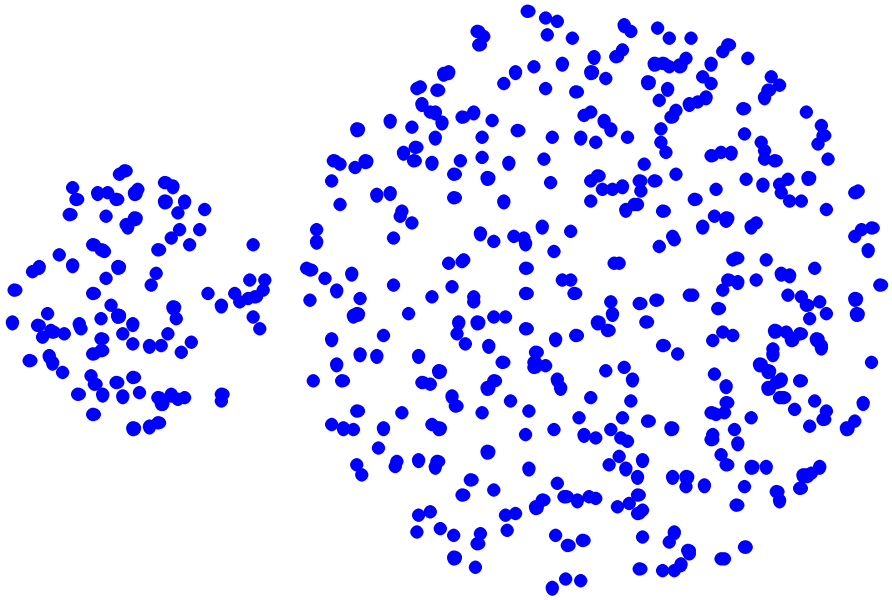
Original Points



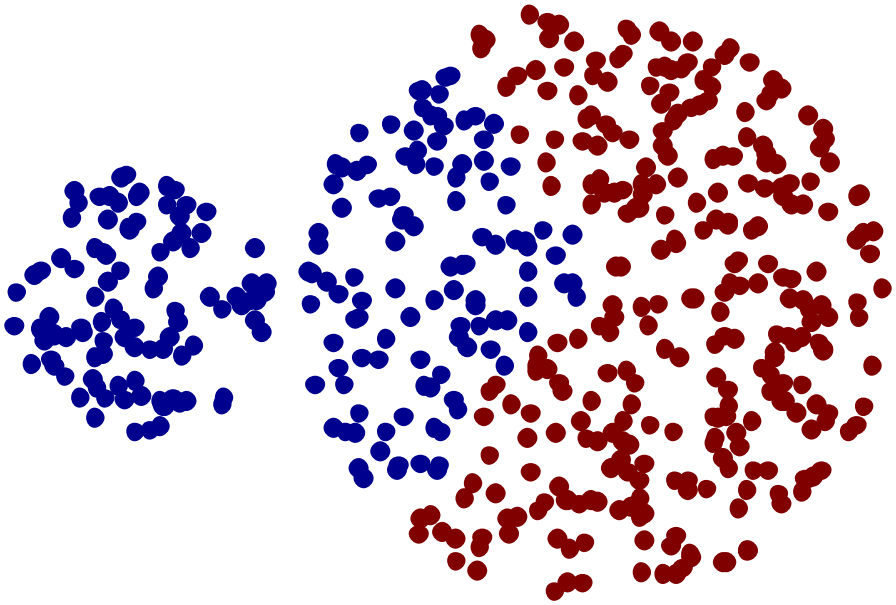
Two Clusters

Limitations of MAX

- Tends to break large clusters
- Biased towards globular clusters



Original Points

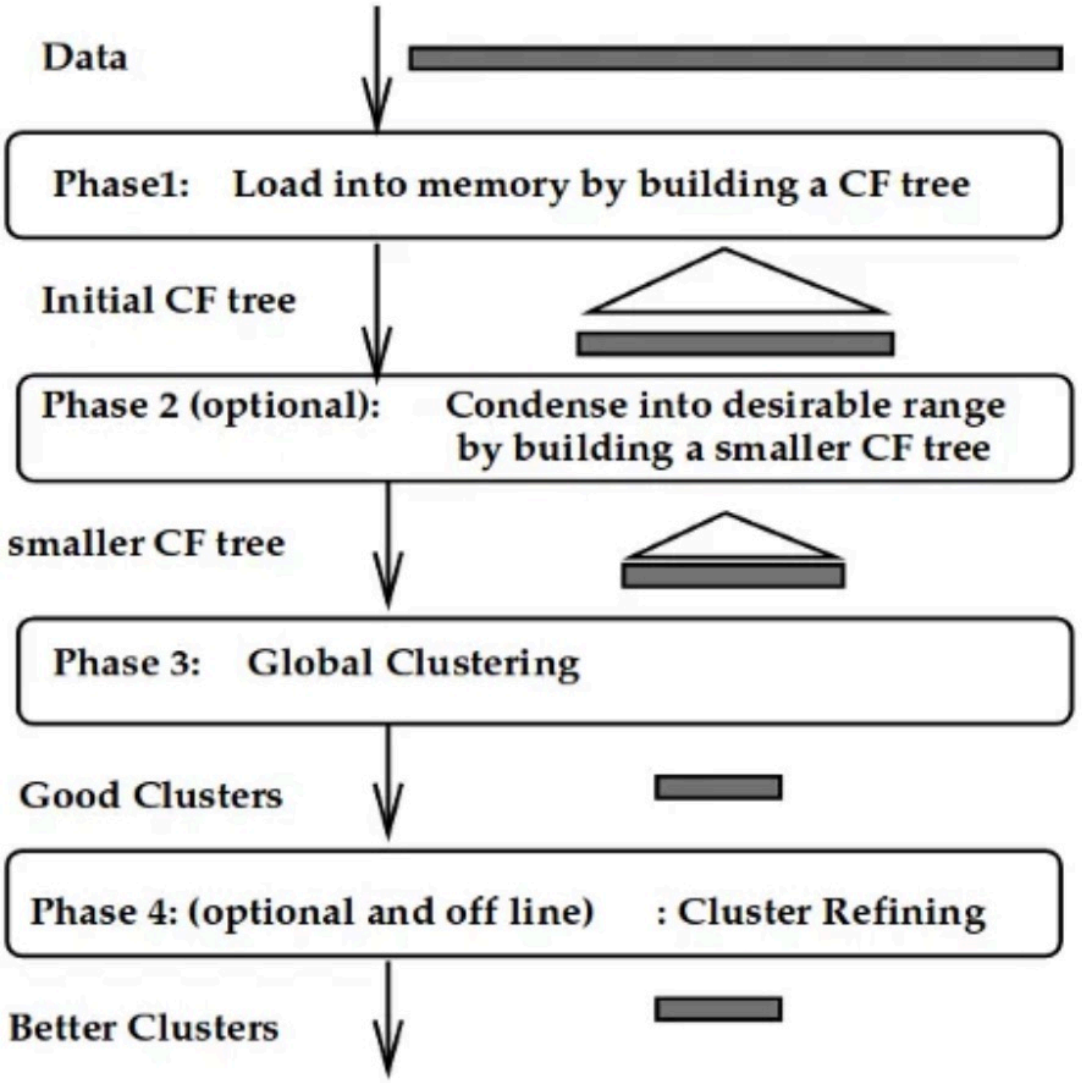


Two Clusters

BIRCH

- Birch (Balanced Iterative Reducing and Clustering using Hierarchies, ’99)
  - is designed for clustering a large amount of numerical data
  - integration of
    - hierarchical clustering (macro-clustering)
    - iterative partitioning (micro-clustering)
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree

THE BIRCH CLUSTERING ALGORITHM:



An outline of the BIRCH Algorithm

BIRCH (cont.)

Clustering Feature: *CF = (N, LS, SS)*

N: **Number of data points**

LS: $\sum_1^N x_i$ (Linear Sum)

SS: $\sum_1^N x_i^2$ (Squared Sum)

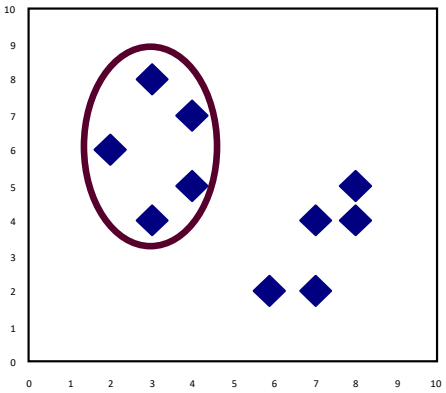
↓ allows efficient computation

Centroid *C*: *LS*/*N*

Radius *R*: $\sqrt{\frac{\sum_1^N (x_i - C)^2}{N}}$

$$= \sqrt{\frac{\sum_1^N (x_i^2 - 2x_i C + C^2)}{N}}$$

$$= \sqrt{\frac{SS - 2C \times LS + N \times C^2}{N}} = \sqrt{\frac{SS}{N} - \left(\frac{LS}{N}\right)^2}$$



LS *SS*

CF = (5, (16,30),(54,190))

