Knowledge Discovery and Data Mining

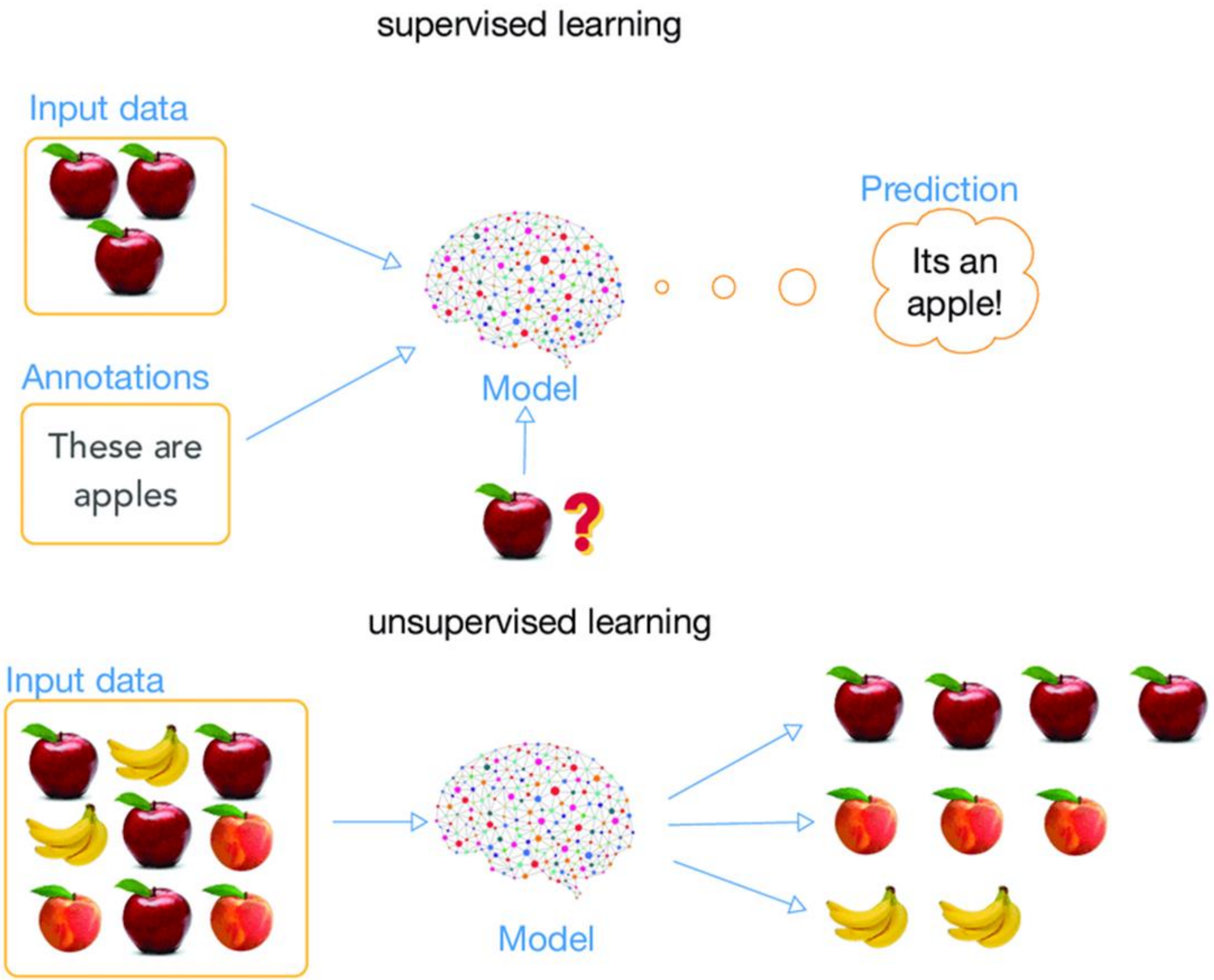
Class 4 Unsupervised Learning

Xuan Song
songx@sustech.edu.cn

Introduction

Unsupervised Learning

- In machine learning, the problem of **unsupervised learning** is that of trying to find hidden structure in unlabeled data.



Unsupervised Learning

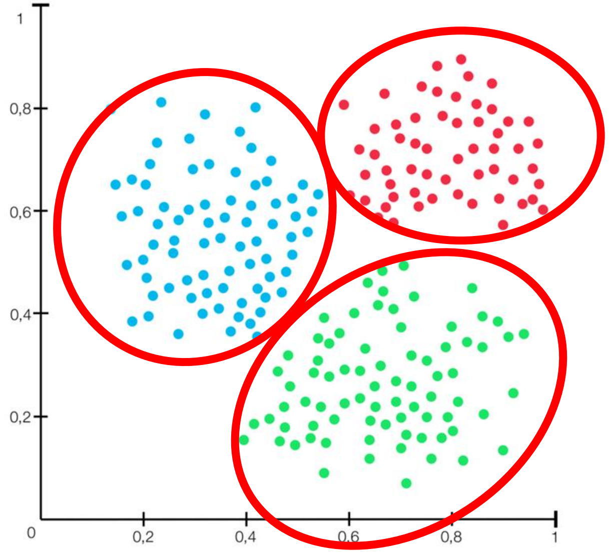
- Some of the most common algorithms used in unsupervised learning include:
  - (1) **Clustering** (e.g., k-means, mixture models, hierarchical clustering)
  - (2) Anomaly detection
  - (3) Neural Networks
  - (4) Approaches for learning latent variable models
    - Expectation–maximization algorithm (EM)
    - Blind signal separation techniques (PCA, Non-negative matrix factorization, etc.)

Application: Video

Cluster Analysis

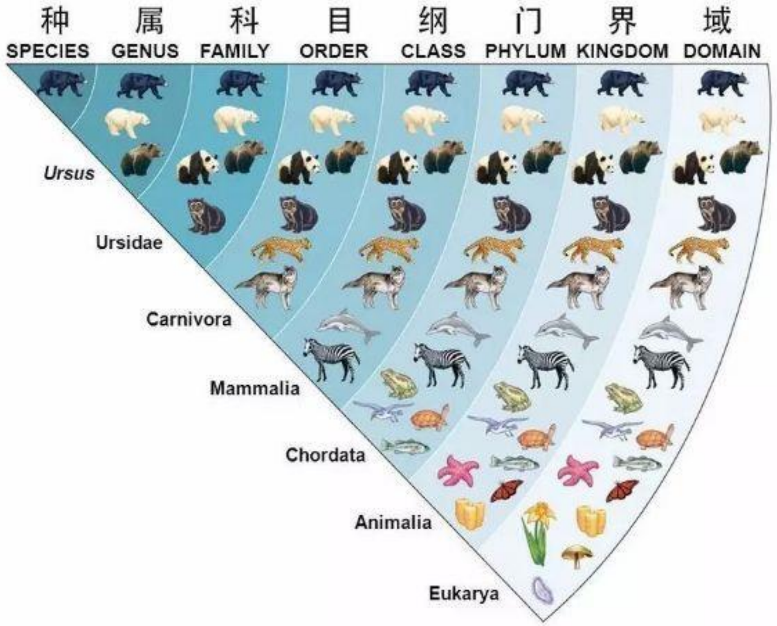
Cluster Analysis

- Cluster analysis groups data objects based only on information found in the data that describes the objects and their relationships.
- **Goal:** the objects within a group be similar (or related) to one another and different from (or unrelated to) the objects in other groups.

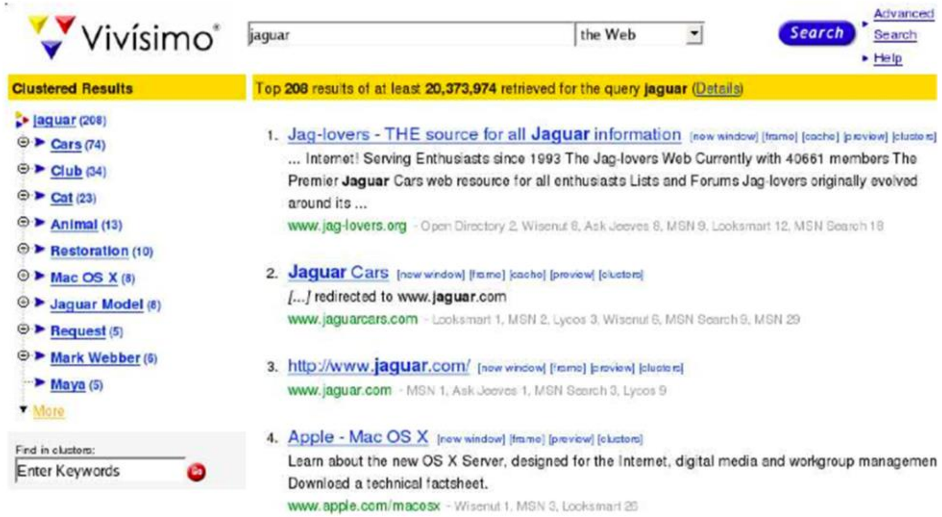


Application of Cluster Analysis

Market Segmentation



Assistance in deriving taxonomic criteria for plants and animals



Search result clustering

Notion of a Cluster can be Ambiguous

How many clusters?