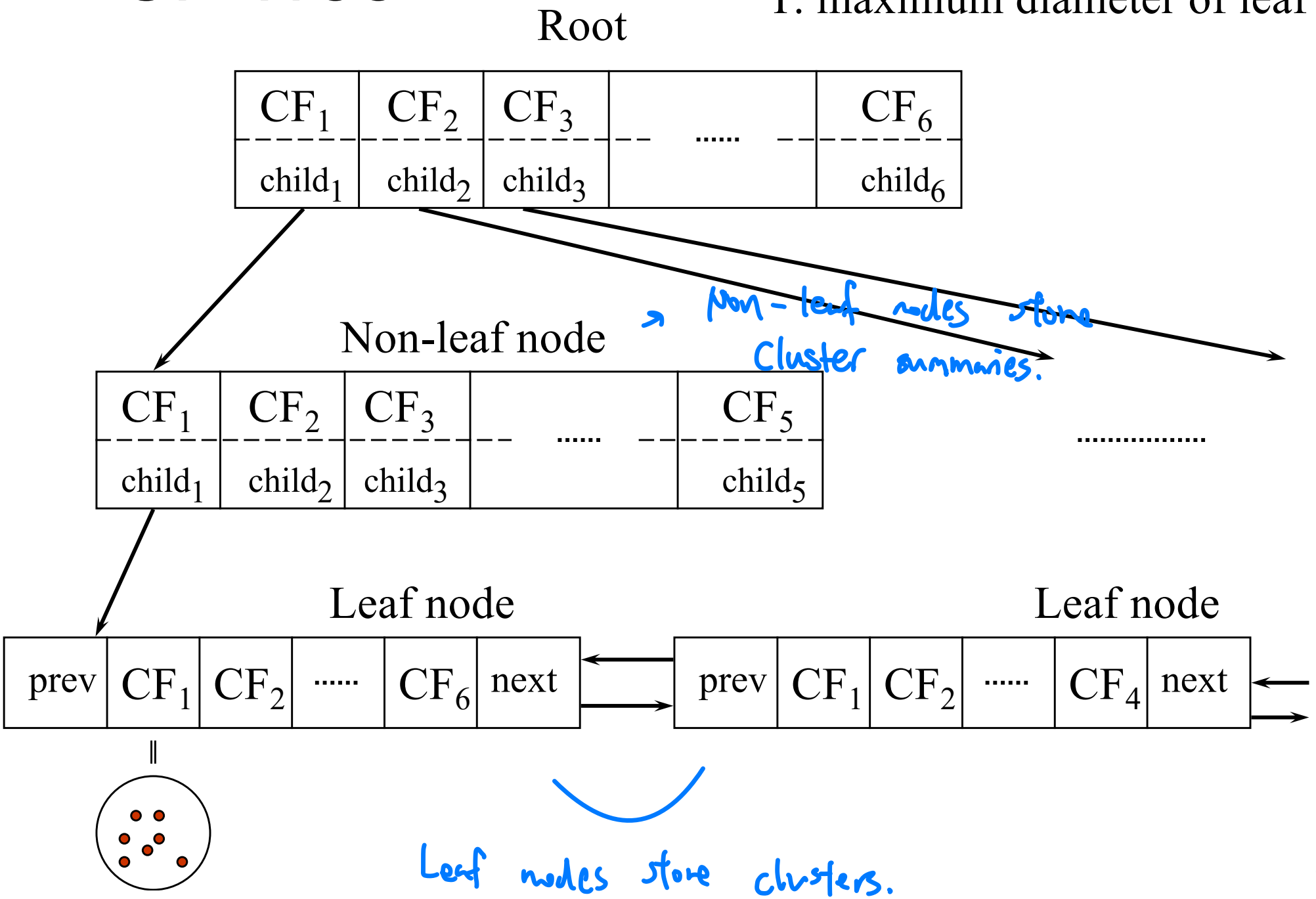
(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

MKShan

CF Tree

B: branching factor

T: maximum diameter of leaf



PARAMETERS OF BIRCH:

There are three parameters in this algorithm, which needs to be tuned.

1. *threshold* : threshold is the maximum number of data points a sub-cluster in the leaf node of the CF tree can hold.
2. *branching factor* : This parameter specifies the maximum number of CF sub-clusters in each node (internal node).
3. *number of clusters* : The number of clusters to be returned after the entire BIRCH algorithm is complete i.e., number of clusters after the final clustering step. If set to None, the final clustering step is not performed and intermediate clusters are returned.

BIRCH (cont.)

- Scales linearly
  - finds a good clustering with a single scan
  - improves the quality with a few additional scans
- Weakness
  - handles only numeric data
  - sensitive to the order of the data record

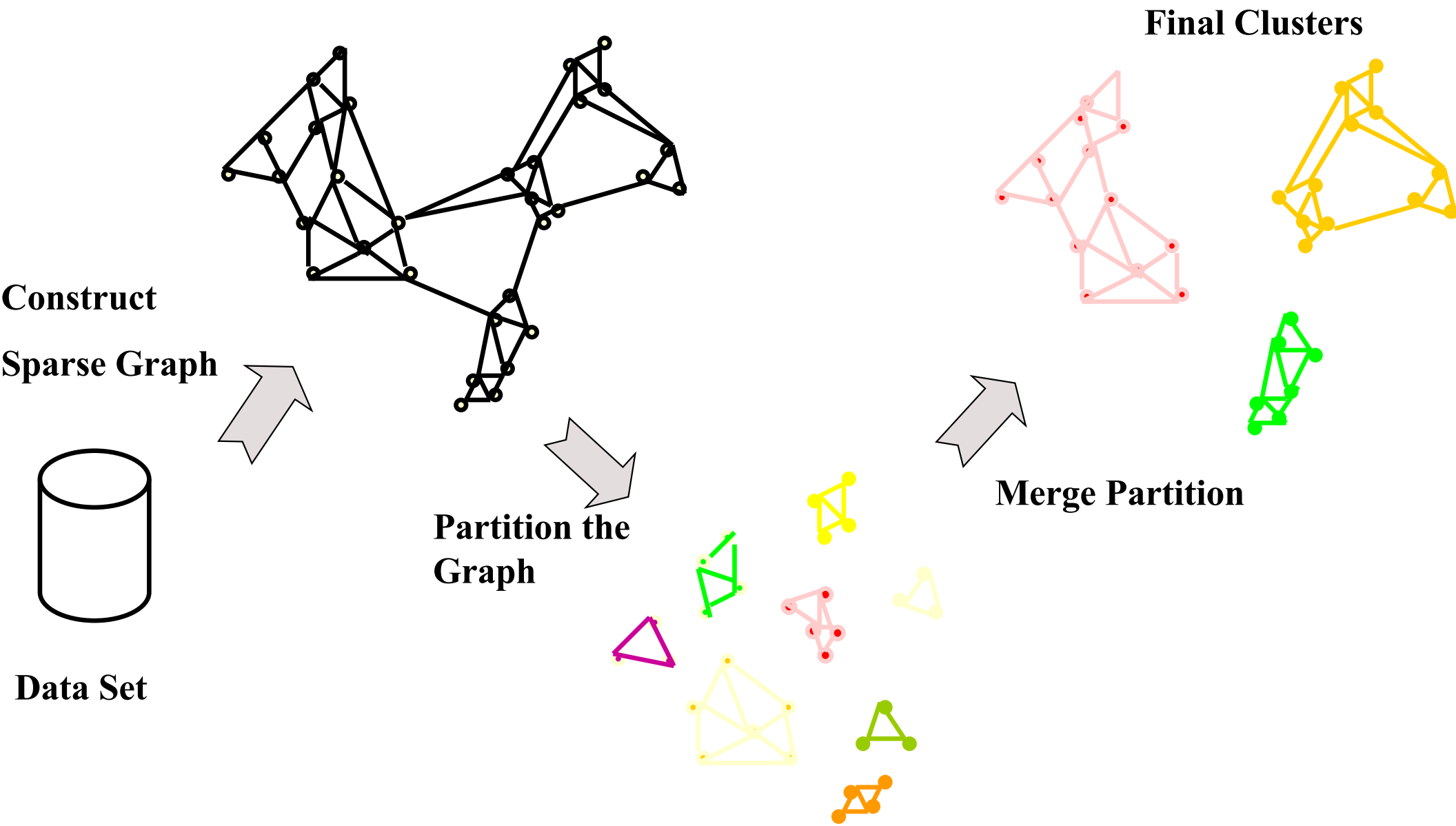
CHAMELEON

- CHAMELEON (Hierarchical Clustering Using Dynamic Modeling, '99)
- Measures the similarity based on a dynamic model
  - Cluster similarity is assessed based on
    - how well-connected objects are within a cluster
    - the proximity of clusters
  - Two clusters are merged if
    - their interconnectivity is high
    - they are close together

CHAMELEON (cont.)

- Algorithm
  1. Uses a k-nearest-neighbor graph to construct a sparse weighted graph
  2. Use a graph partitioning algorithm to partition the k-nearest-neighbor graph into a large number of relatively small sub-clusters
  3. Use an agglomerative hierarchical clustering algorithm that repeatedly merges subclusters based on their similarity.

CHAMELEON (cont.)



CHAMELEON (cont.)

- K-nearest-neighbor graph
  - each vertex represents a data object
  - there exist an edge between two vertices (objects) if one object is among the k-most-similar ones of the other.
- Graph partitioning algorithms
  - Minimum Spanning Tree algorithms or
  - Partitions k-nearest-neighbor graph such that the edge cut is minimized

Minimum Spanning Tree Hierarchical Clustering Algorithm

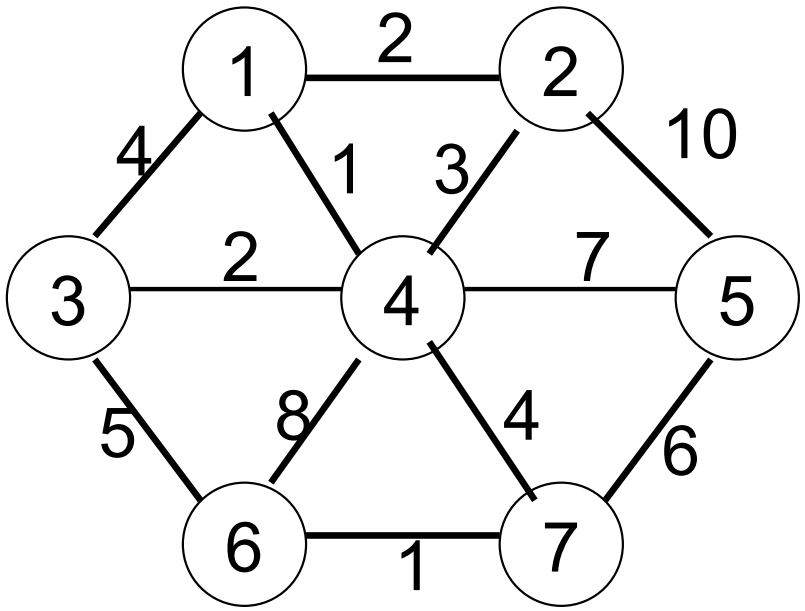
- Algorithm

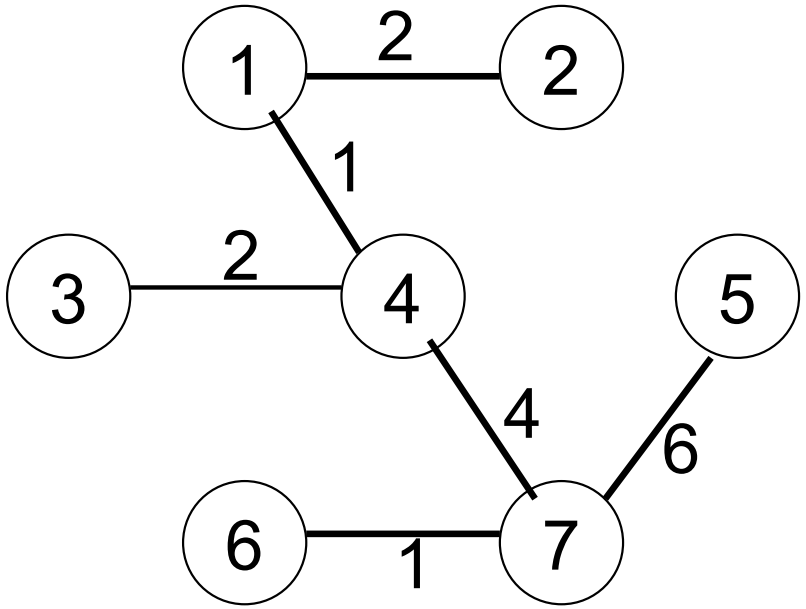
1. Compute a minimum spanning tree for the k-nearest neighbor graph
2. Repeat
3. Create a new cluster by breaking the link corresponding to the largest dissimilarity (distance)
4. Until only singleton clusters remain

* Divisive hierarchical clustering algorithm

Minimum Spanning Tree
Hierarchical Clustering Algorithm (cont.)

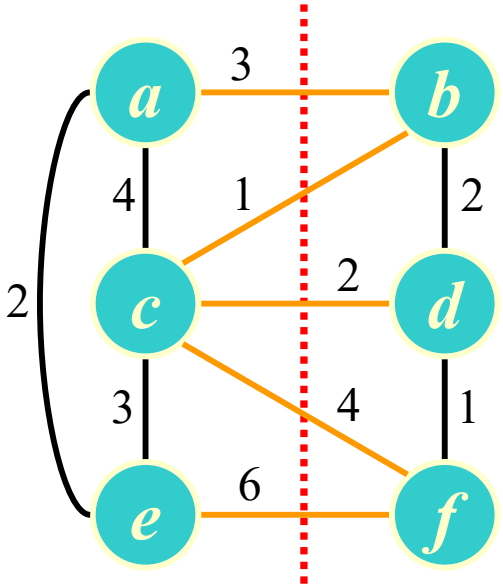
- Spanning Tree: a tree formed from graph edges that connects all the vertices
- Problem: Given an undirected connected weighted graph $G = (V, E)$, Find a spanning tree T of G of minimum cost.

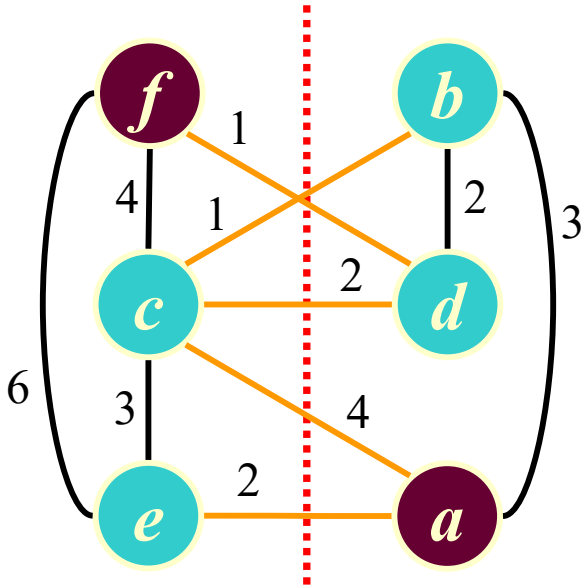




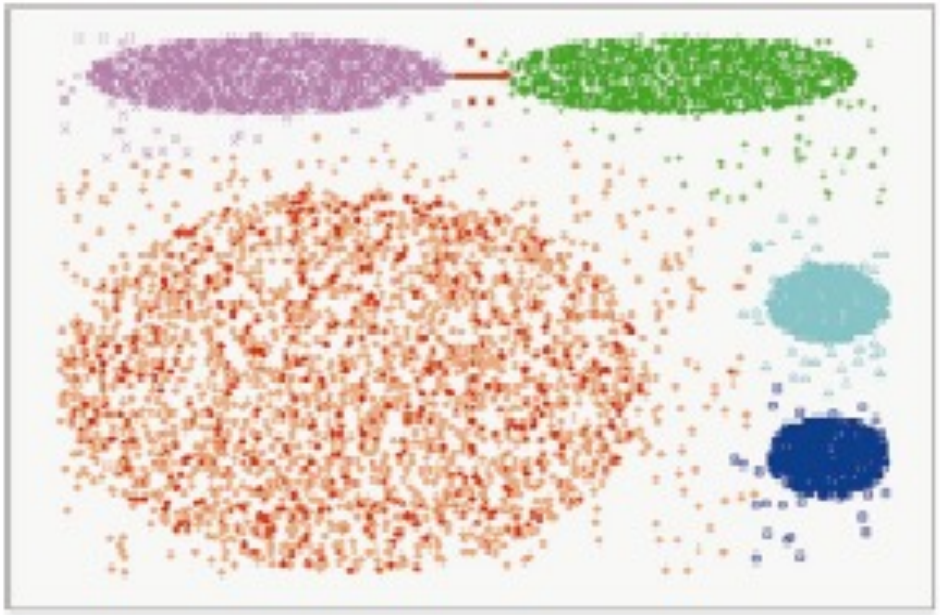
Cut in Graph

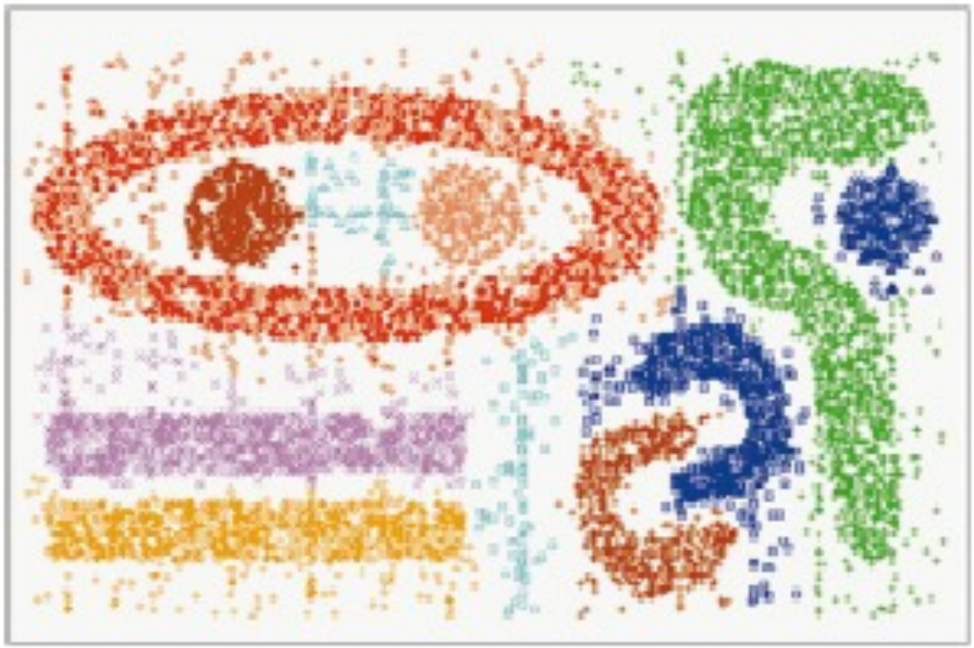
- Cut: a set of edges that separate two group of vertices
- Precise definition of cut
  - A cut is the set of edges $\{(v, w) \in E\}$ such that $v \in A$ and $w \in B$
- Cost of a cut: sum of cut weights
- Minimum cut for clustering

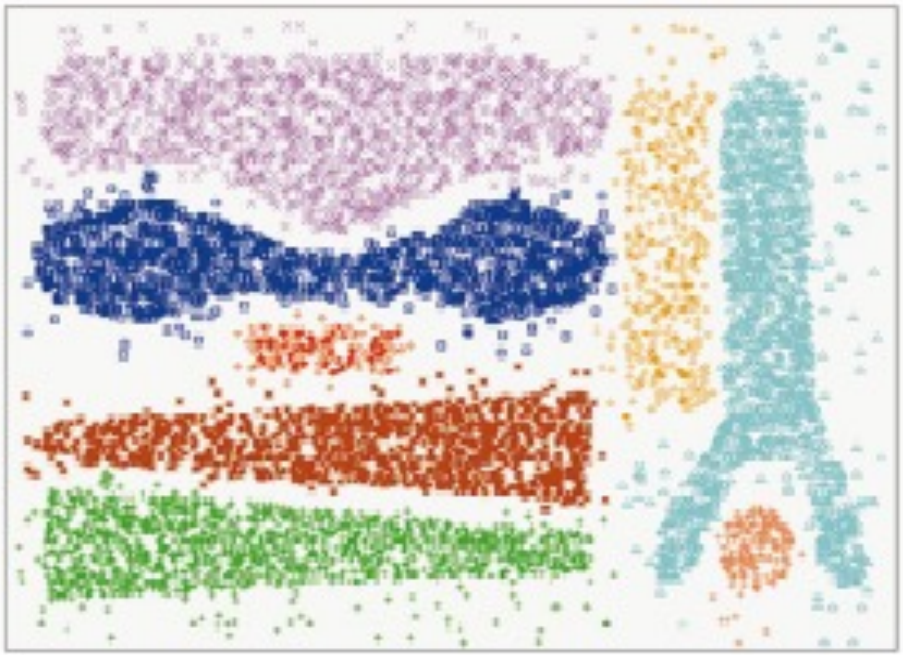




CHAMELEON (Clustering Complex Objects)







Density-based Clustering

Density-Based Clustering Methods

- Clustering based on density (local cluster criterion)
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Approaches
  - DBSCAN (KDD'96)
  - OPTICS (SIGMOD'99).
  - DENCLUE (KDD'98)
  - CLIQUE (SIGMOD'98)

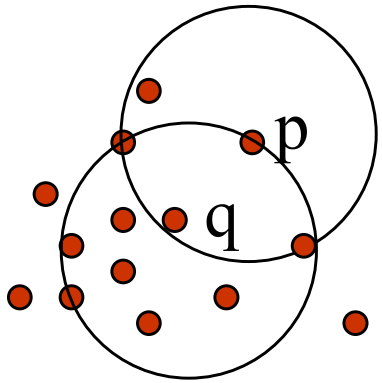
DBSCAN

- DBSCAN: Density Based Spatial Clustering of Applications with Noise.
  - A cluster is defined as a maximal set of density-connected points
  - Discovers clusters of arbitrary shape in spatial databases with noise

DBSCAN (cont.)

- Two parameters:
  - *Eps*: Maximum radius of the neighborhood (epsilon)
  - *MinPts*: Minimum number of points in an *Eps*-neighborhood of that point
- $N_{Eps}(p)$: *{q belongs to D | dist(p,q) <= Eps}*
- Directly density-reachable: A point *p* is directly density-reachable from a point *q* wrt. *Eps*, *MinPts* if
  - 1) *p* belongs to $N_{Eps}(q)$
  - 2) core point condition:

$$|N_{Eps}(q)| >= MinPts$$

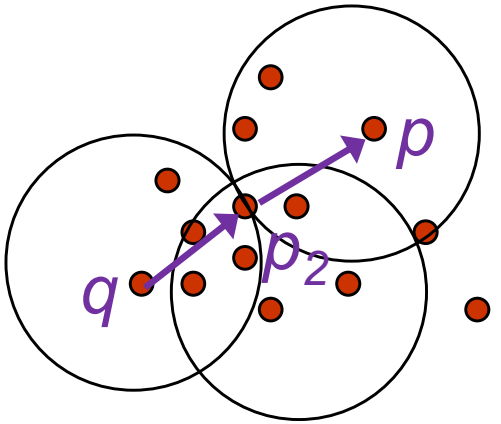


MinPts = 5

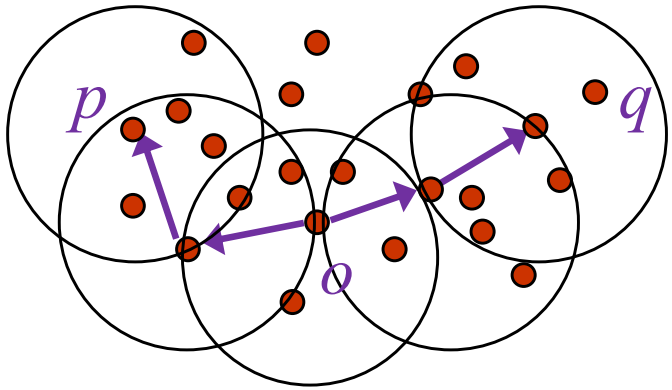
Eps = 1 cm

DBSCAN (Cont.)