Notion of a Cluster can be Ambiguous

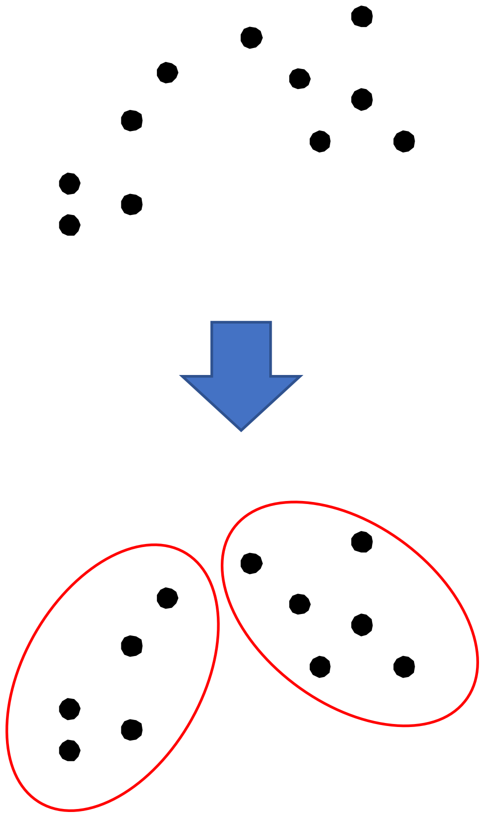
Six clusters

Two clusters

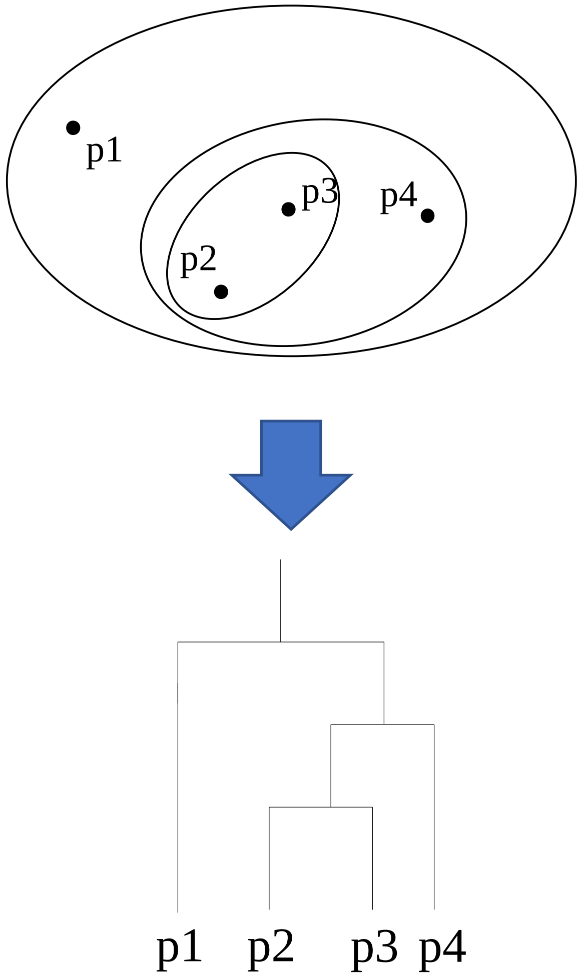
Four clusters

Clustering Methods

- (1) Partitional Methods

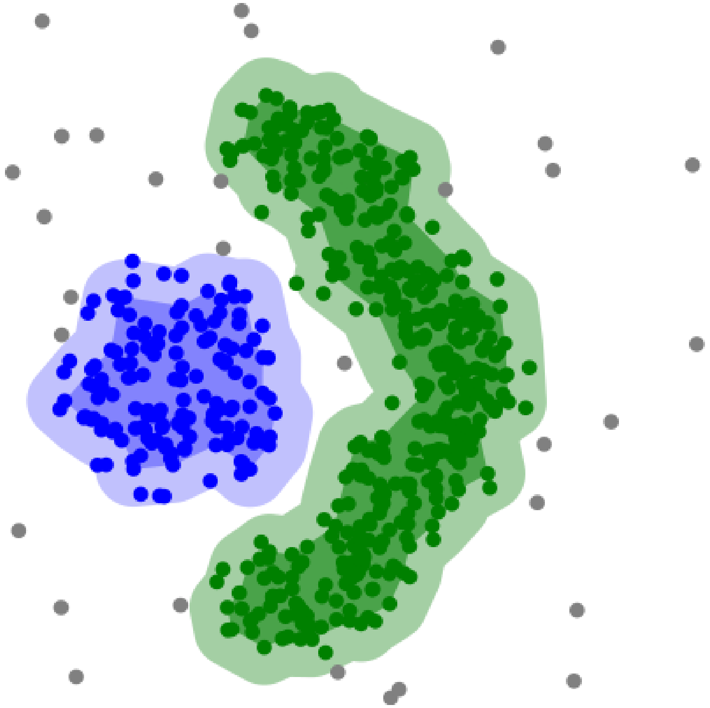


- (2) Hierarchical Methods

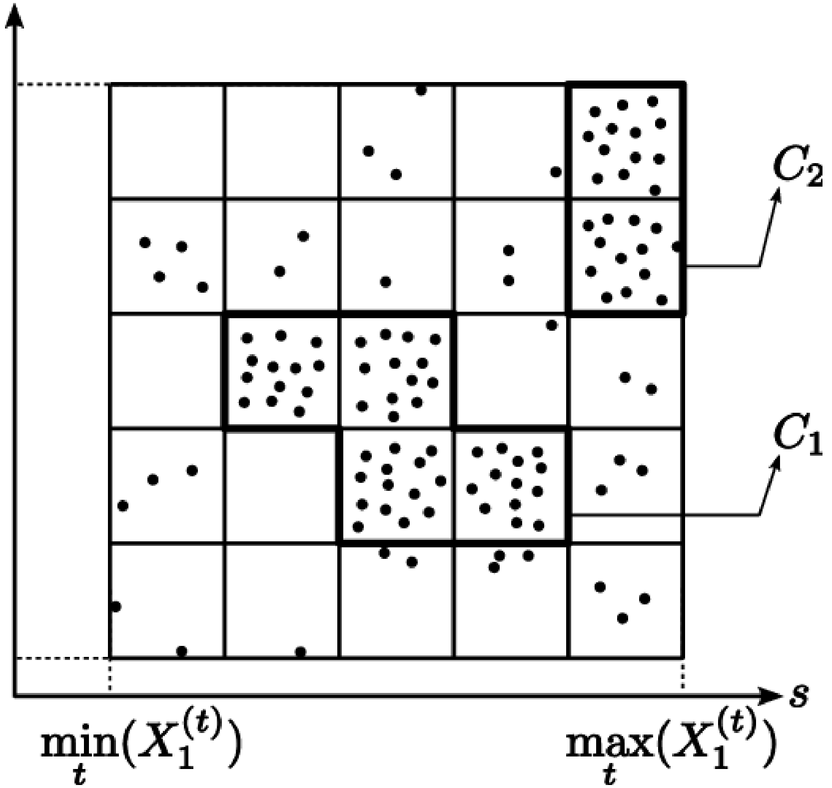


Clustering Methods

- (3) Density-based methods



- (4) Grid-based Methods

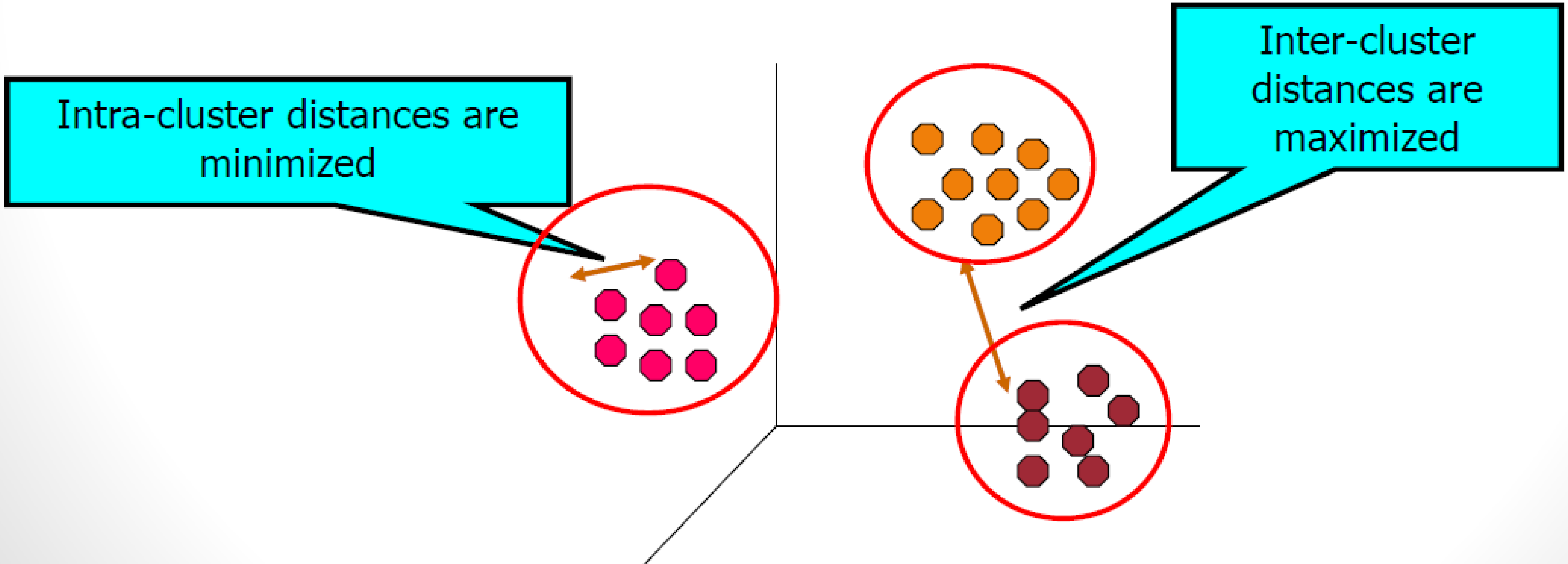


Clustering Algorithms

- K-means
- Hierarchical clustering
- Density-based clustering

Segmentation and Cluster Analysis

- Cluster is a group of similar objects (cases, points, observations, examples, members, customers, patients, locations, etc)
- Finding the groups of cases/observations/ objects in the population such that the objects are
  - Homogeneous within the group (high intra-class similarity)
  - Heterogeneous between the groups(low inter-class similarity)



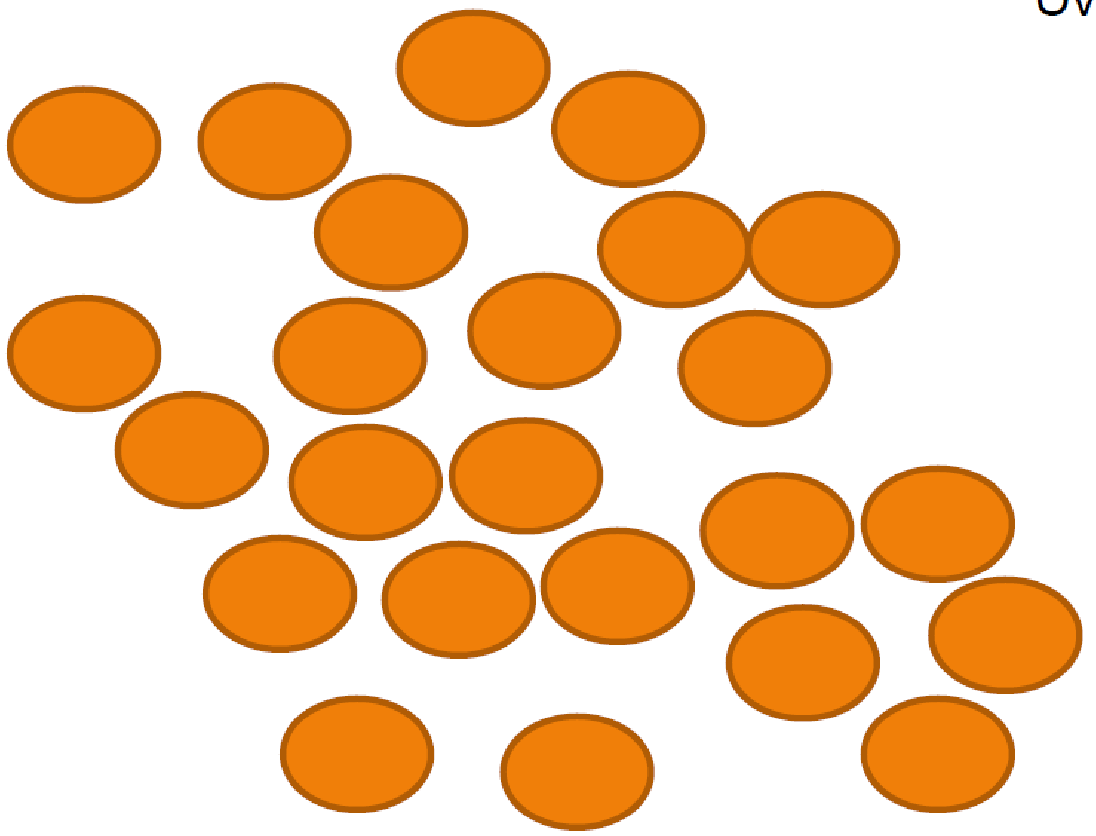
K-means

- Partitional clustering approach
- Number of clusters, K, must be specified
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- The basic algorithm is very simple

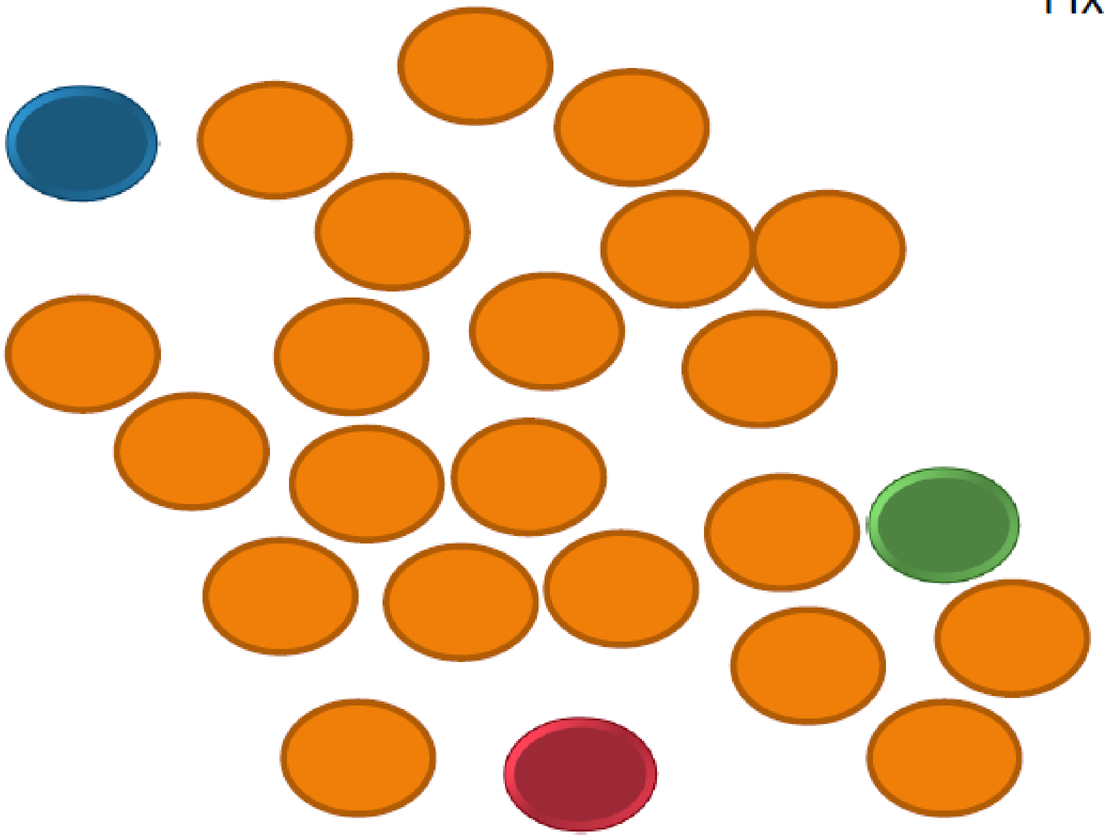
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

K-Means clustering

Overall population

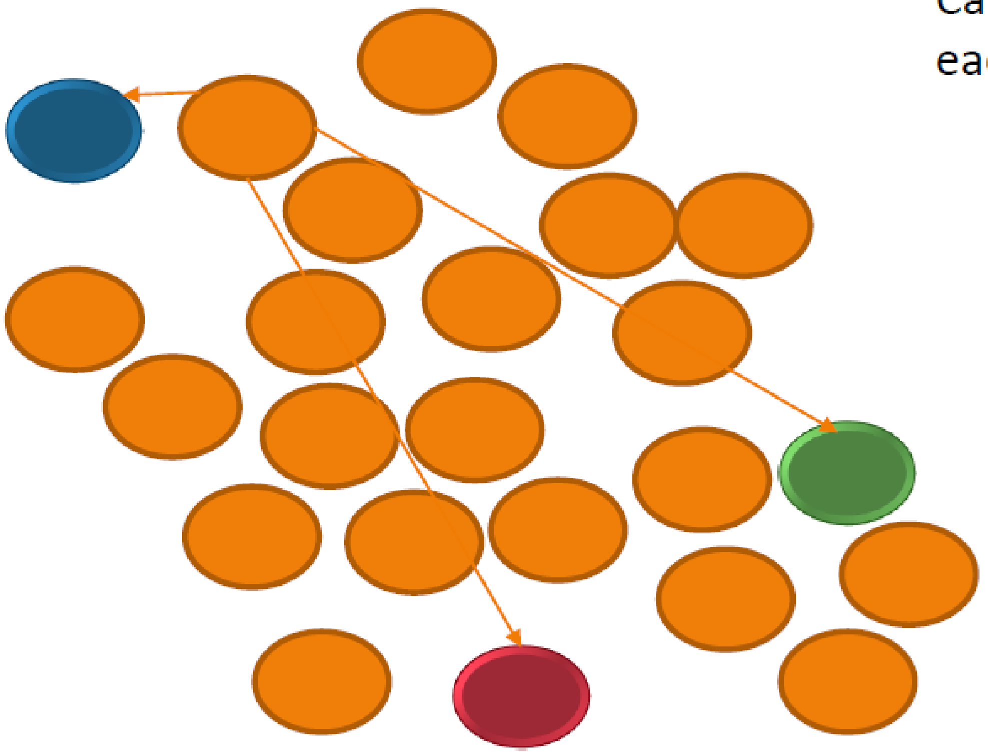


K-Means clustering



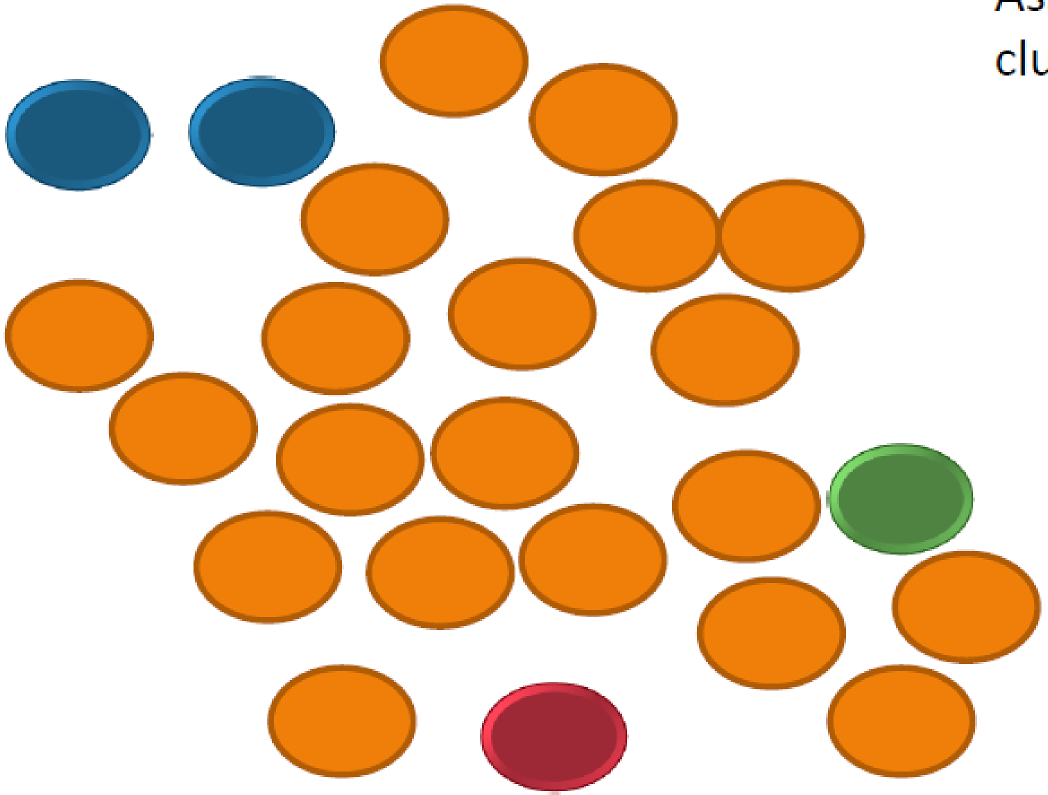
Fix the number of clusters

K-Means clustering



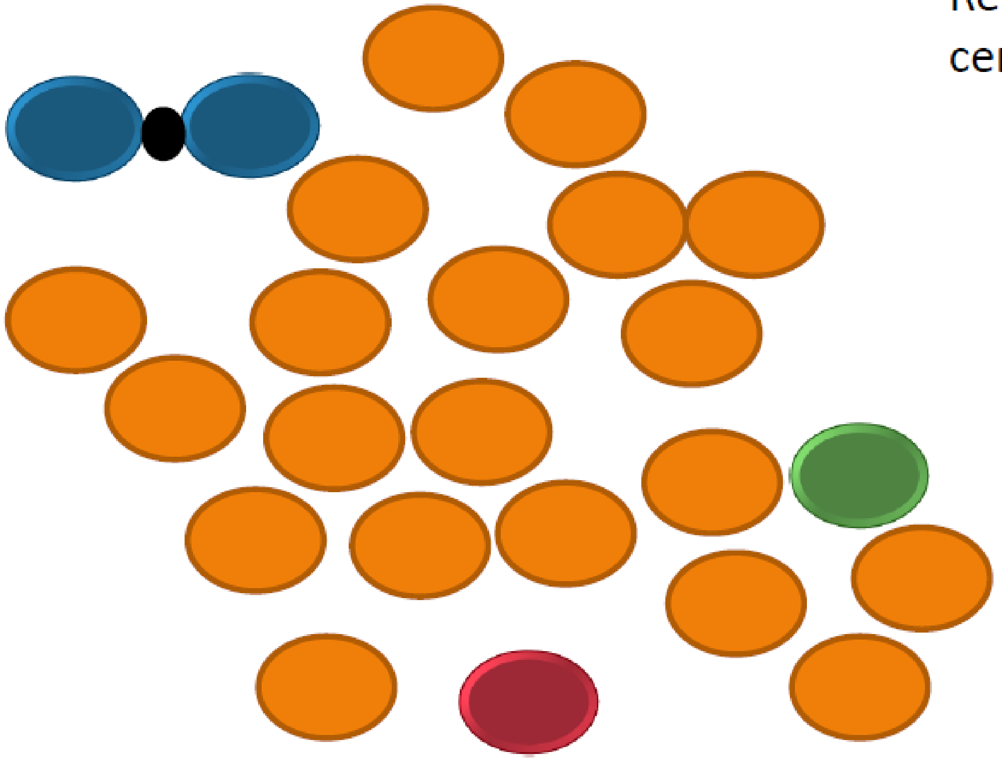
Calculate the distance of each case from all clusters