- Density-reachable
  - A point *p* is density-reachable from a point *q* wrt. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density-reachable from p_i



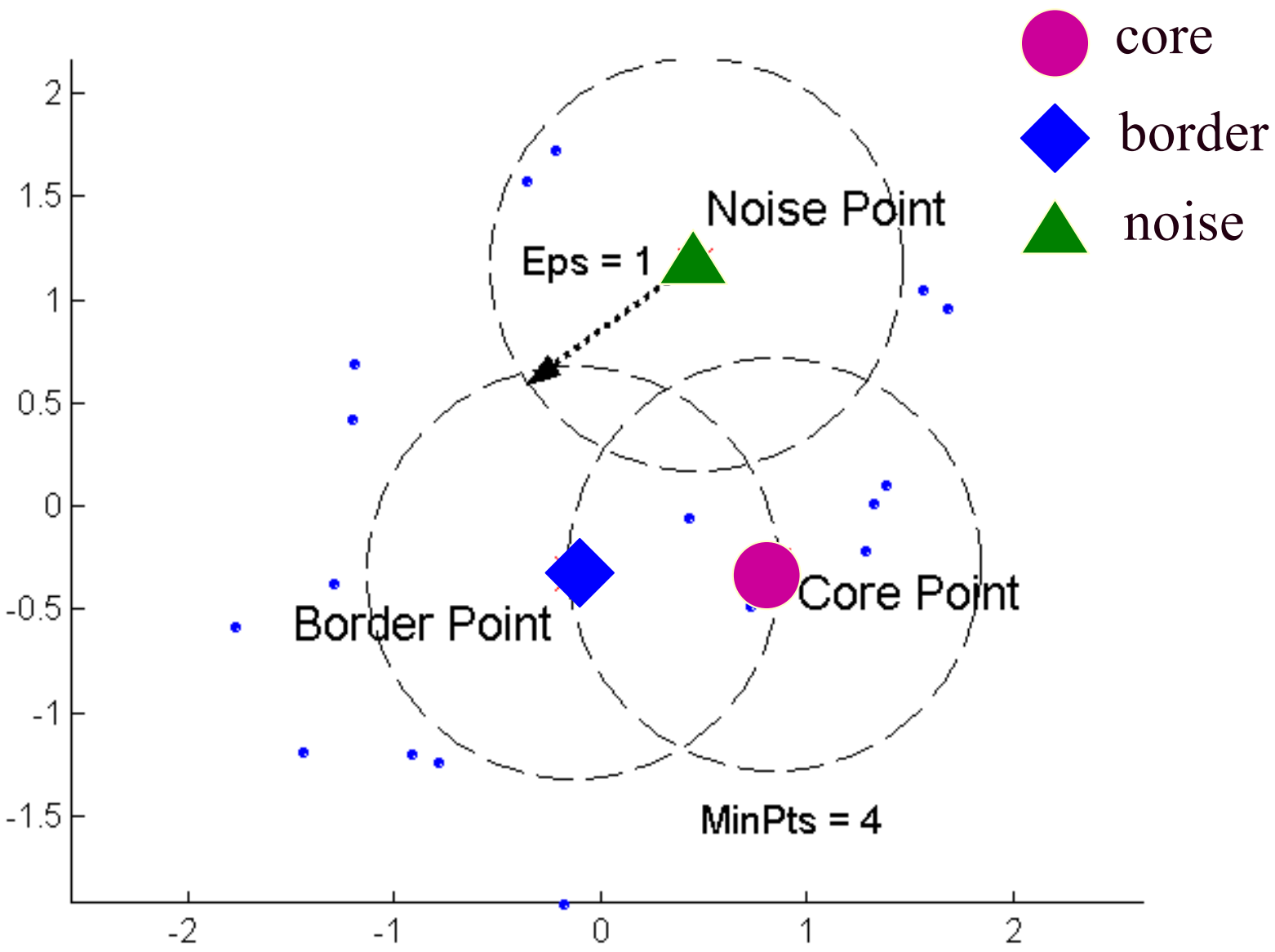
- Density-connected
  - A point *p* is density-connected to a point *q* wrt. *Eps*, *MinPts* if there is a point *o* such that both *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts*.



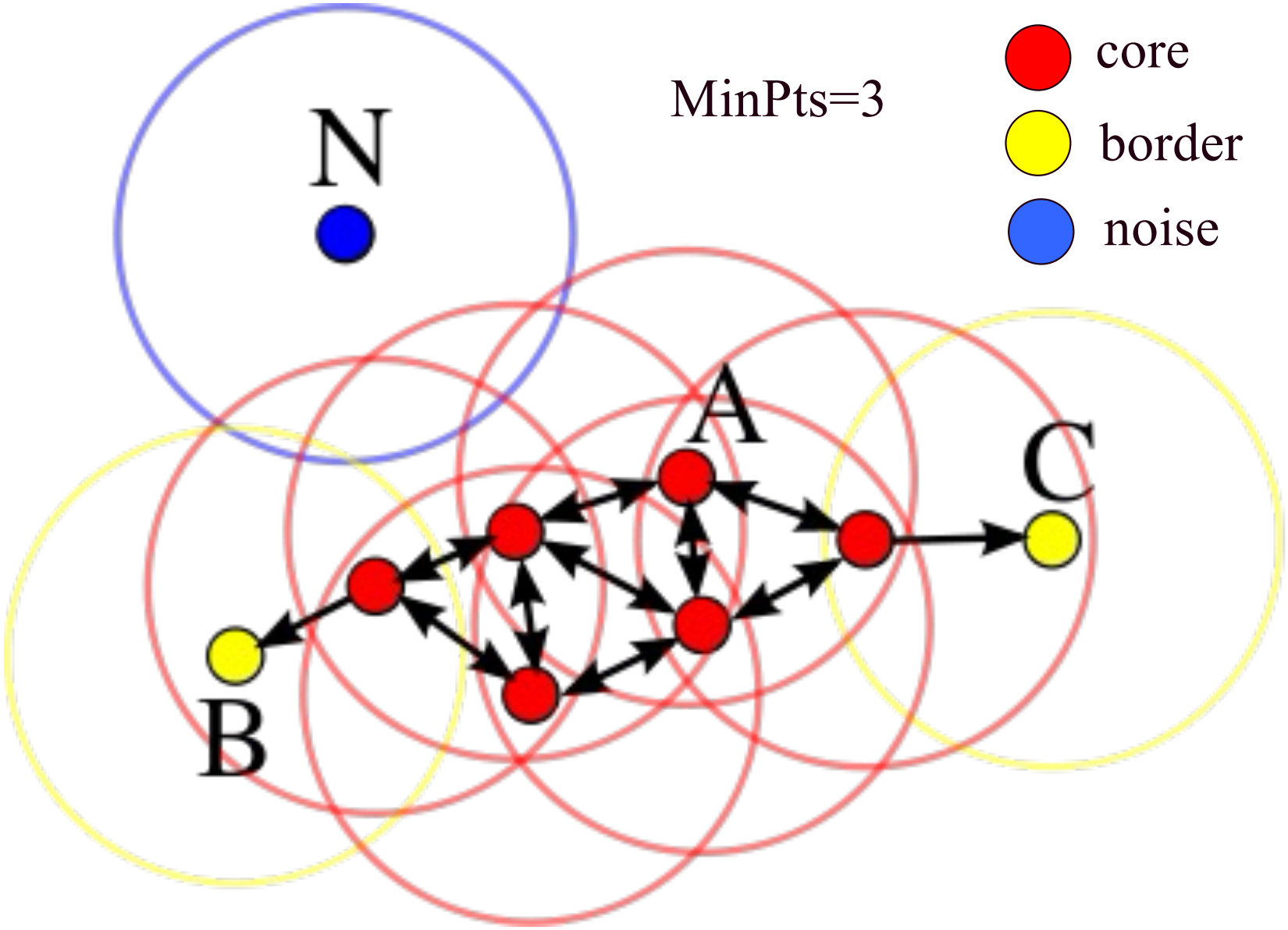
DBSCAN (cont.)

- Density-based Cluster
  - a maximal set of density-connected points
- 3 types of points
  - Core point: a point if it has more than a specified number of points (MinPts) within Eps
  - Border point: a point has fewer than MinPts within Eps, but is in the neighborhood of a core point
  - Noise point: any point that is not a core point or a border point.

DBSCAN:
Core, Border, and Noise Points



DBSCAN:
Core, Border, and Noise Points



DBSCAN (cont.)