K-Means clustering



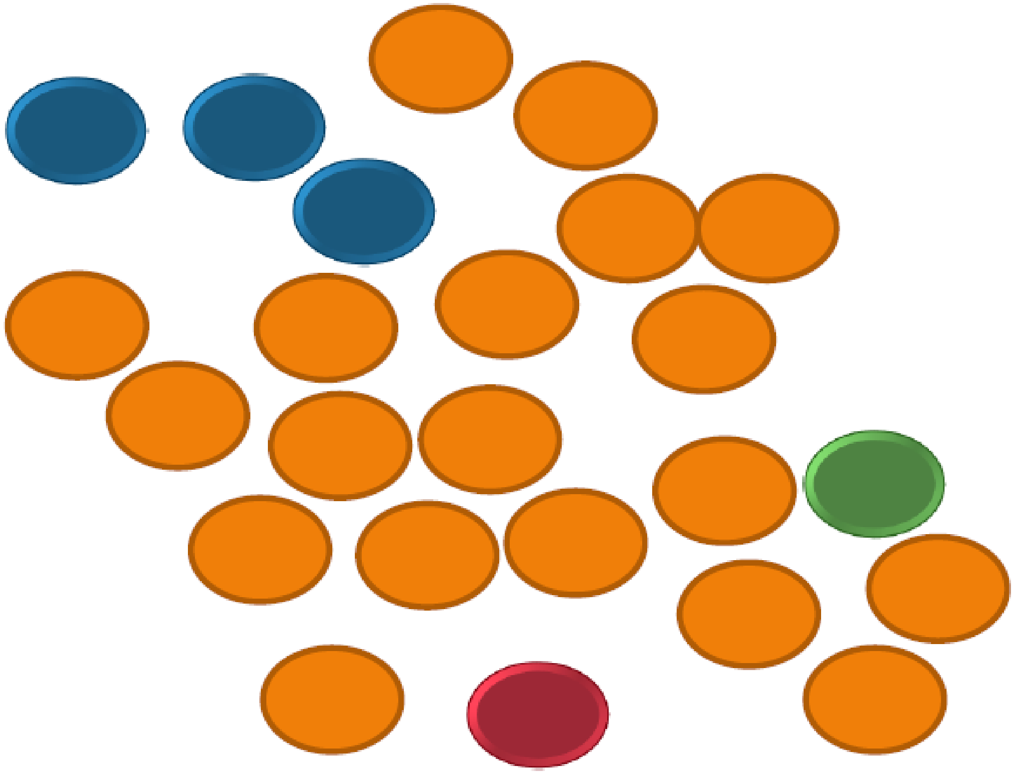
Assign each case to nearest cluster

K-Means clustering

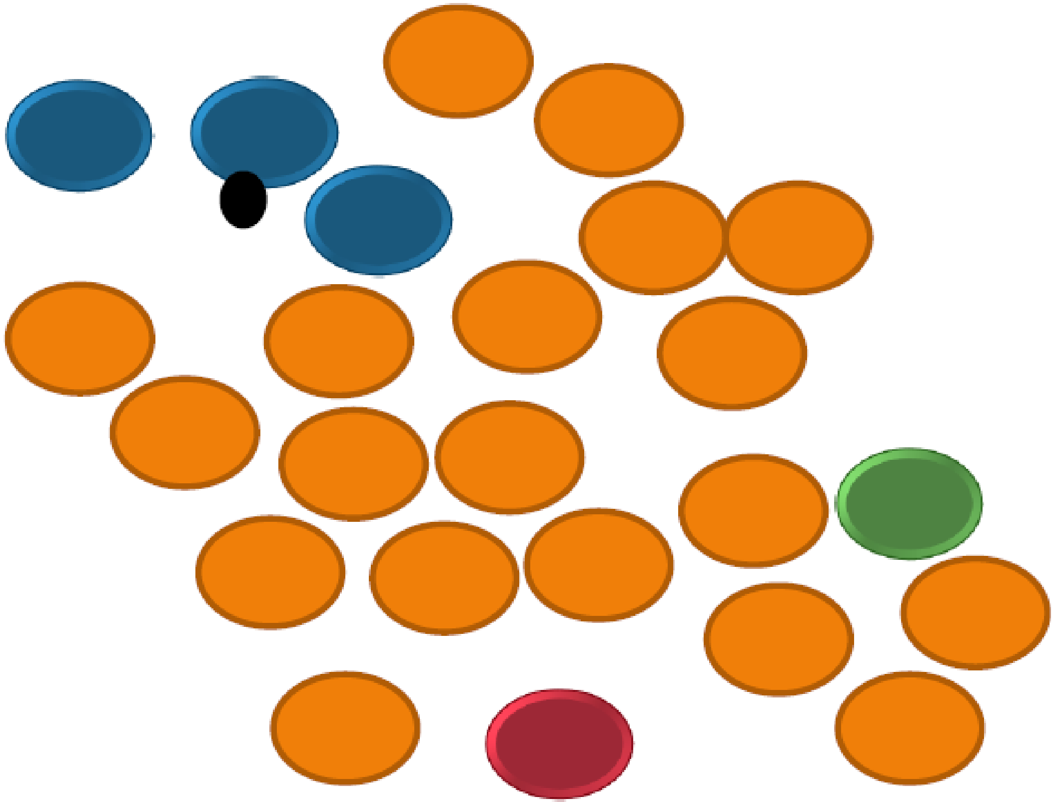


Re calculate the cluster centers

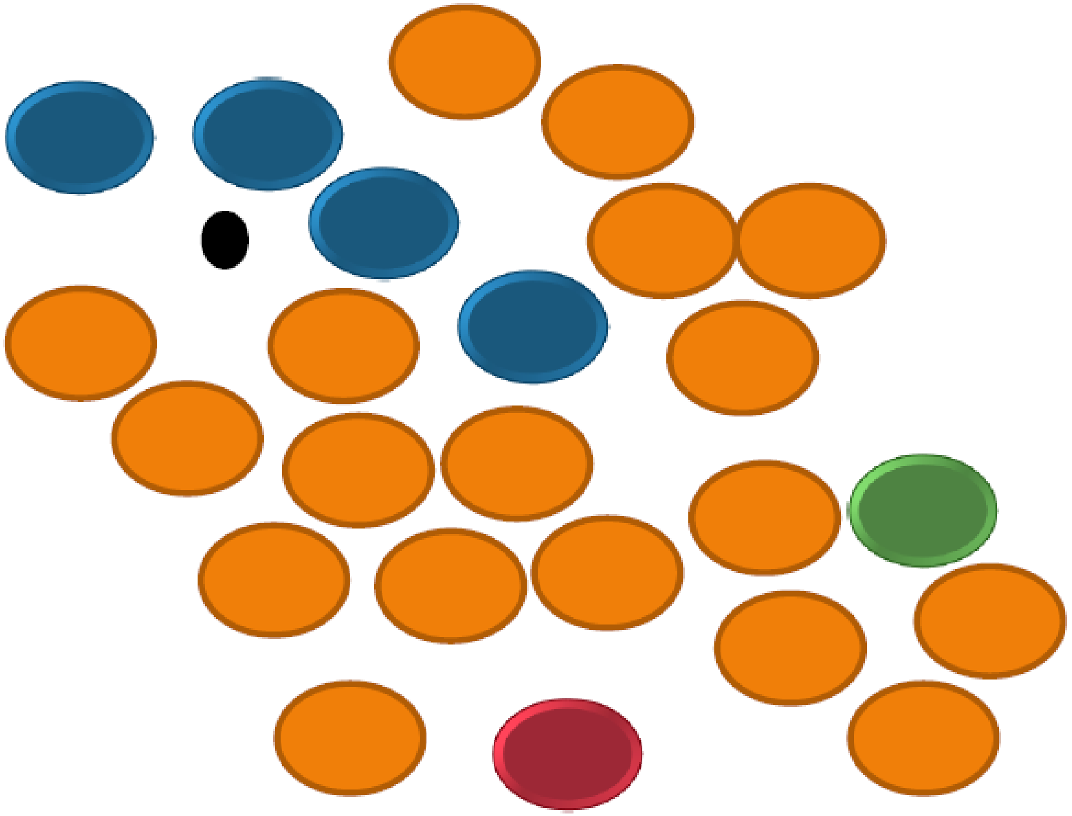
K-Means clustering



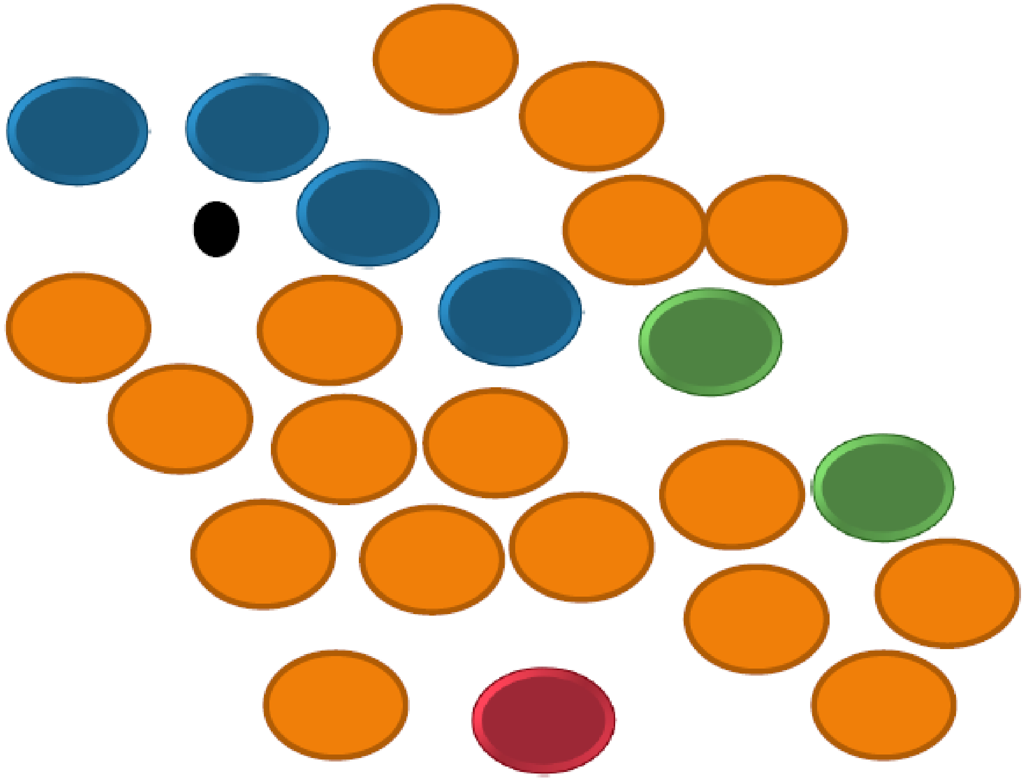
K-Means clustering



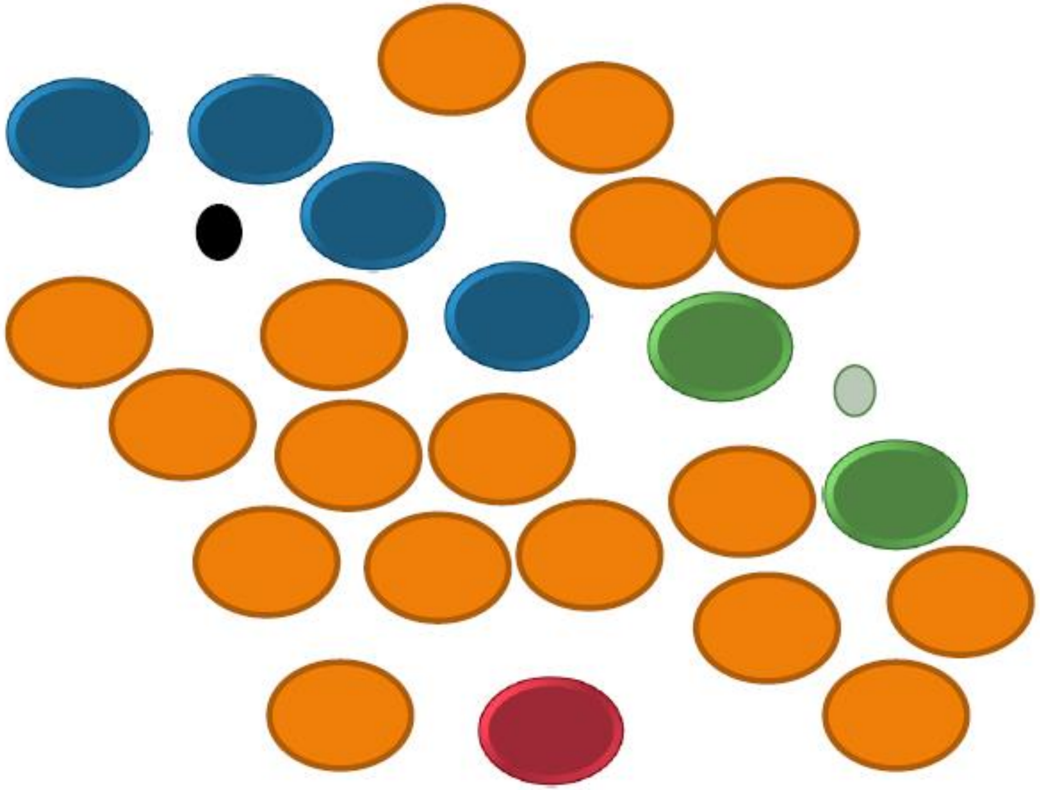
K-Means clustering



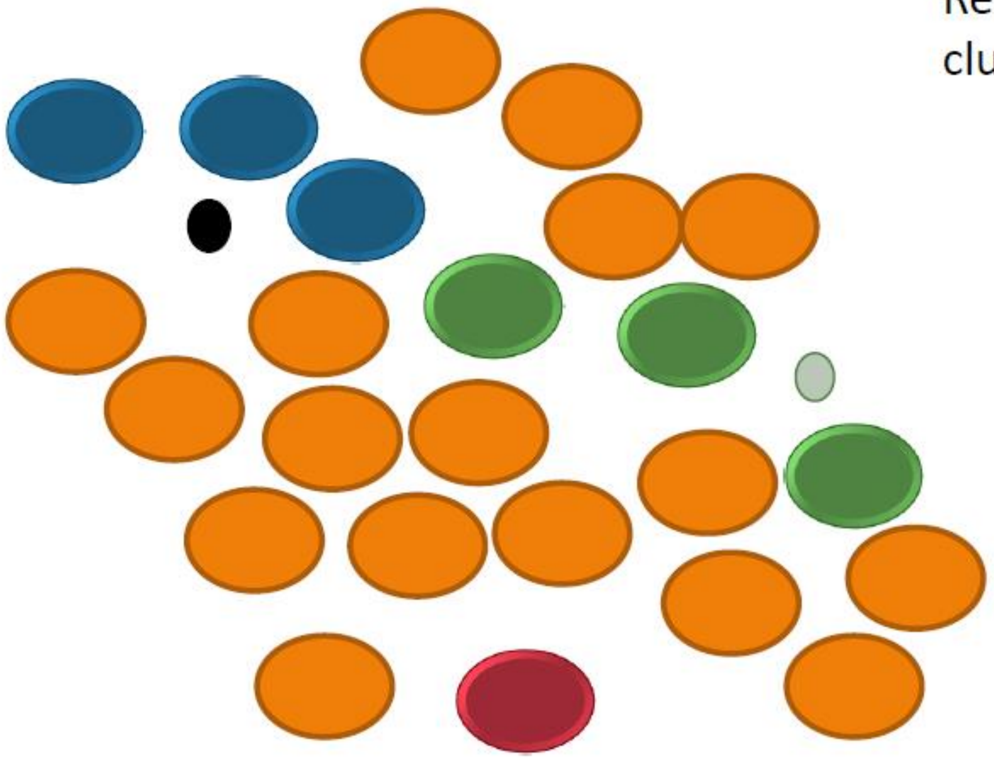
K-Means clustering



K-Means clustering

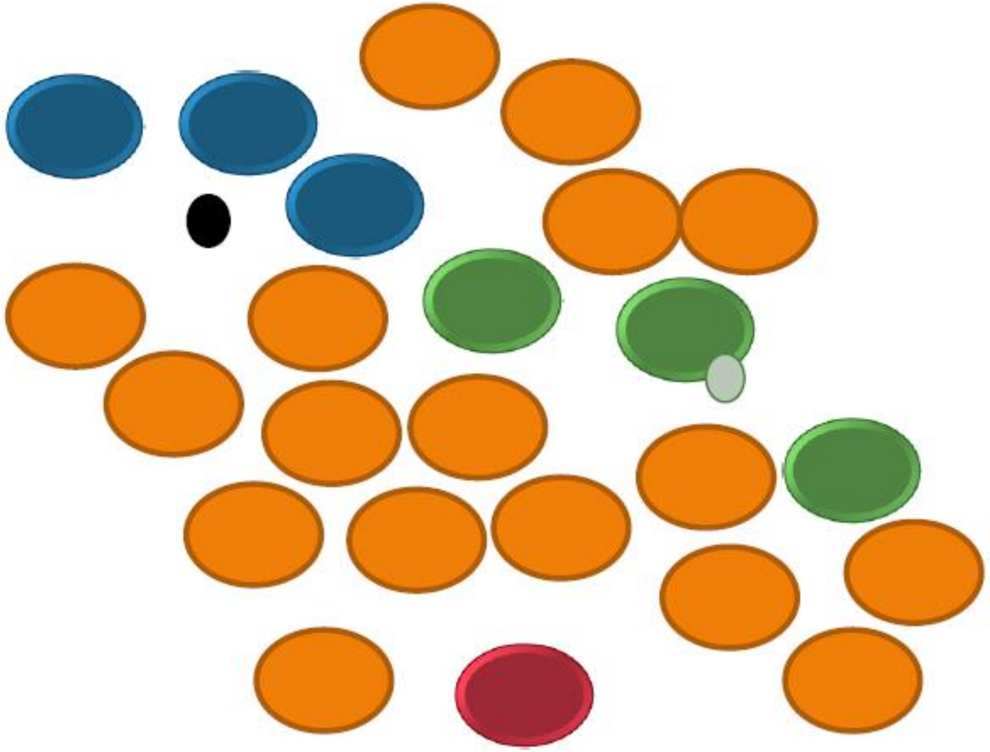


K-Means clustering

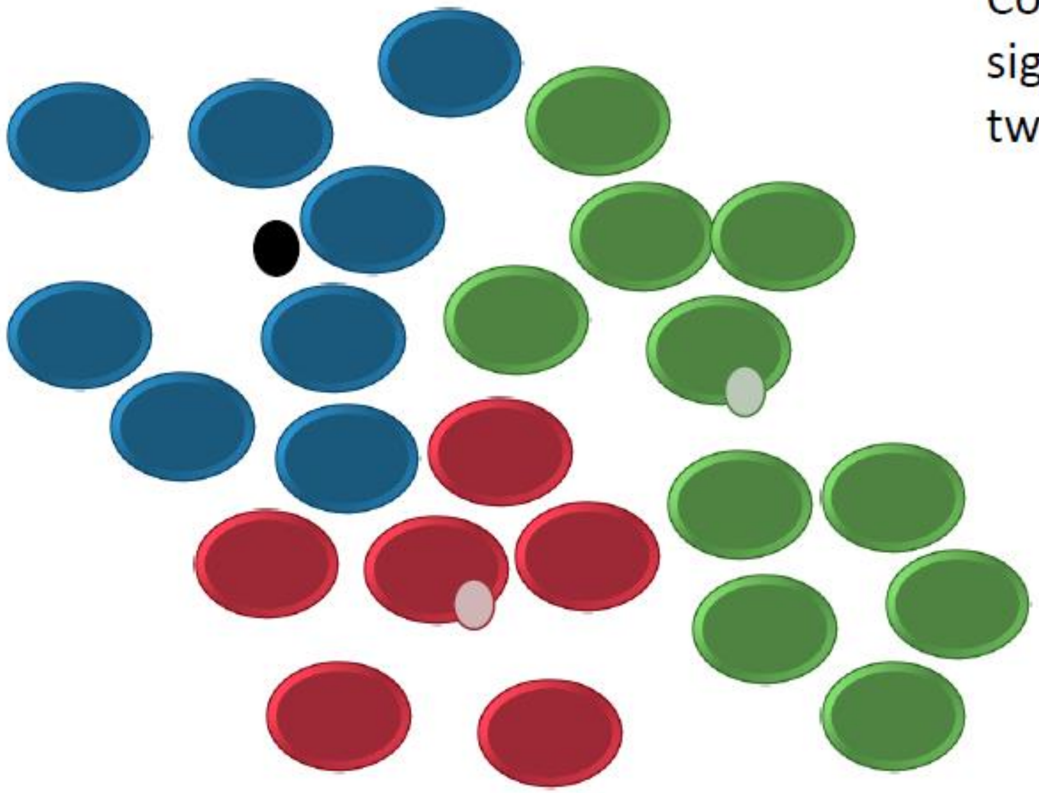


Reassign after changing the cluster centers

K-Means clustering

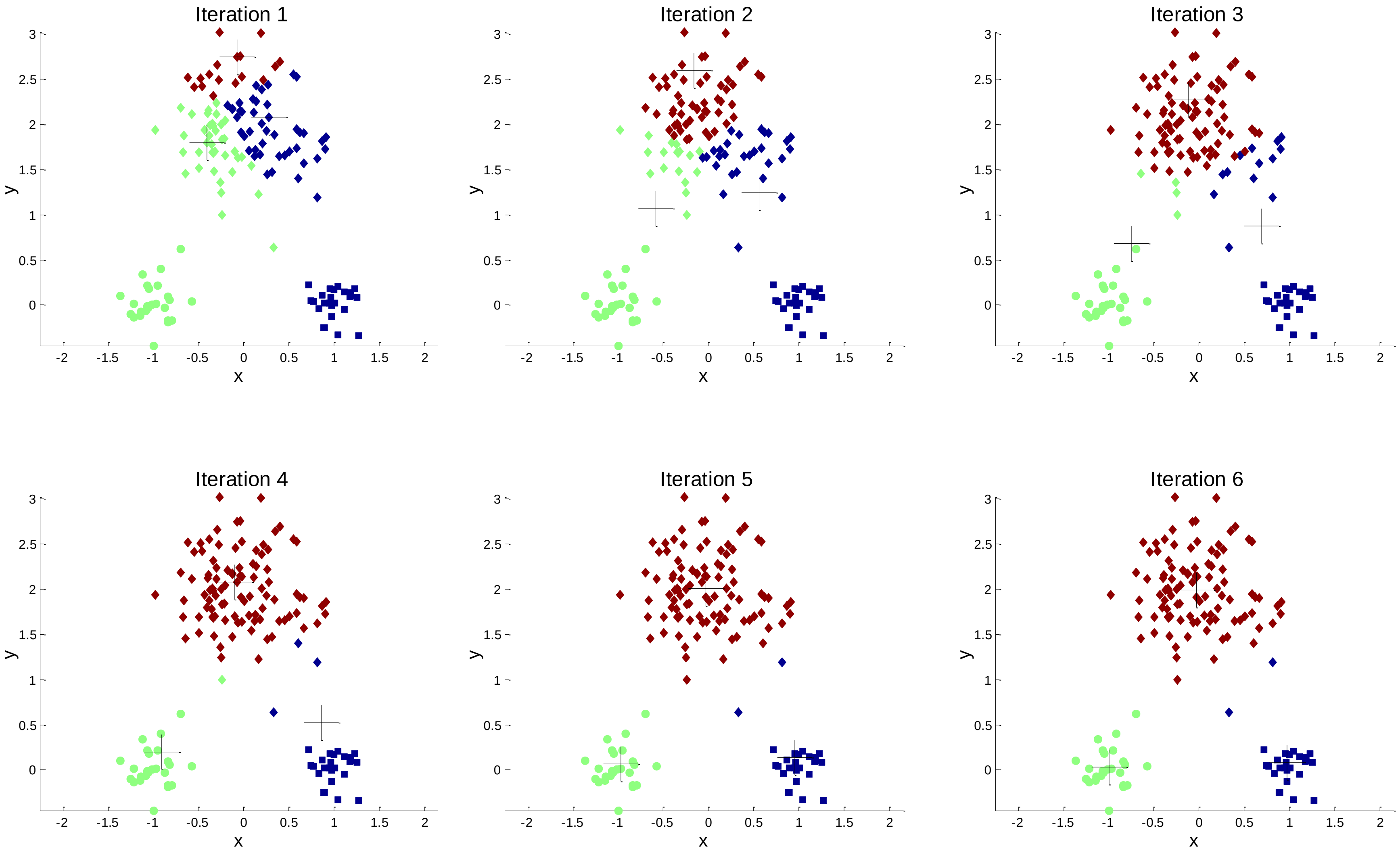


K-Means clustering



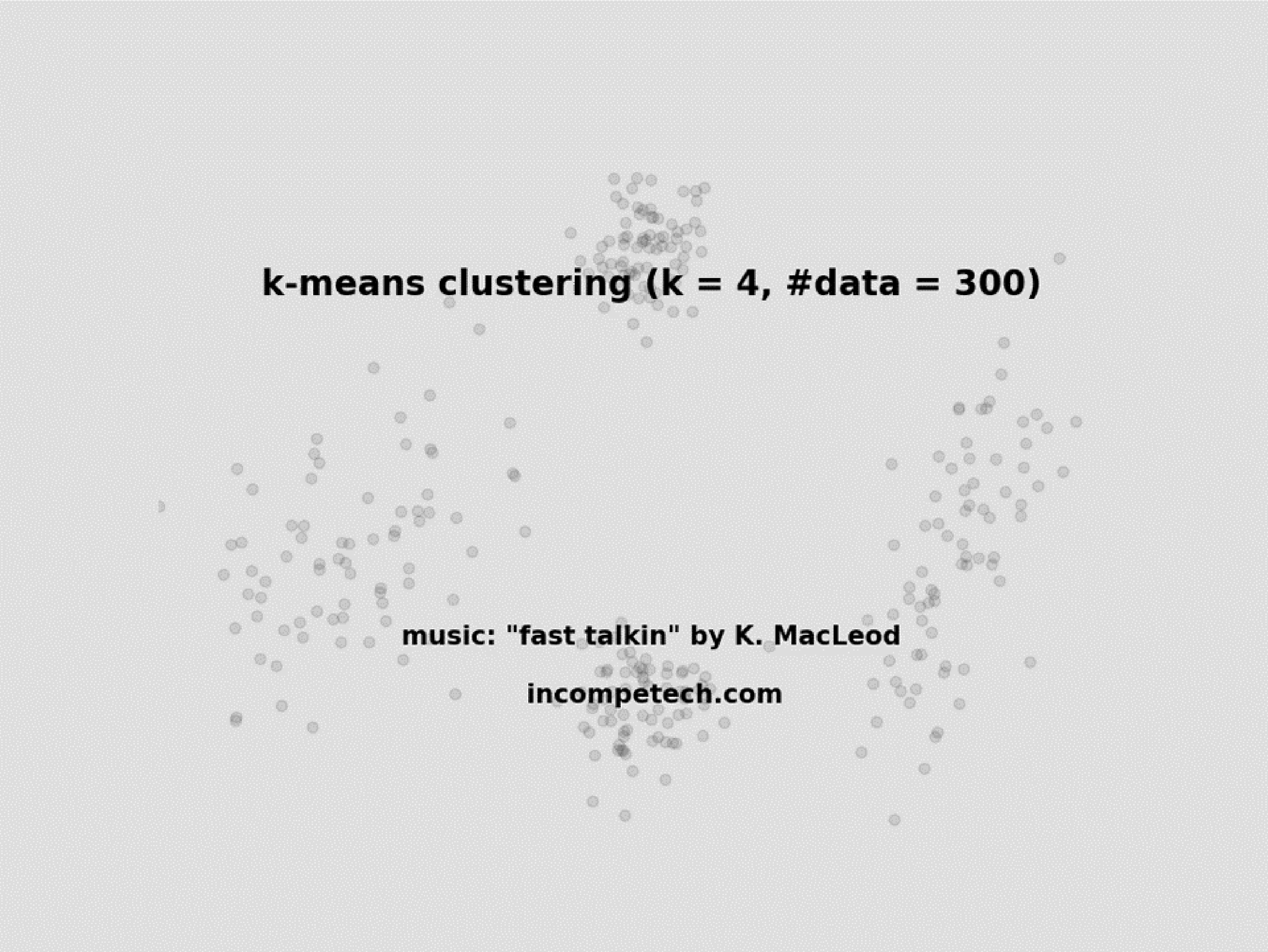
Continue till there is no significant change between two iterations

K-means



Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
x
y

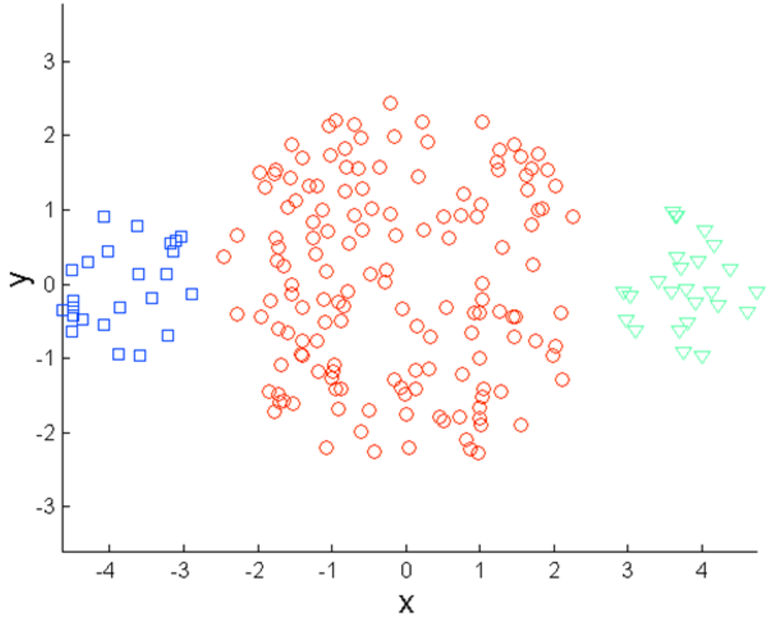
Video: K-means Clustering



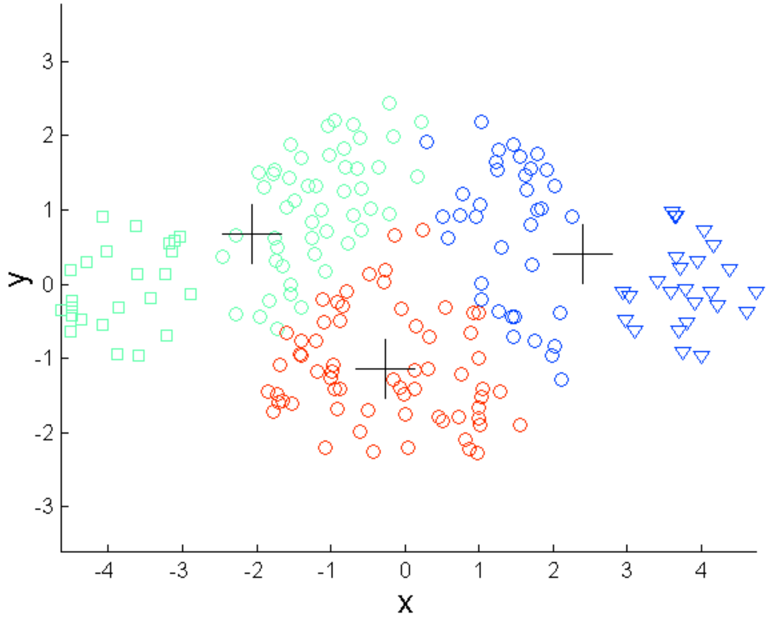
Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