- Algorithm

Repeat

Arbitrary select a point ***p***

Retrieve all points density-reachable from ***p*** wrt. ***Eps*** and ***MinPts***. // Density Reachability

If ***p*** is a core point

a cluster is formed by collecting directly density-reachable objects from ***p*** (may merge density-reachable clusters, density-connected)

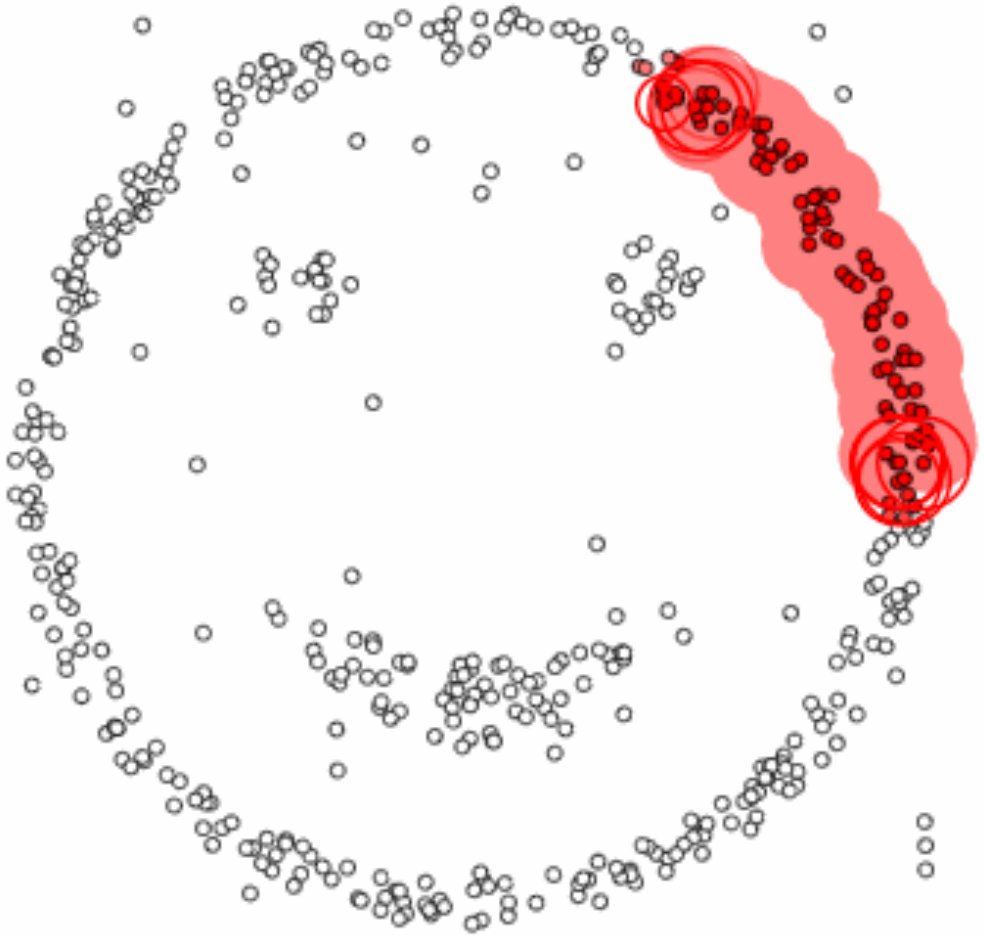
else ***p*** is a border point

no points are density-reachable from ***p***

Until all of the points have been processed.

core point check

DBSCAN



DBSCAN (cont.)

- Algorithm

Label all points as core, border, or noise points.

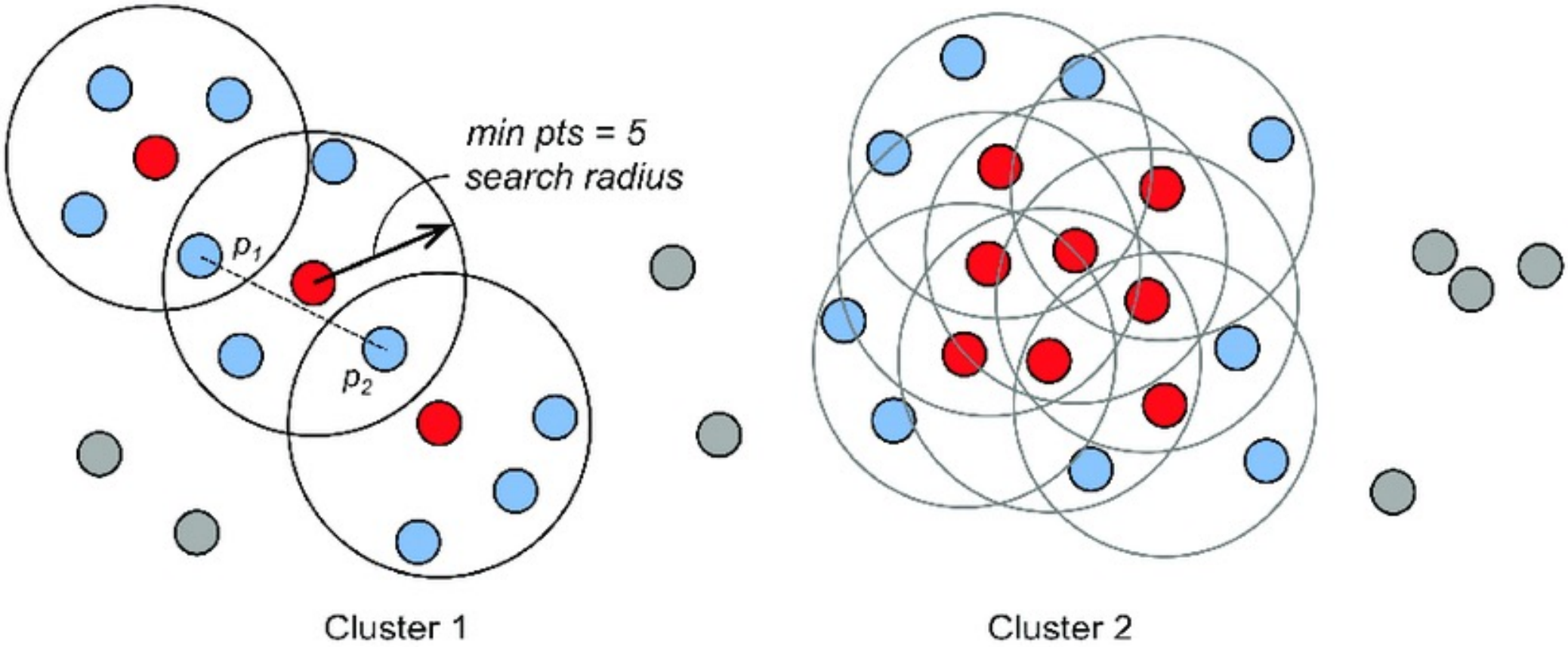
Eliminate noise points

Put an edge between all core points
within a distance ***Eps*** of each other

Make each group of connected core points
into a separate cluster

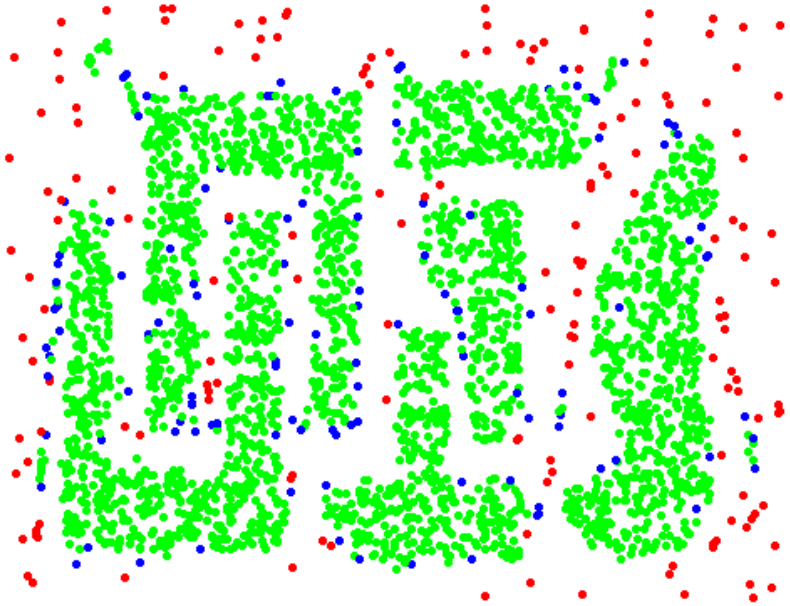
Assign each border point to one of the clusters of
its associated core points

DBSCAN:
Core, Border, and Noise Points



DBSCAN: Core, Border and Noise Points

Original Points



core, border and noise

Eps = 10, MinPts = 4

DBSCAN (cont.)

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

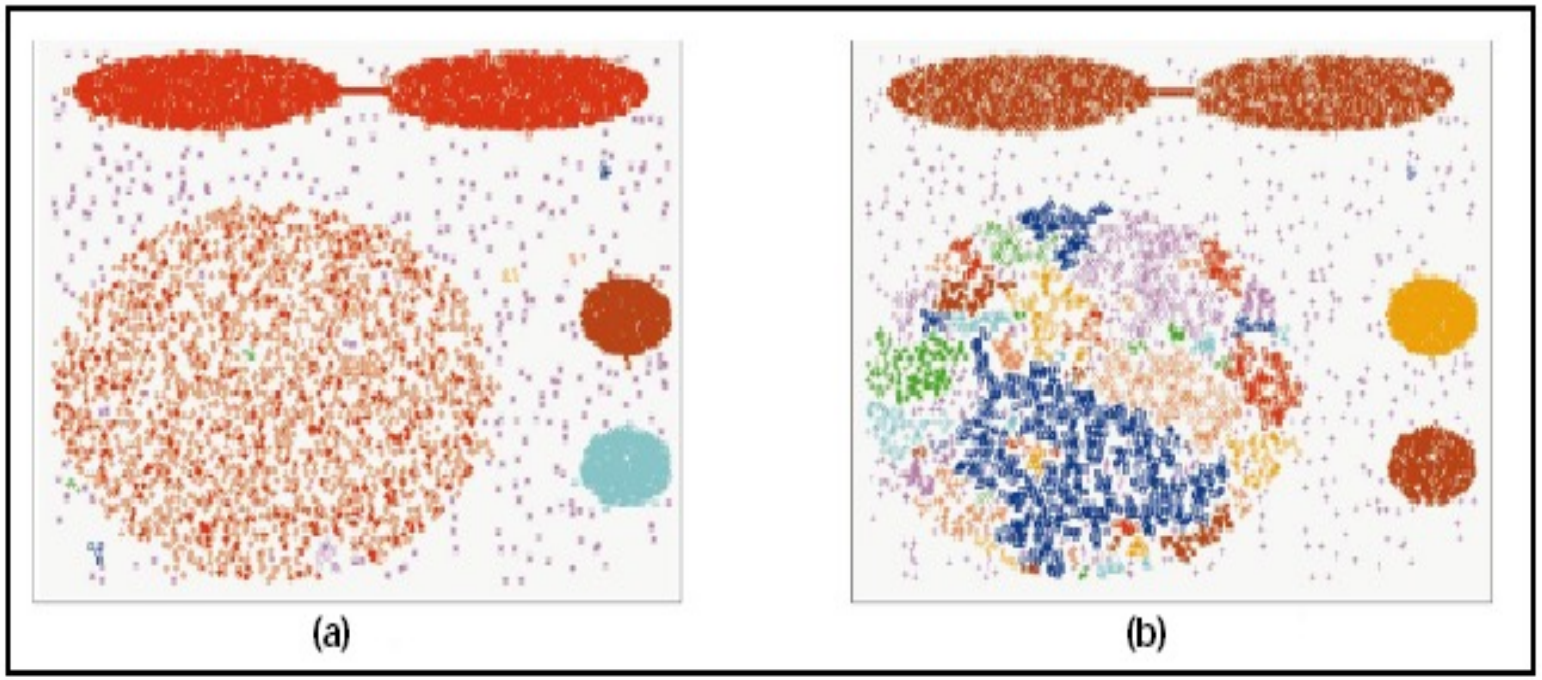
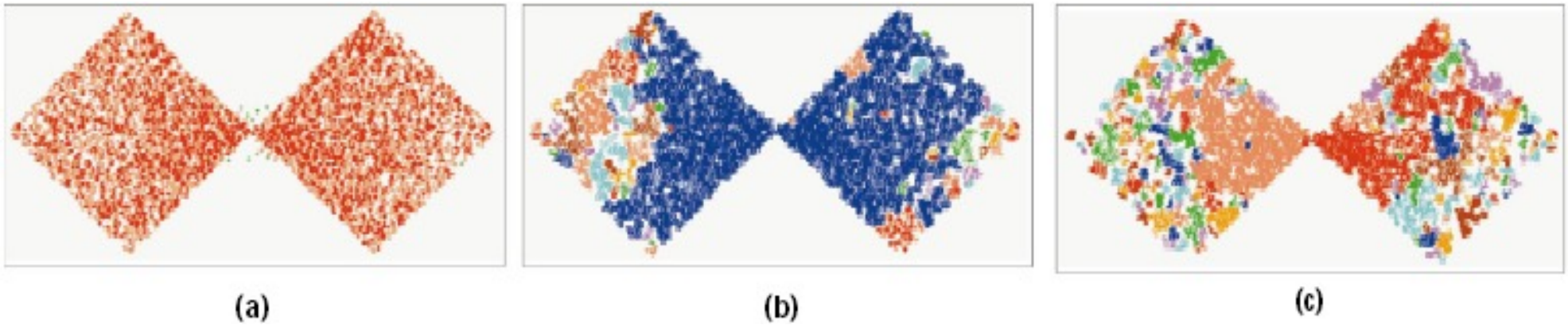