Limitations of K-means: Differing Sizes

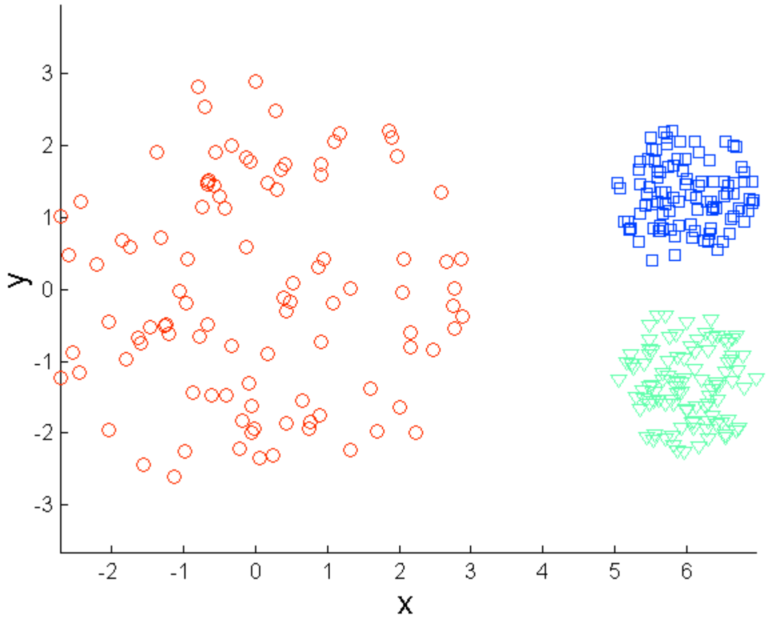


Original Points

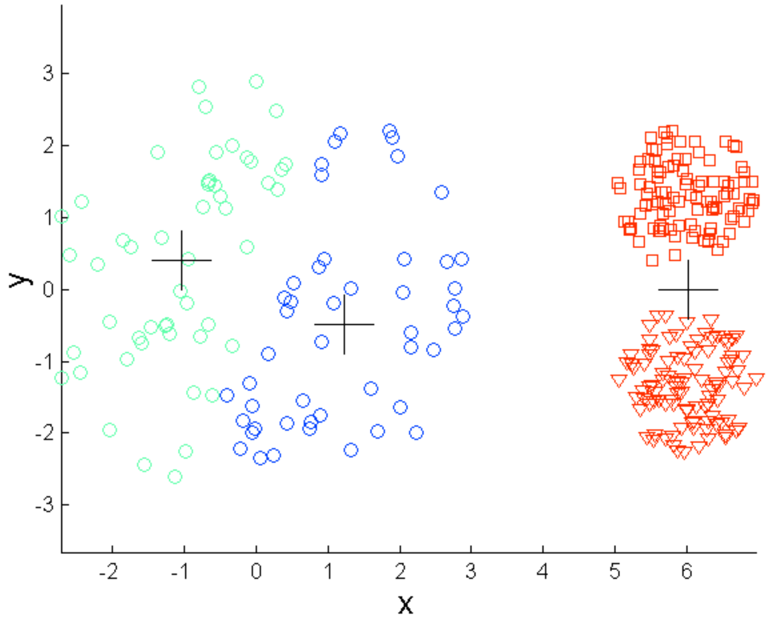


K-means (3 Clusters)

Limitations of K-means: Differing Density

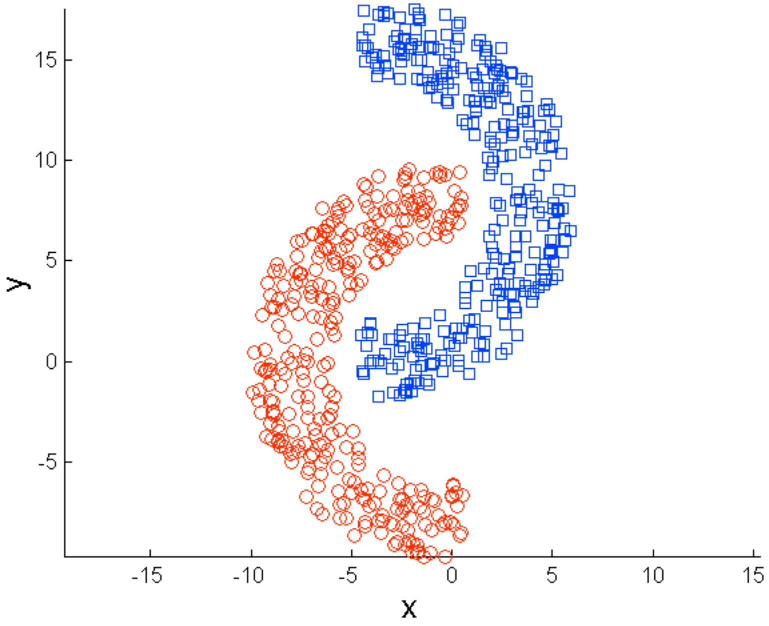


Original Points

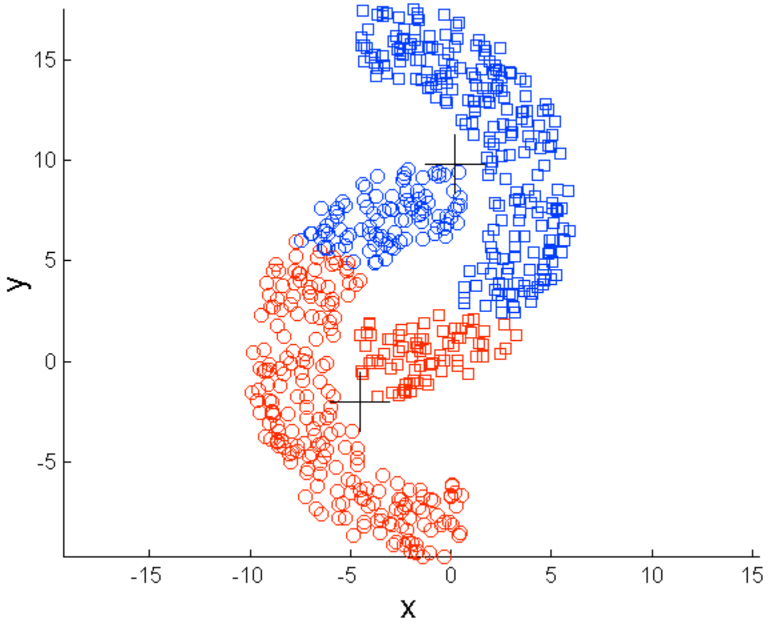


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



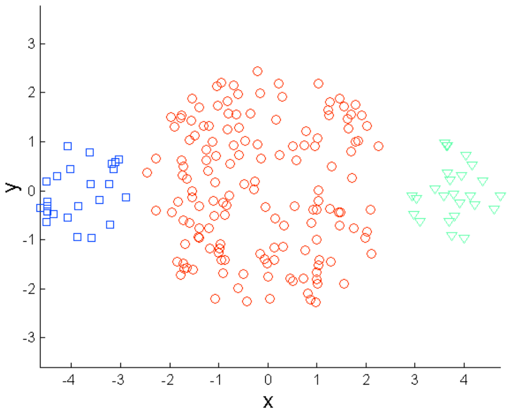
Original Points



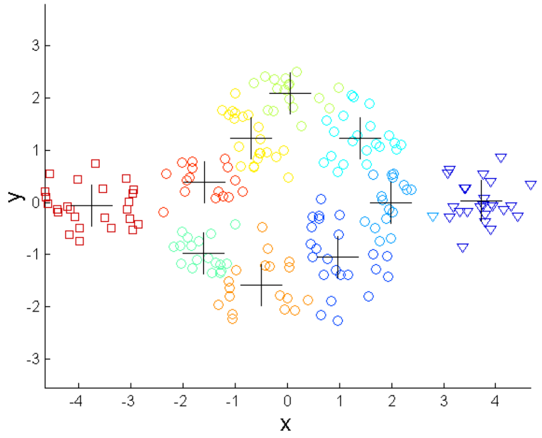
K-means (2 Clusters)

Overcoming K-means Limitations

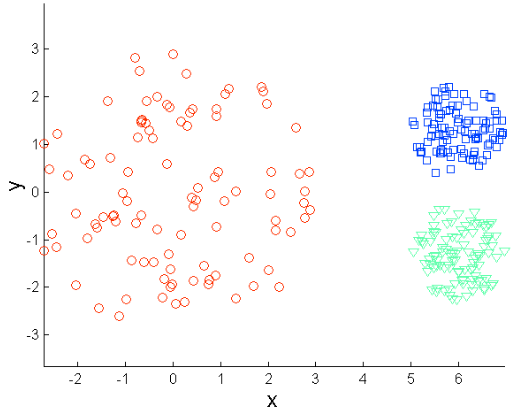
Original Points



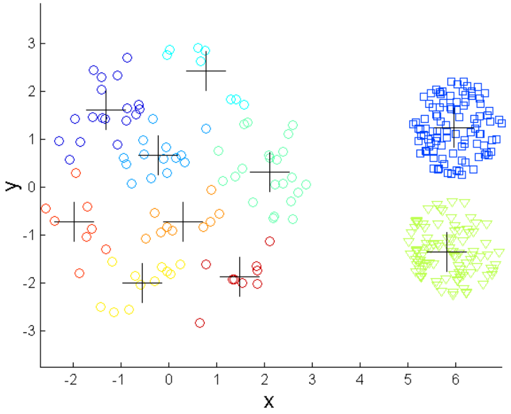
K-means Clusters



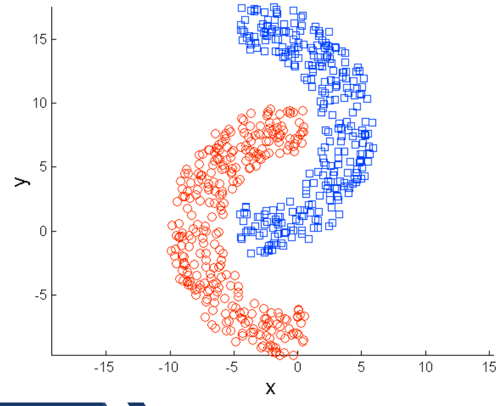
Original Points



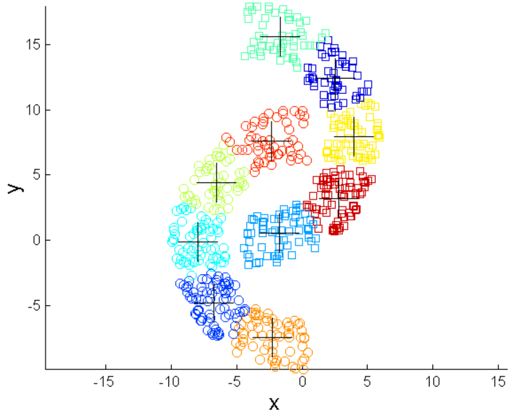
K-means Clusters



Original Points



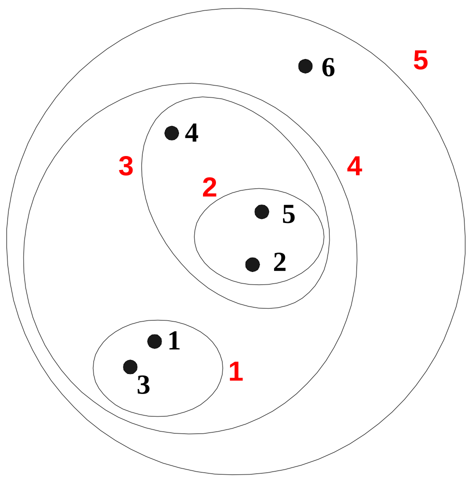
K-means Clusters

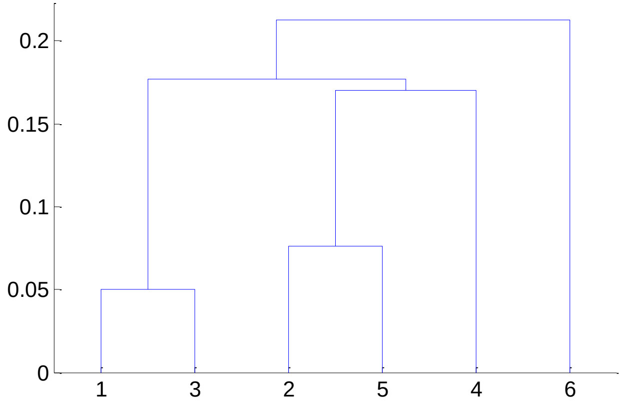


One solution is to use many clusters.
Find parts of clusters, but need to put together.

Hierarchical Clustering

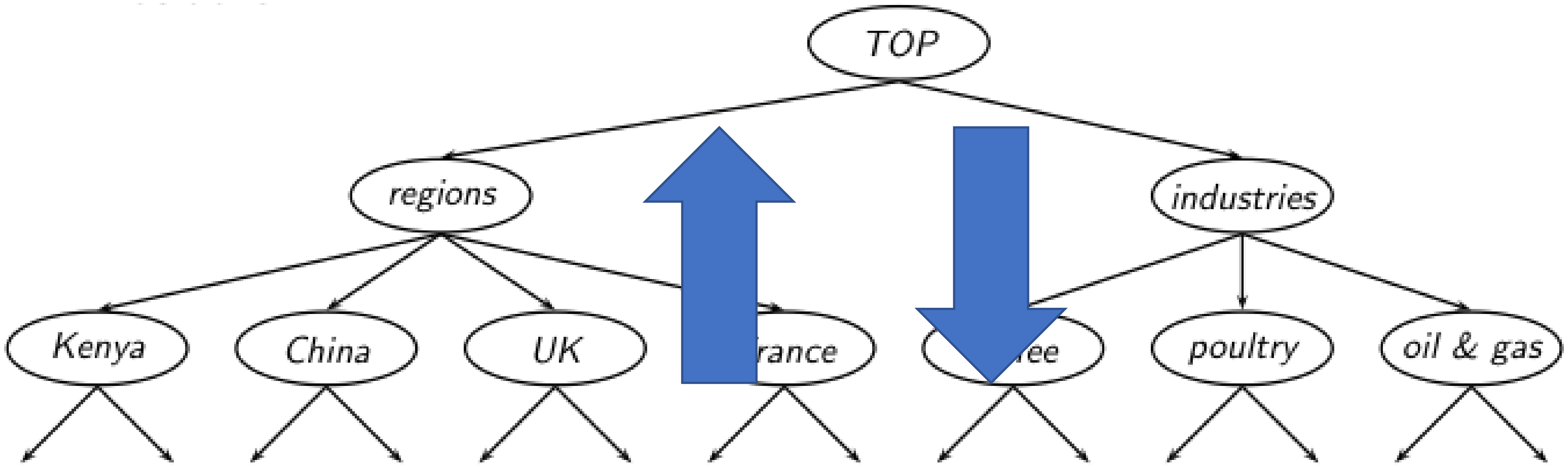
- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





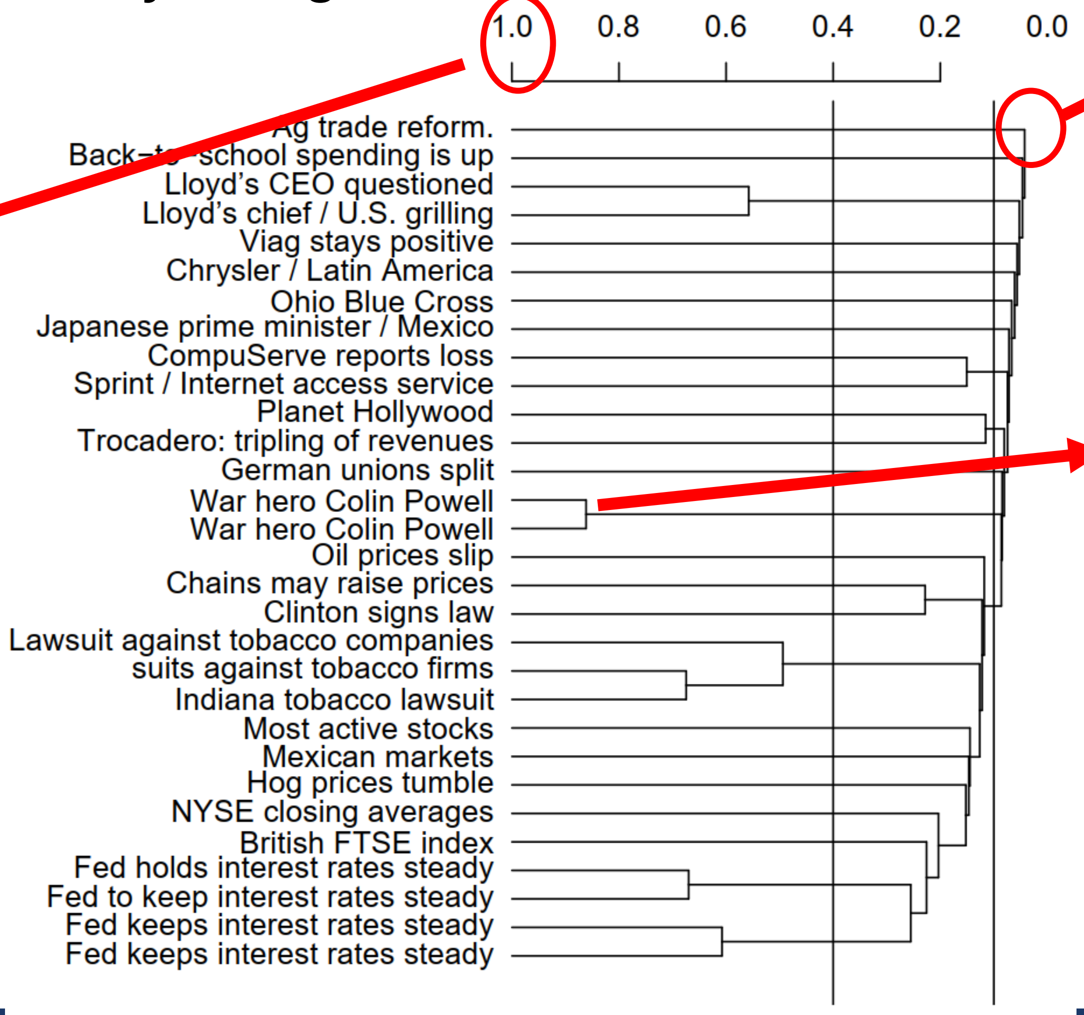
Hierarchical Clustering

- Bottom-up algorithms: hierarchical agglomerative clustering(HAC)
- Top-down algorithms: hierarchical divisive clustering



Hierarchical Agglomerative Clustering (HAC)

- Most popular hierarchical clustering technique
  - Key Idea: Successively merge **closest** clusters



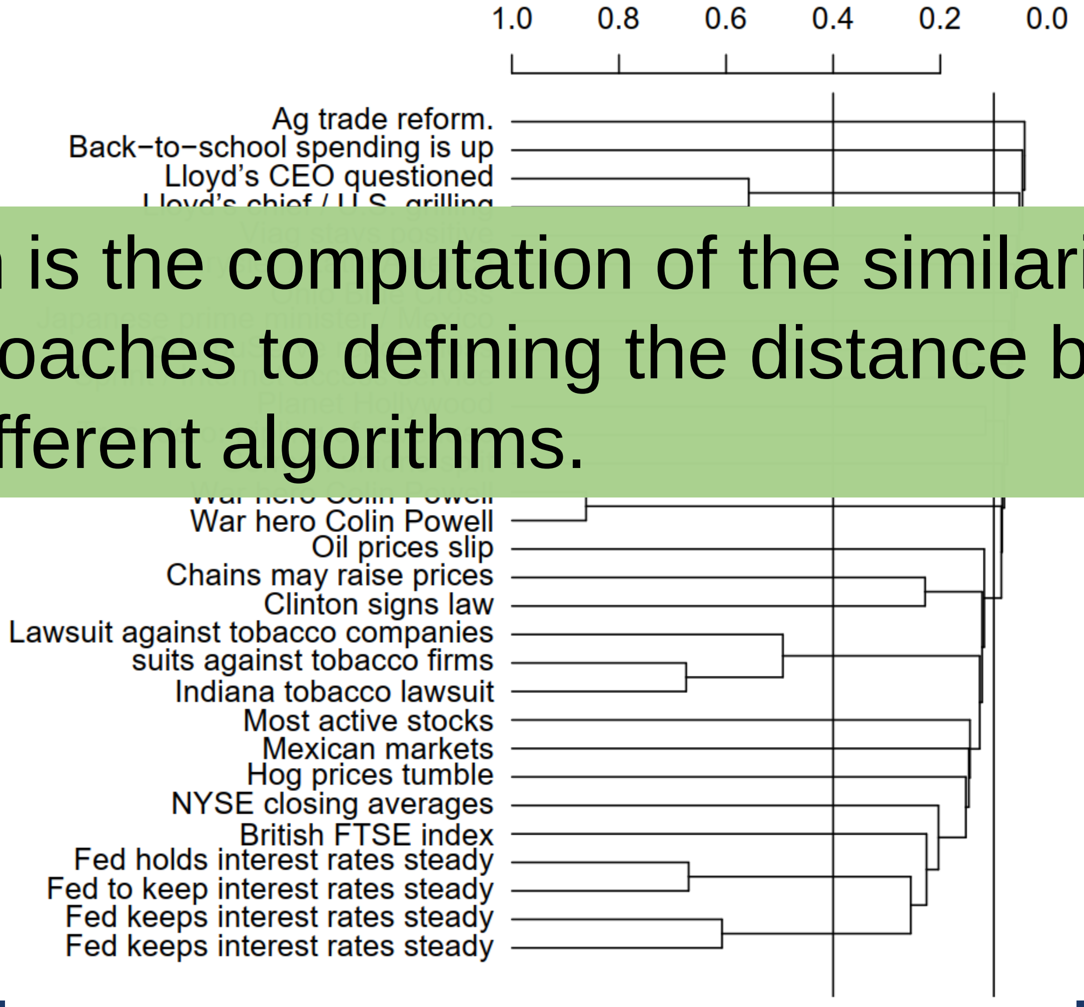
Similarity between clusters (combination similarity)

the last agglomerate

First agglomerate

Agglomerative Clustering Algorithm

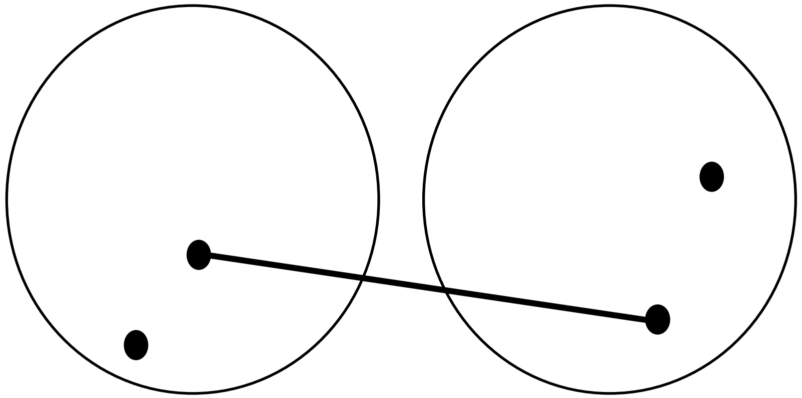
- Most popular hierarchical clustering technique
  - Key Idea: Successively merge **closest** clusters



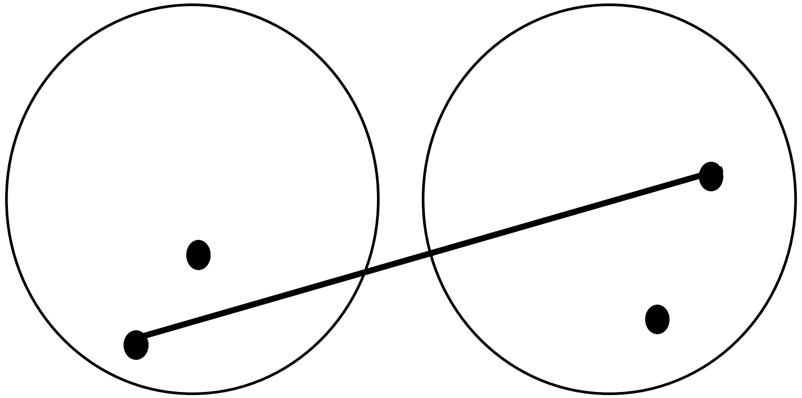
(1) Key operation is the computation of the similarity of two clusters.
(2) Different approaches to defining the distance between clusters distinguish the different algorithms.