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



DBSCAN

k-means

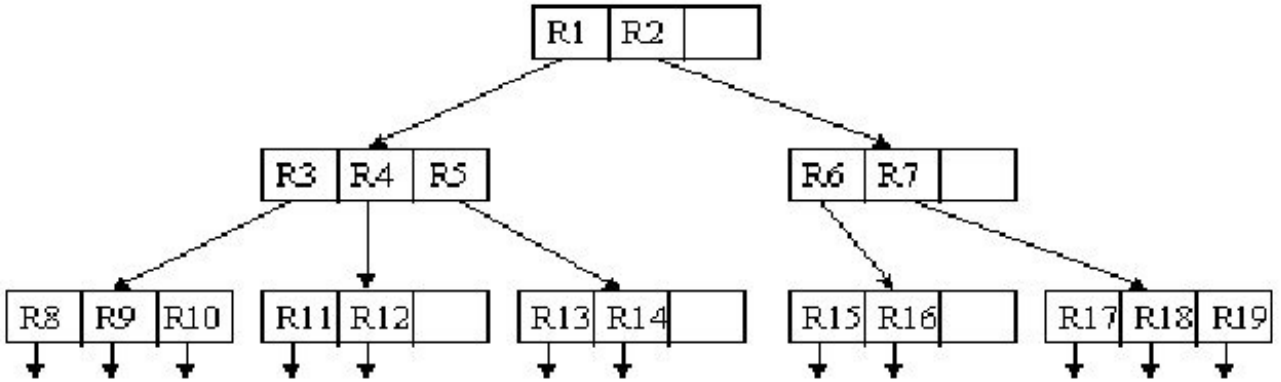
DBSCAN (cont.)

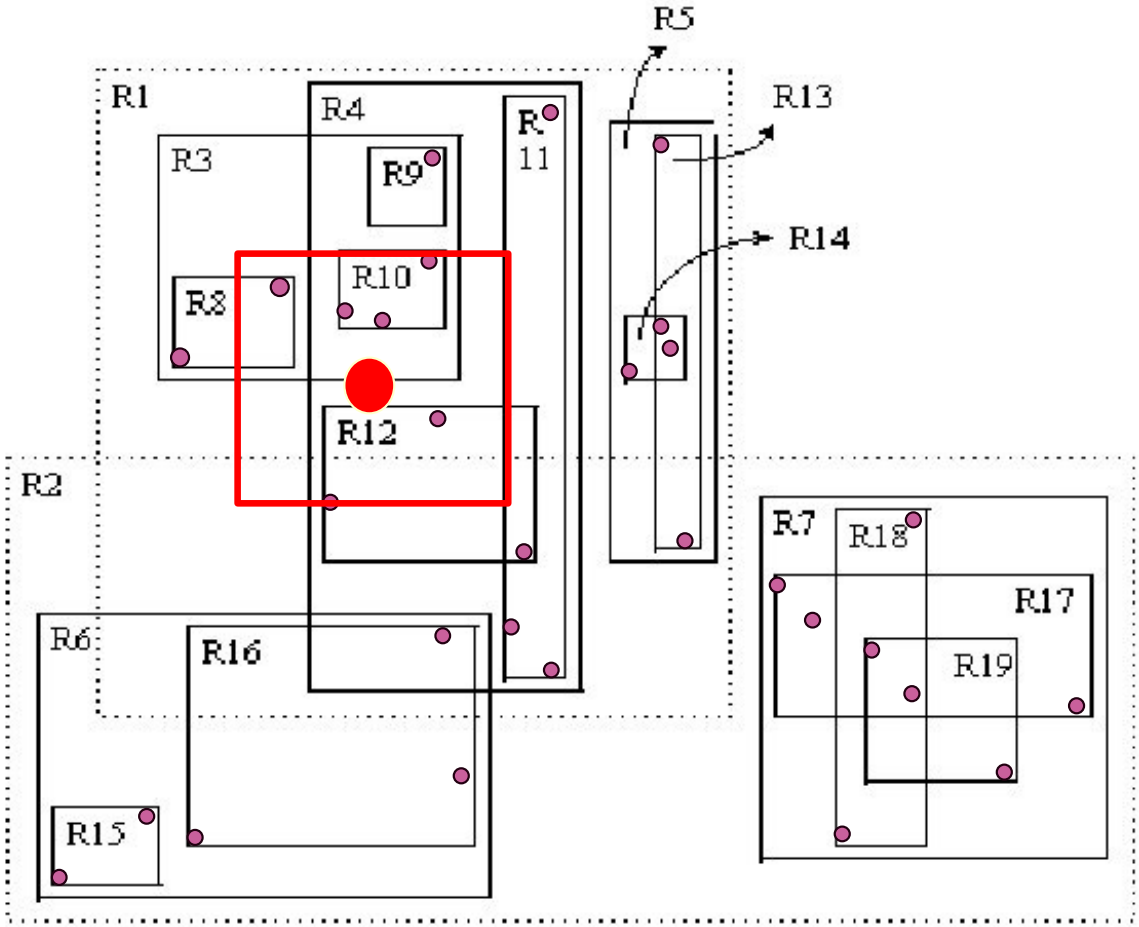
- Discovers clusters of arbitrary shape in spatial databases with noise
- If a spatial index (e.g. R-Tree) is used, O(nlogn)

  Otherwise, $O(n^2)$
- Deterministic except border points (sensitive to input order)

mentioned in database systems

R-Tree for Range Query (SIGMOD 1984)





Model-based Clustering

Model-based Clustering

- Assumption: data are generated by a mathematical model.
- attempt to optimize the fit between data and mathematical model
- Approaches
  - statistical approach: EM (Expectation Maximization)
  - neural network approach: SOM(Self-Organizing feature Map)

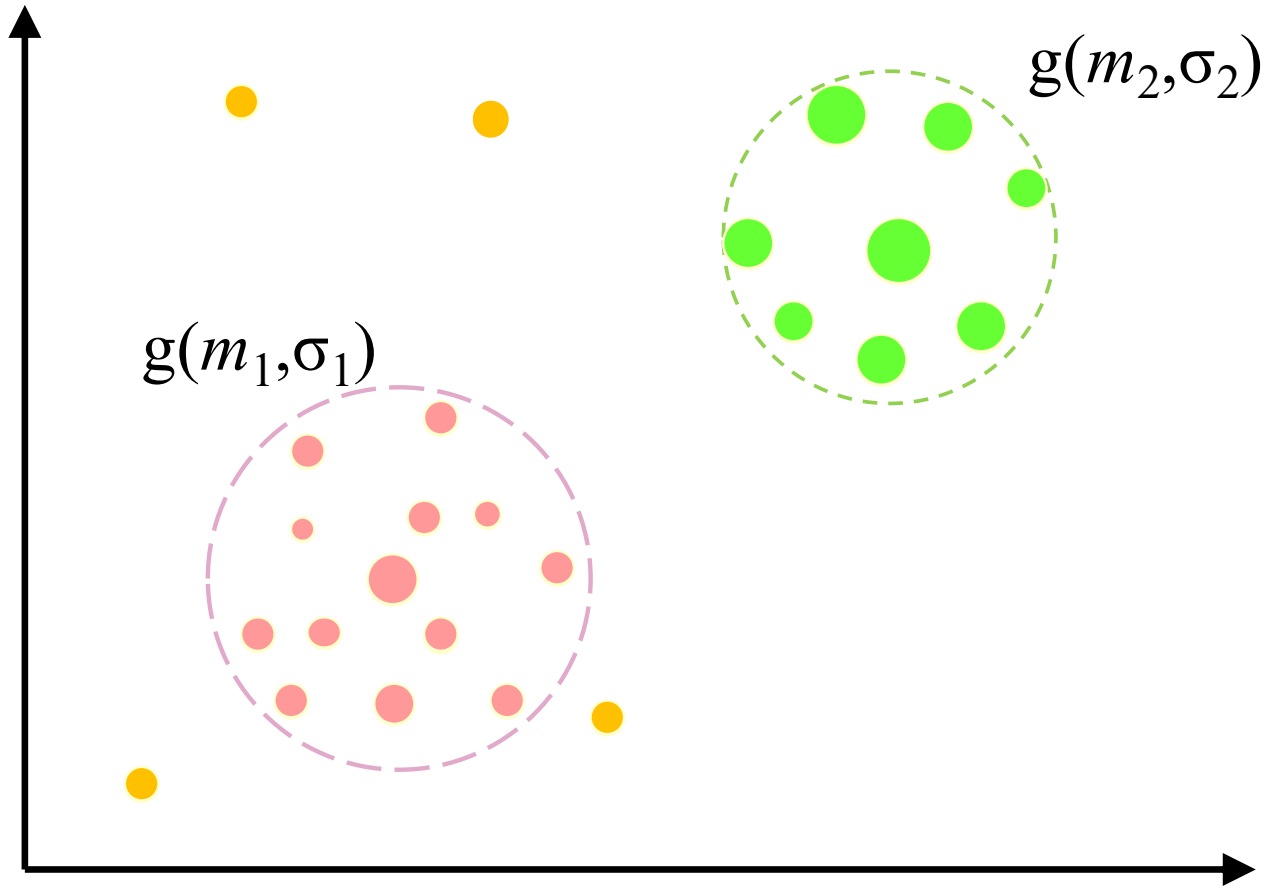
Rationale of Expectation Maximization

- Each cluster can be represented mathematically by a parametric probability distribution
- The entire data is a mixture of component distributions (of the same type).

  e.g. entire data is a mixture of 2 component distributions

  $g(m_1,\sigma_1)$ & $g(m_2,\sigma_2)$ * Gaussian distribution
- Clustering
  - cluster data using a finite mixture density model of *k* probability distributions (*k*: #(clusters))
  - estimate the parameters of the probability distribution so as to best fit the data.