How to Define Inter-Cluster Distance

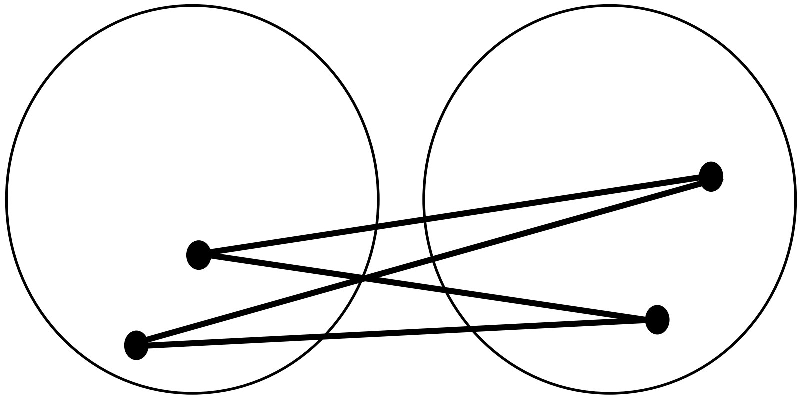
- Similarity Measures



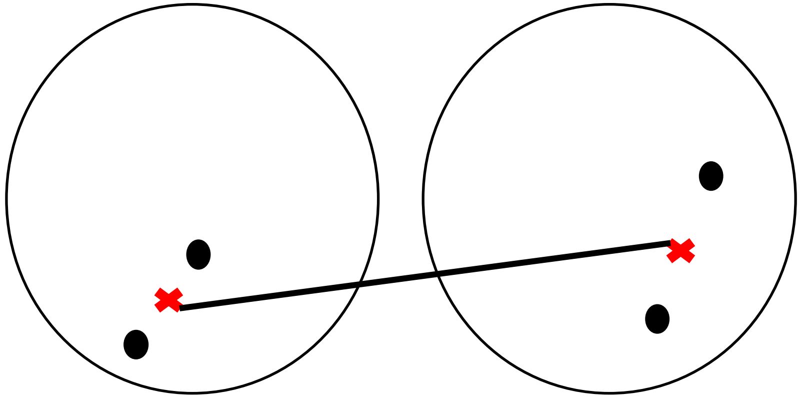
minimum distance



maximum distance

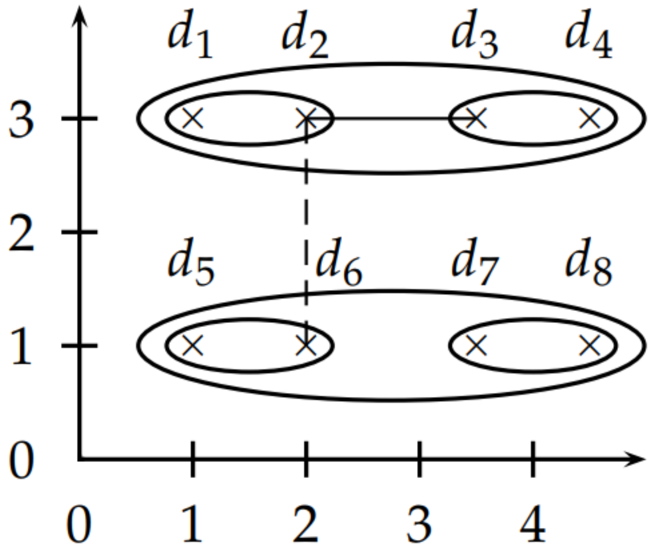


average distance

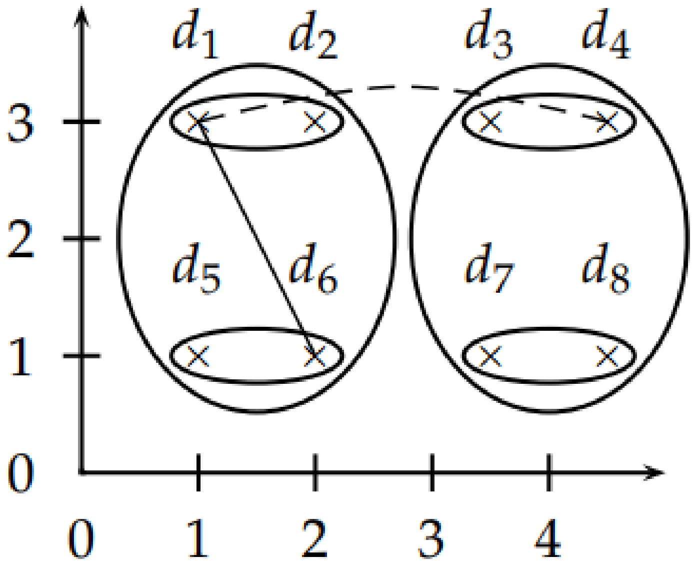


mean distance

Different similarity measures make different results

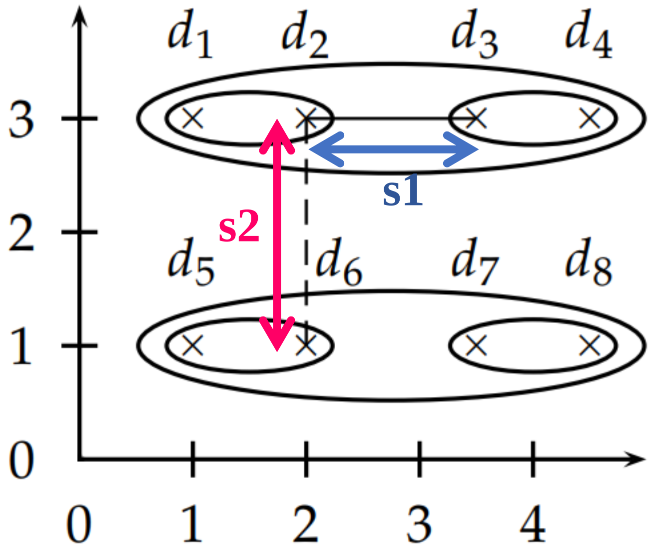


minimum distance

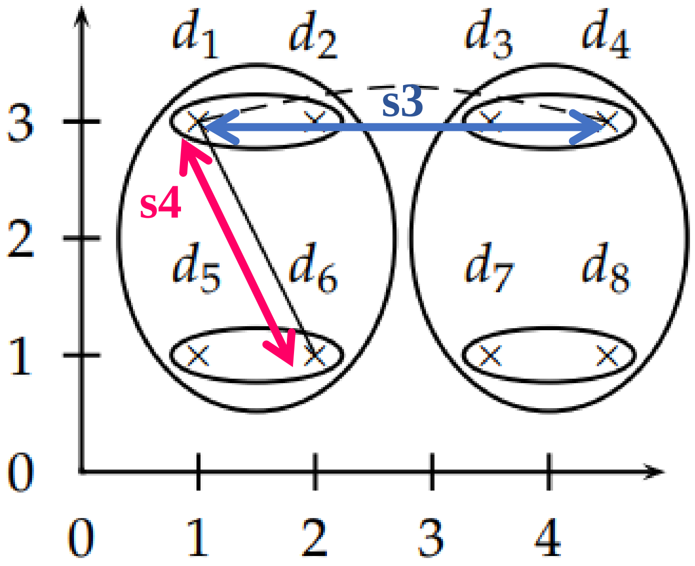


maximum distance

Different similarity measures make different results

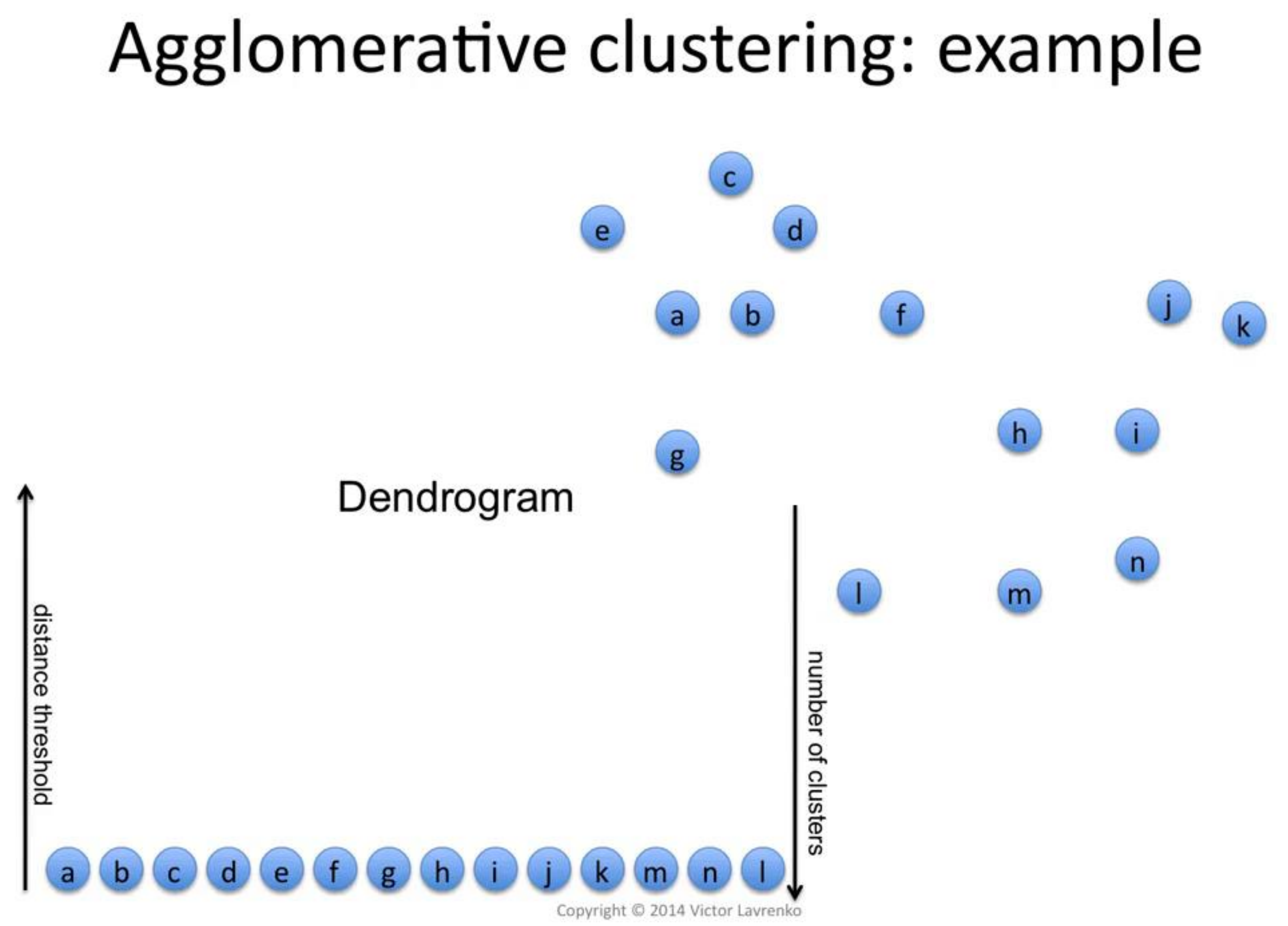


minimum distance



maximum distance

Video: Agglomerative clustering: dendrogram



Density Based Clustering

- Clusters are regions of high density that are separated from one another by regions on low density.

DBSCAN

- DBSCAN is a density-based algorithm
  - Density = number of points within a specified radius (Eps)

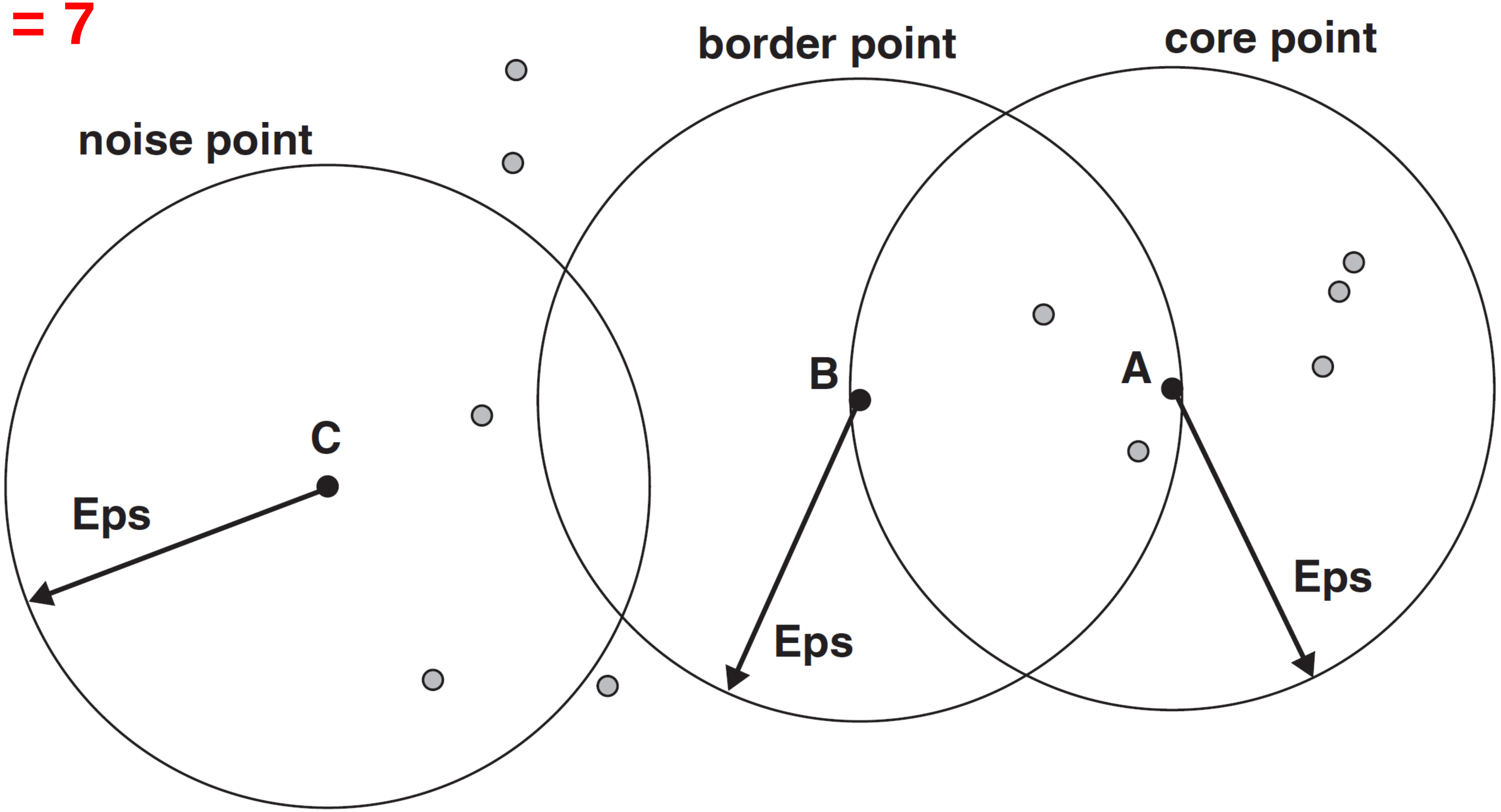
  - Three kinds of points:
  - A point is a **core point** if it has at least a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
    - Counts the point itself

  - A **border point** is not a core point, but is in the neighborhood of a core point

  - A **noise point** is any point that is not a core point or a border point

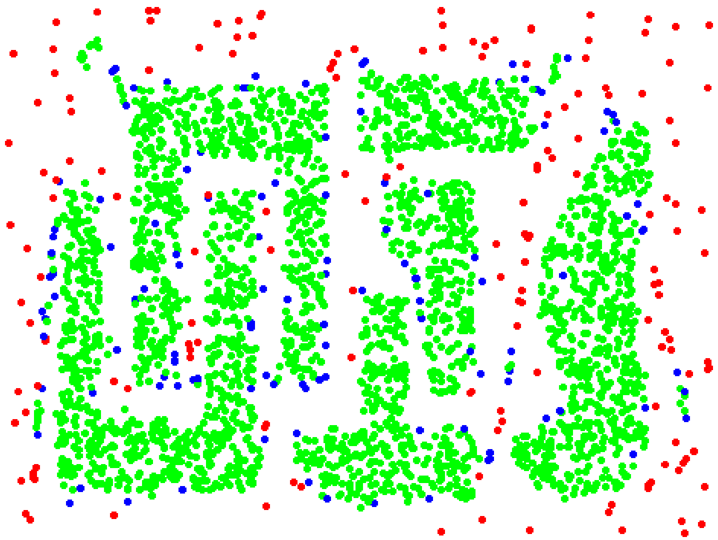
DBSCAN

MinPts = 7



DBSCAN

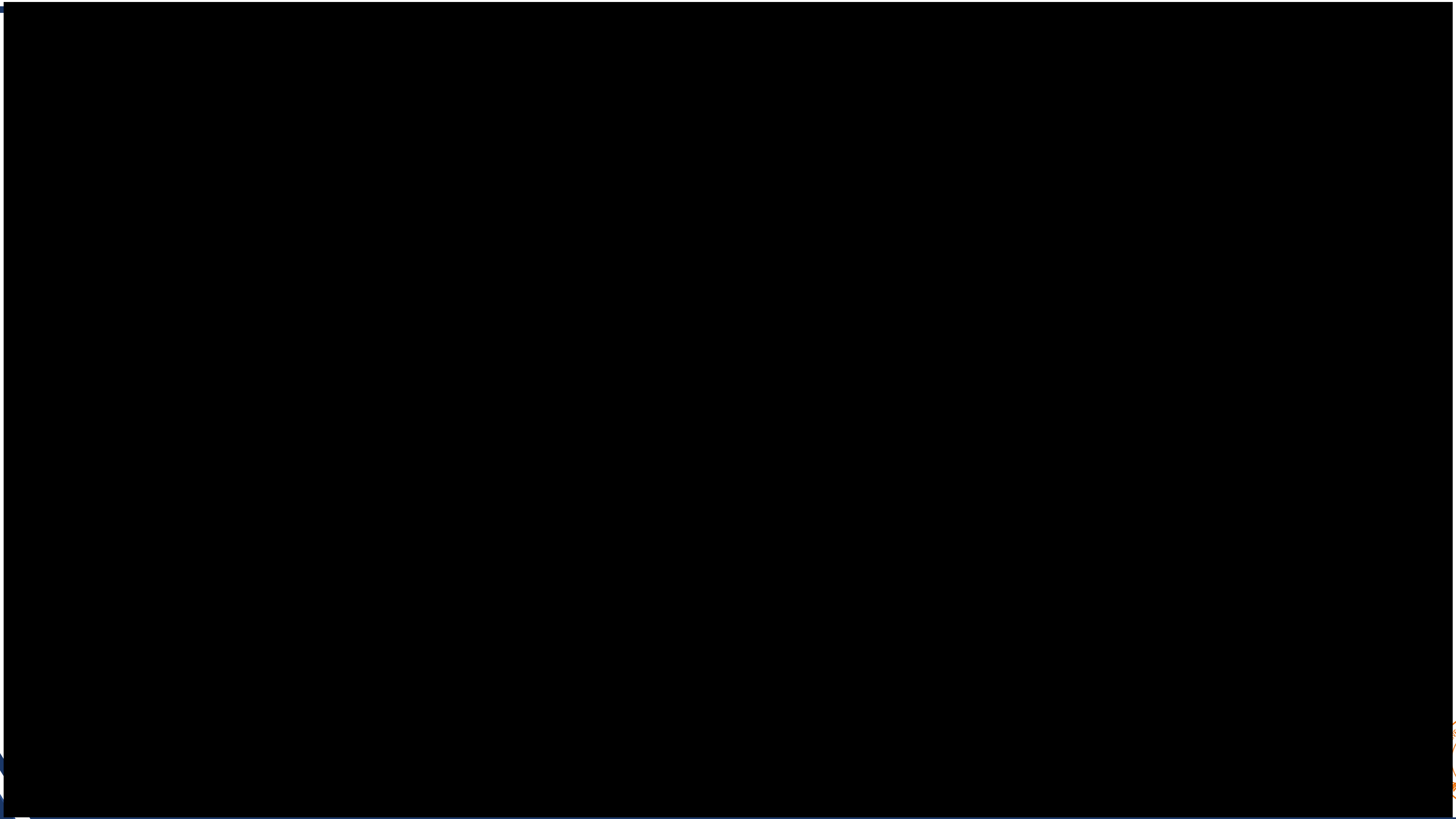
Original Points



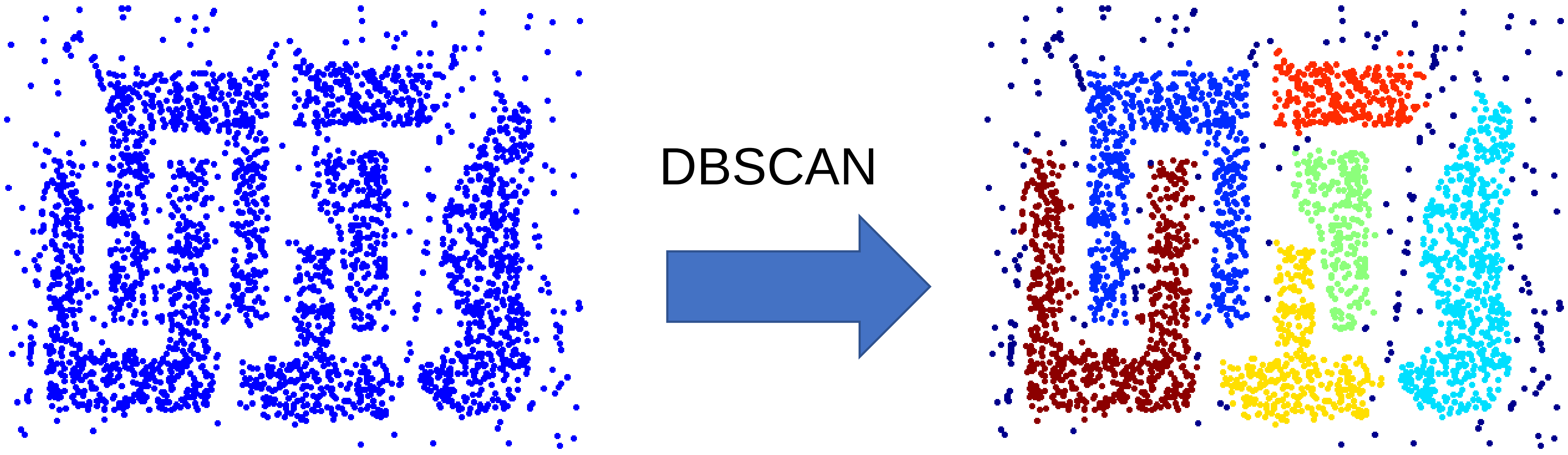
Point types: core, border and noise

Eps = 10, MinPts = 4

DBSCAN



DBSCAN



Original Points

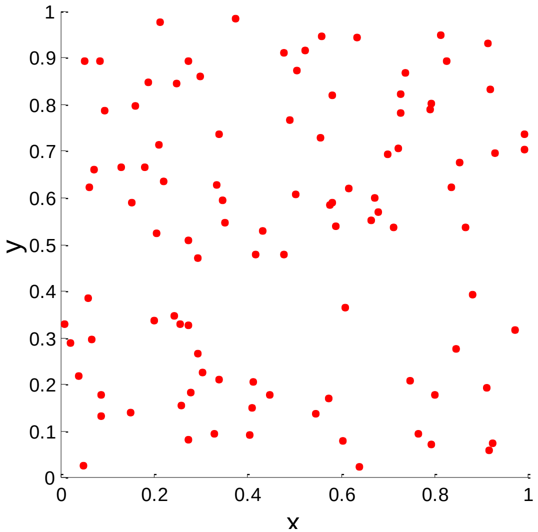
Clusters

Cluster Validity

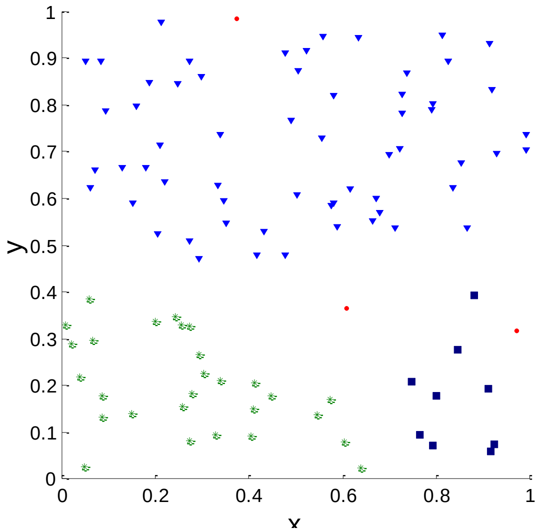
- How to evaluate the “goodness” of the resulting clusters?
- Why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Cluster Validation