Rationale of Expectation Maximization (cont.)



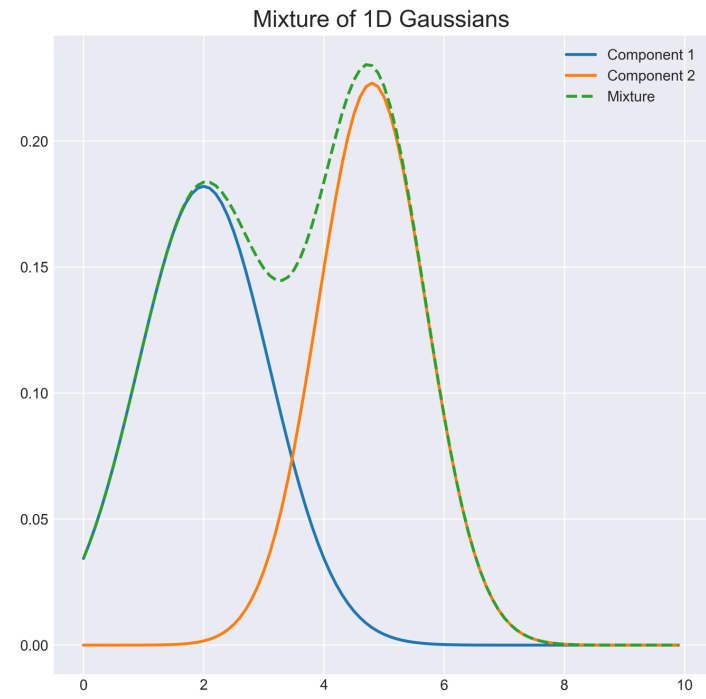
The *K*-Means Clustering Method

- Given *k*, the *k-means* algorithm:
  - Partition objects into *k* nonempty subsets
  - Compute mean as the centroids of the clusters of the current partition
  - Relocate each object to the nearest cluster
  - Go back to Step 2, stop when no more new relocation

Expectation Maximization

- EM (Gaussian Mixture Model, GMM)
  - Assume data is generated from a mixture of K Gaussian models

$$P(x) = \sum_{k=1}^{K} P(C_k) G(x|m_k, \sigma_k)$$

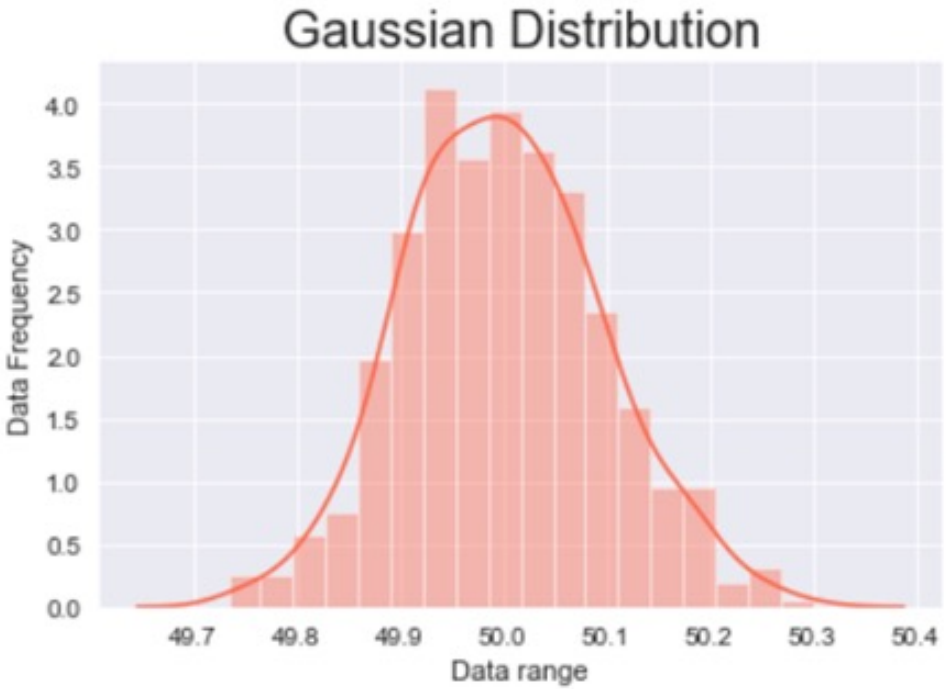


Expectation Maximization (cont.)

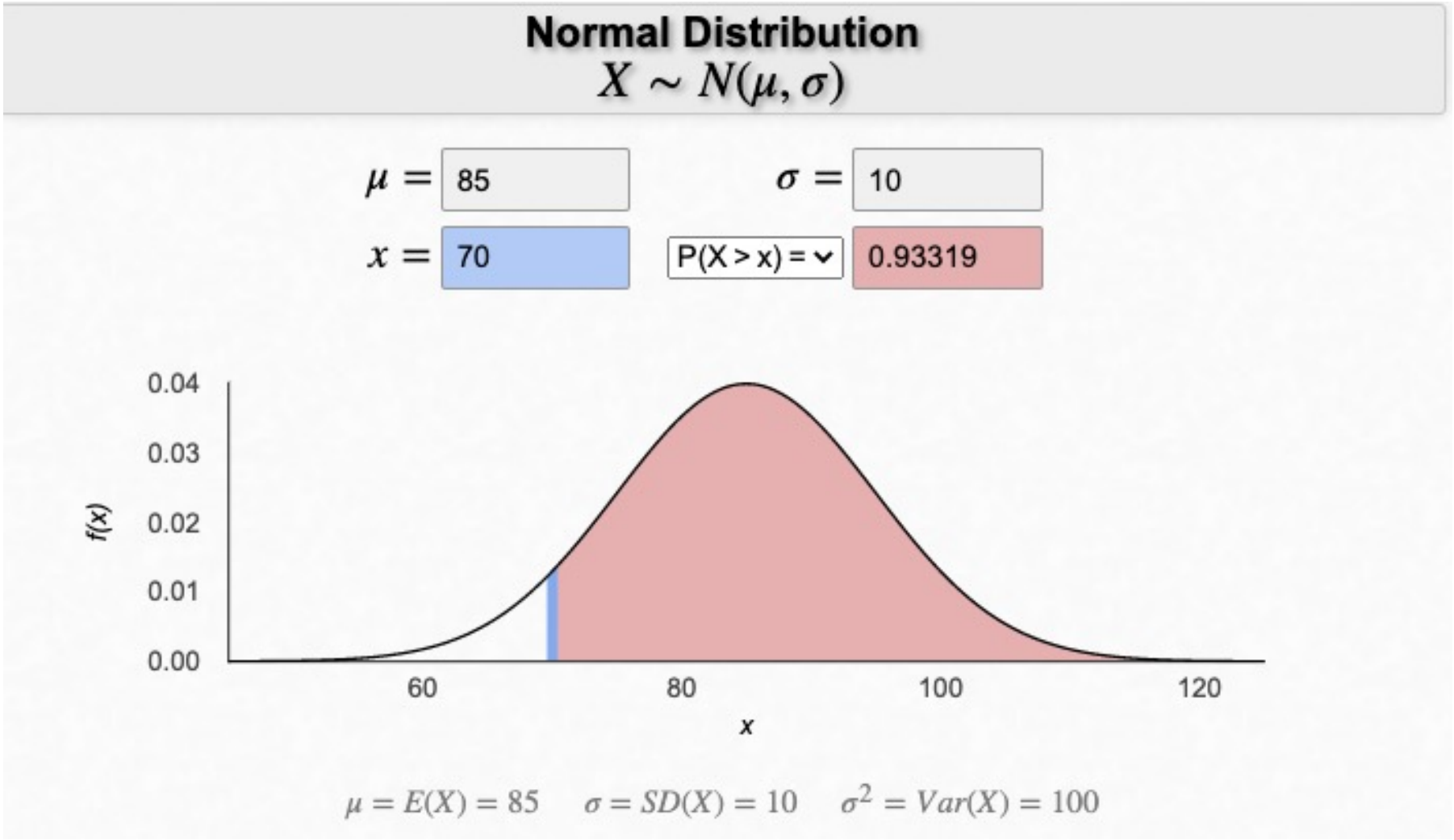
- Gaussian Distribution (Normal distribution)
  - Bell curve
  - Probability density function

$$G(x|m,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-m}{\sigma})^2}$$

m: mean, σ: standard deviation



Expectation Maximization (cont.)



Expectation Maximization (cont.)

- EM (Gaussian Mixture Model, GMM)
  - A popular iterative refinement algorithm
  - Assume data is generated from a mixture of *K* Gaussian models

$$P(x) = \sum_{k=1}^{K} P(C_k) G(x|m_k, \sigma_k)$$

  - An extension to k-means
    - Assign an object to the cluster with which it is most similar, based on the cluster mean
    - Instead of assigning each object to a dedicated cluster, EM assigns each object to a cluster according to a weight representing the probability of membership
    - New means are computed based on weighted measures

Expectation Maximization (cont.)

- A popular iterative refinement algorithm
  1. Make in initial guess of parameter vector: randomly select *k* object to represent the cluster means, as well as making guesses for the additional parameters
  2. Iteratively refine the clusters based on two steps
     - Expectation step: assign each data point x_i to cluster C_k with the following probability based on Bayes rule

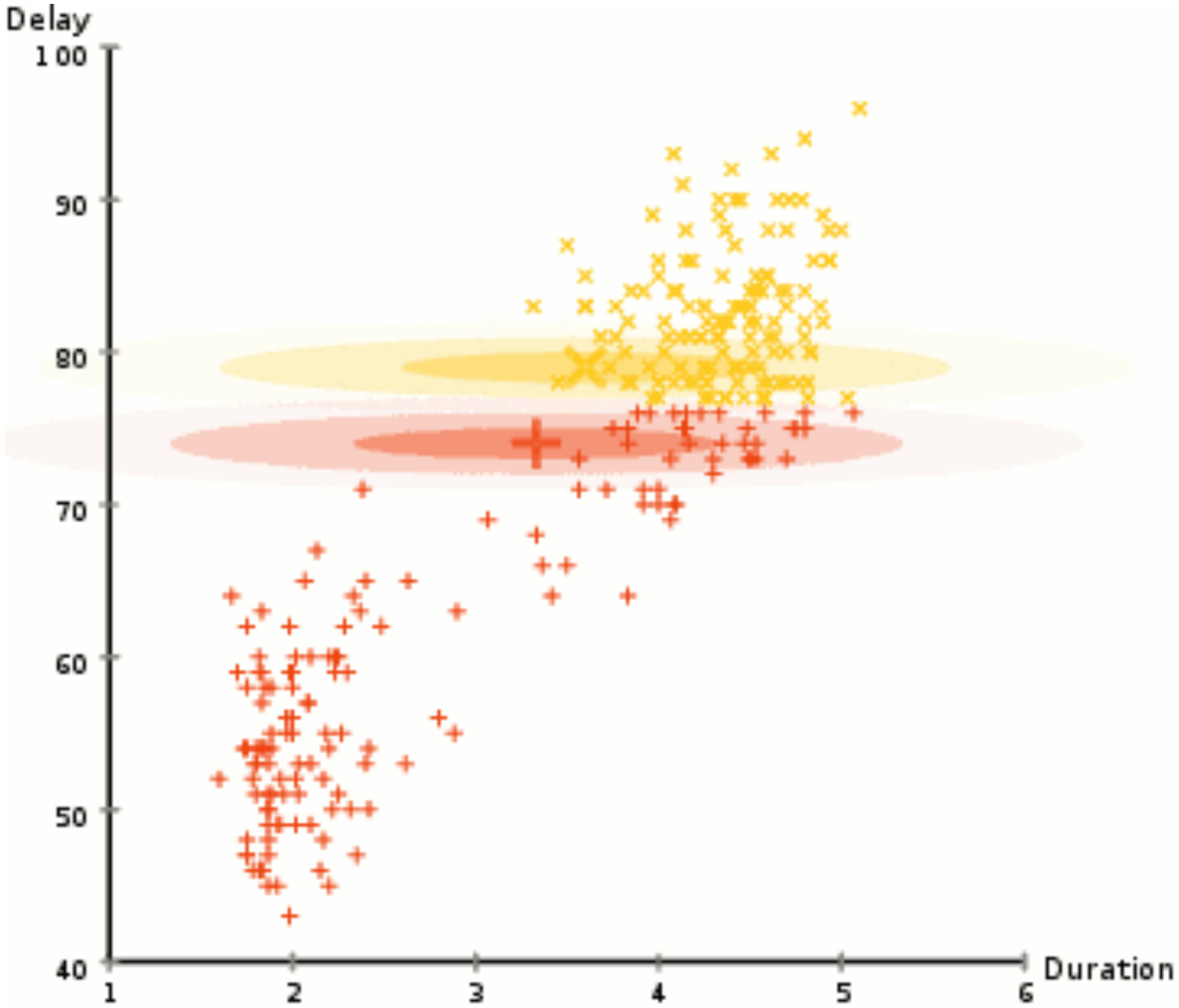
$$P(x_i \in C_k) = p(C_k \mid x_i) = p(C_k)p(x_i \mid C_k) \Big/ p(x_i)$$

     - Maximization step: re-estimation of model parameters.

e.g.

$$m_k = \frac{\sum_{i=1}^{n} x_i \times p(C_k|x_i)}{\sum_{i=1}^{n} p(C_k|x_i)}$$

GMM



Expectation Maximization: Example

– To cluster 20000 points, generated by $G(-4, 2)$ & $G(4, 2)$, into 2 clusters, assume

(1) standard deviation of both distributions is 2 ($\sigma_1 = \sigma_2 = 2$)

(2) points were generated with equal probability from both distributions

1. Make in initial guess of parameter vector: $m_1 = -2$, $m_2 = 3$

$$C_1 = G(m_1, \sigma_1) = G(-2, 2),\ C_2 = G(m_2, \sigma_2) = G(3, 2)$$

2. Iteratively refine the clusters based on two steps

– Expectation step: assign each data point x_i to cluster C_k

$$P(x_i \in C_1) = p(C_1 \mid x_i) = \frac{p(C_1)p(x_i \mid C_1)}{p(x_i)} = \frac{0.5p(x_i \mid C_1)}{0.5p(x_i \mid C_1) + 0.5p(x_i \mid C_2)}$$

e.g. for $x_i = 0$,

$$p(x_i \mid C_1) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_i - m_1)^2}{2\sigma_1^2}} = \frac{1}{\sqrt{2\pi}2} e^{-\frac{(0-(-2))^2}{2\times 2^2}} = 0.12$$

Expectation Maximization: Example (cont.)

2. Iteratively refine the clusters based on two steps
   - Expectation step: assign each data point x_i to cluster C_k

e.g. for x_i=0,

$$P(x_i \in C_1) = \frac{p(C_1)p(x_i \mid C_1)}{p(x_i)} = \frac{0.5p(x_i \mid C_1)}{0.5p(x_i \mid C_1)+0.5p(x_i \mid C_2)} = \frac{0.12}{0.12+0.06} = 0.66$$

$$P(x_i \in C_2) = \frac{p(C_2)p(x_i \mid C_2)}{p(x_i)} = \frac{0.5p(x_i \mid C_2)}{0.5p(x_i \mid C_1)+0.5p(x_i \mid C_2)} = \frac{0.06}{0.12+0.06} = 0.33$$

   - Maximization step: compute new estimate for m_1, m_2

$$m_1 = \frac{\sum_{i=1}^{20000} x_i \times p(C_1|x_i)}{\sum_{i=1}^{20000} p(C_1|x_i)}$$

$$m_2 = \frac{\sum_{i=1}^{20000} x_i \times p(C_2|x_i)}{\sum_{i=1}^{20000} p(C_2|x_i)}$$

: weighted average of all pts, where the weights are the probabilities of each data point belonging to that cluster $p(C_k|x_i)$

Expectation Maximization (cont.)

- EM algorithm is simple & easy to implement
- In practice, it converges fast but may not reach the global optima.
- Computation complexity: O(*dnt*),

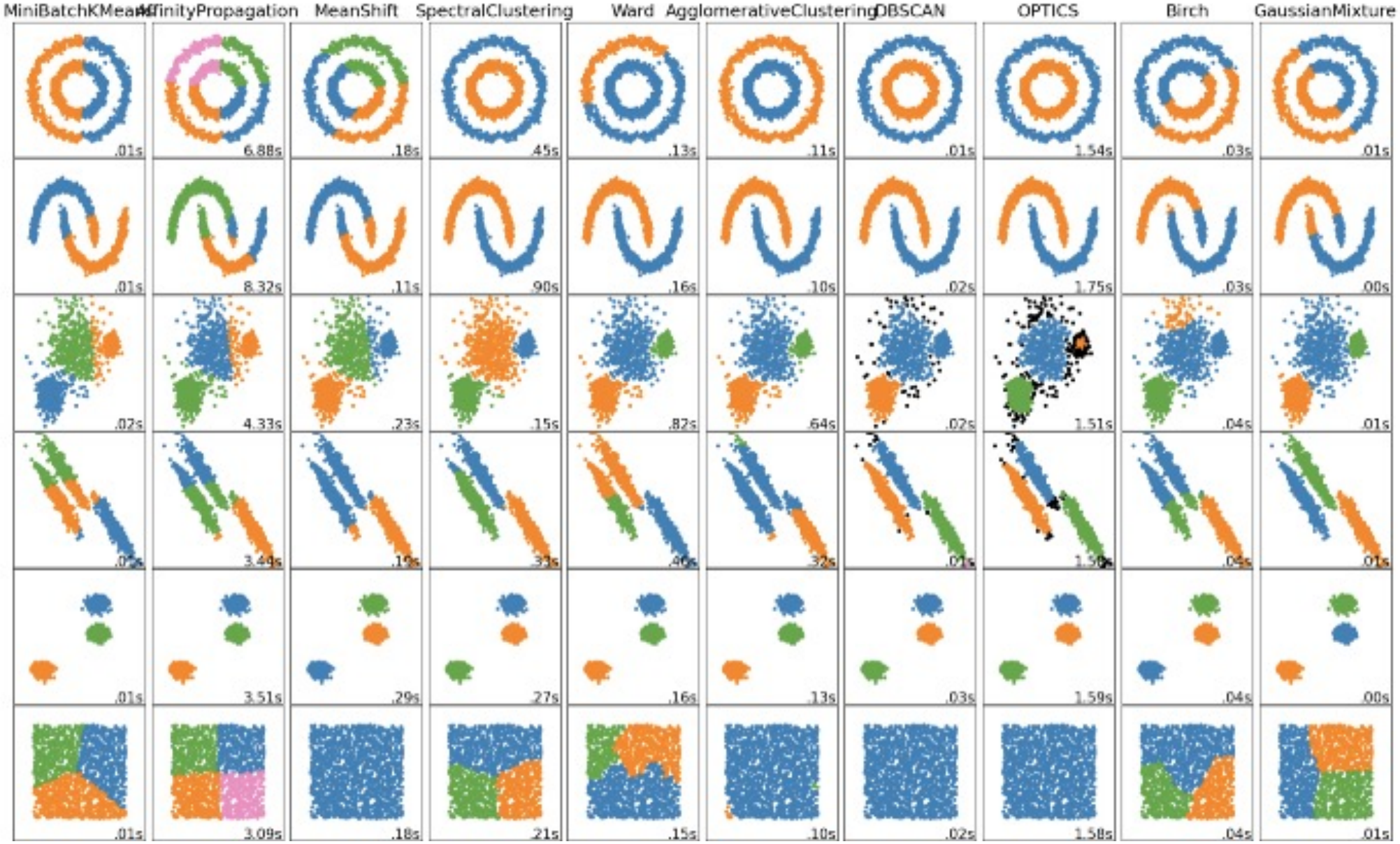
  d: #(input features), *n*:#(objects), *t*: #(iterations)
- More general than K-means soft clustering
- Objects are allowed to belong to more than one cluster
- A cluster is modeled as a statistics distribution
- Can find clusters of different sizes
- Not practical for models with large number of components

Conclusions

- Clustering: process of grouping a set of physical or abstract objects into classes of similar objects
- Good clustering (produce high quality clusters)
  - intra-cluster similarity is high
  - inter-cluster class similarity is low
- Quality factors
  - Similarity measure and its implementation
  - definition and representation of cluster chosen
  - clustering algorithm

Conclusions

- Approaches
  - Partition-based
    - K-means
    - K-Medoids: PAM
  - Hierarchical
    - Single Link, Complete Link, Average Link
    - BIRCH, Chameleon
  - Density-based
    - DBSCAN
  - Model-based
    - GMM


MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
OPTICS
Birch
GaussianMixture
.01s
6.88s
.18s
.45s
.13s
.11s
.01s
1.54s
.03s
.01s
.01s
8.32s
.11s
.90s
.16s
.10s
.02s
1.75s
.03s
.00s
.02s
4.33s
.23s
.15s
.82s
.64s
.02s
1.51s
.04s
.01s
.02s
3.44s
.19s
.33s
.46s
.32s
.01s
1.50s
.04s
.01s
.01s
3.51s
.29s
.27s
.16s
.13s
.03s
1.59s
.04s
.00s
.01s
3.09s
.18s
.21s
.15s
.10s
.02s
1.58s
.04s
.01s

MKShan


尹相志
22分鐘 ·
在某個AI相關審查會議...
廠商：為了解決標注問題，我們
後來改採用非監督式學習
評審：那你們原來用的是..？
廠商：k-means...