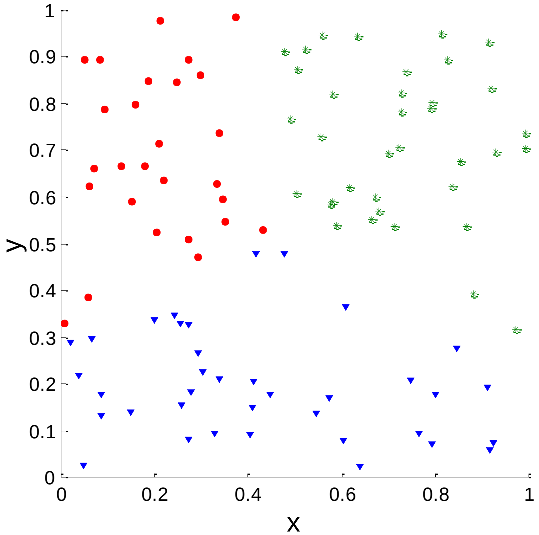
Random Points



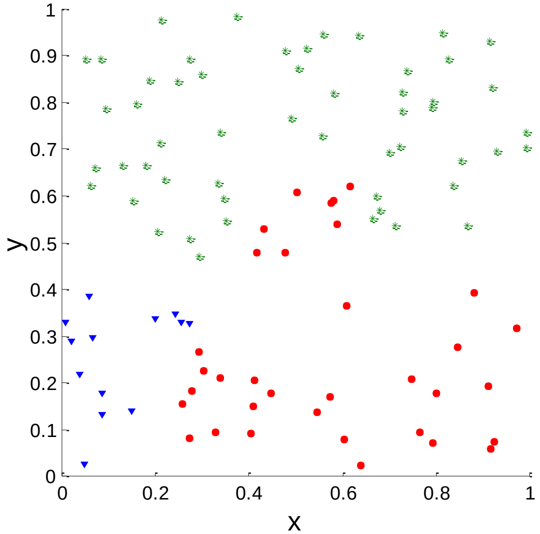
DBSCAN



K-means



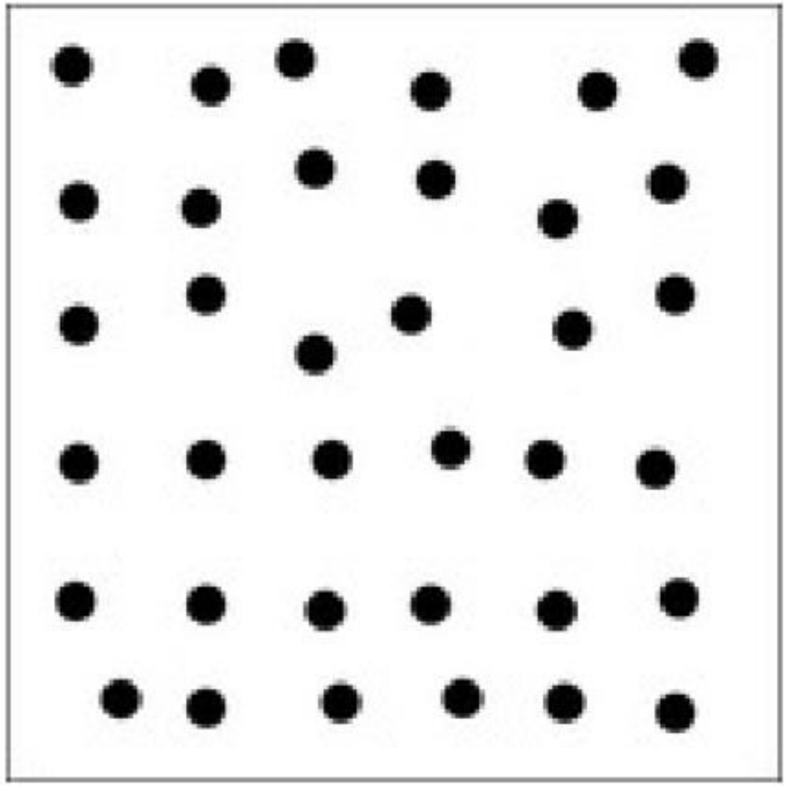
Agglomerative clustering



Different Aspects of Cluster Validation

- Assessing Clustering Tendency
- Determining the number of clusters in a data set
- Measuring clustering quality

Assessing Clustering Tendency



A data set that is uniformly distributed in the data space.

Determining the number of clusters in a data set

- Determining the "right" number of clusters in a data set is important.
  - A simple method is to set the number of clusters to about $\sqrt{\frac{n}{2}}$ for a data set of n points. In expectation, each cluster has $\sqrt{2n}$ points

Measuring Clustering Quality

- Extrinsic Methods: Entropy and Purity
- Internal Measures: Silhouette Coefficient

Advanced Cluster Analysis

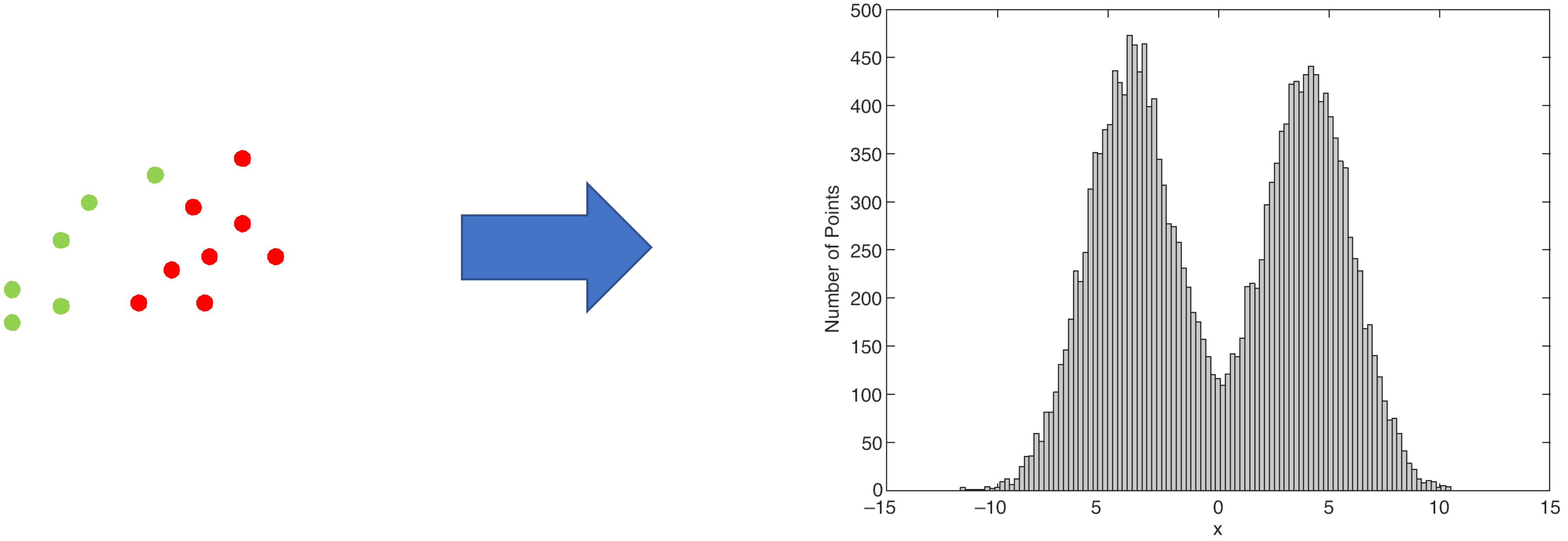
Advanced Topics

- Probabilistic model-based clustering
- Clustering high-dimensional data

Hard vs Soft Clustering

- Hard clustering: Clusters do not overlap
  - Element either belongs to cluster or it does not
- Soft clustering: cluster may overlap
  - Strength of association between clusters and instances

Probabilistic model-based clustering



Probabilistic model-based clustering

- Each cluster can be represented mathematically by a parametric probability distribution (e.g., Gaussian or Poisson distribution)
- Cluster: Data points (or objects) that most likely belong to the same distribution
- Clustering: Parameter estimation so that they will have a maximum likelihood fit to the model by a mixture of K component distributions (i.e., K clusters)

Probabilistic model-based clustering

- the Expectation-Maximization (EM) algorithm

Initialize the parameters

Repeat

For each point, compute its probability under each distribution

Using these probabilities, update the parameters of each distribution

Until there is no change

K-Means → EM

- **Boot Step**:
  - Initialize K clusters: $C_1, \ldots, C_K$

    (μ_j, Σ_j) and $P(C_j)$ for each cluster j.
- **Iteration Step**:
  - Estimate the cluster of each data (assign points to clusters)

    $$p(C_j \mid x_i)$$ ⇨ Expectation

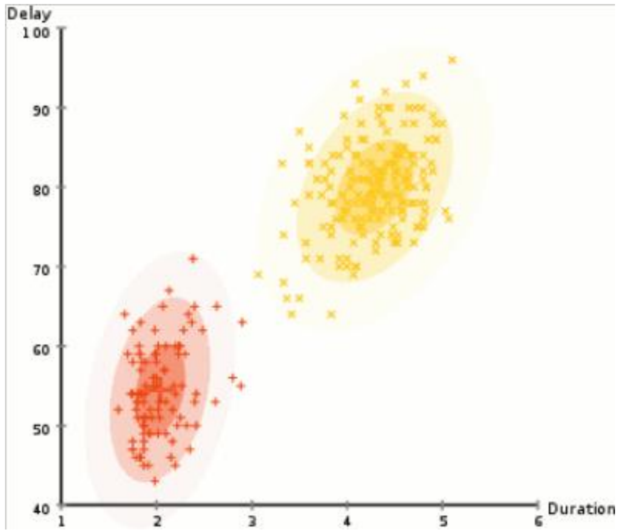
  - Re-estimate the cluster parameters (estimate model parameters)

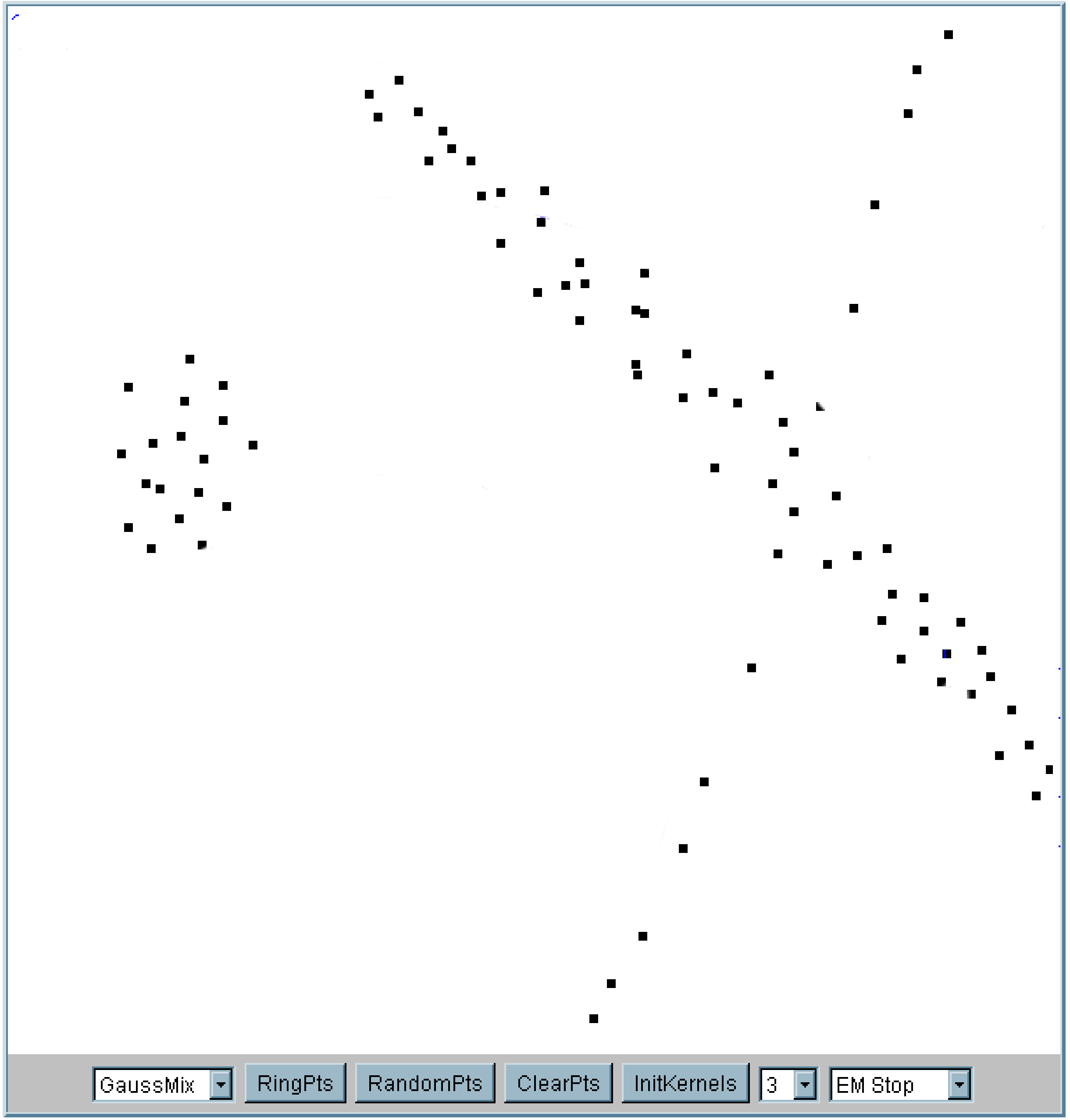
    $$(\mu_j, \Sigma_j), p(C_j) \text{ For each cluster } j$$ ⇨ Maximization

- Assume the distributions of clusters follow Gaussian distribution
- Estimate the parameters (mean and variance) by using EM algorithm.

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

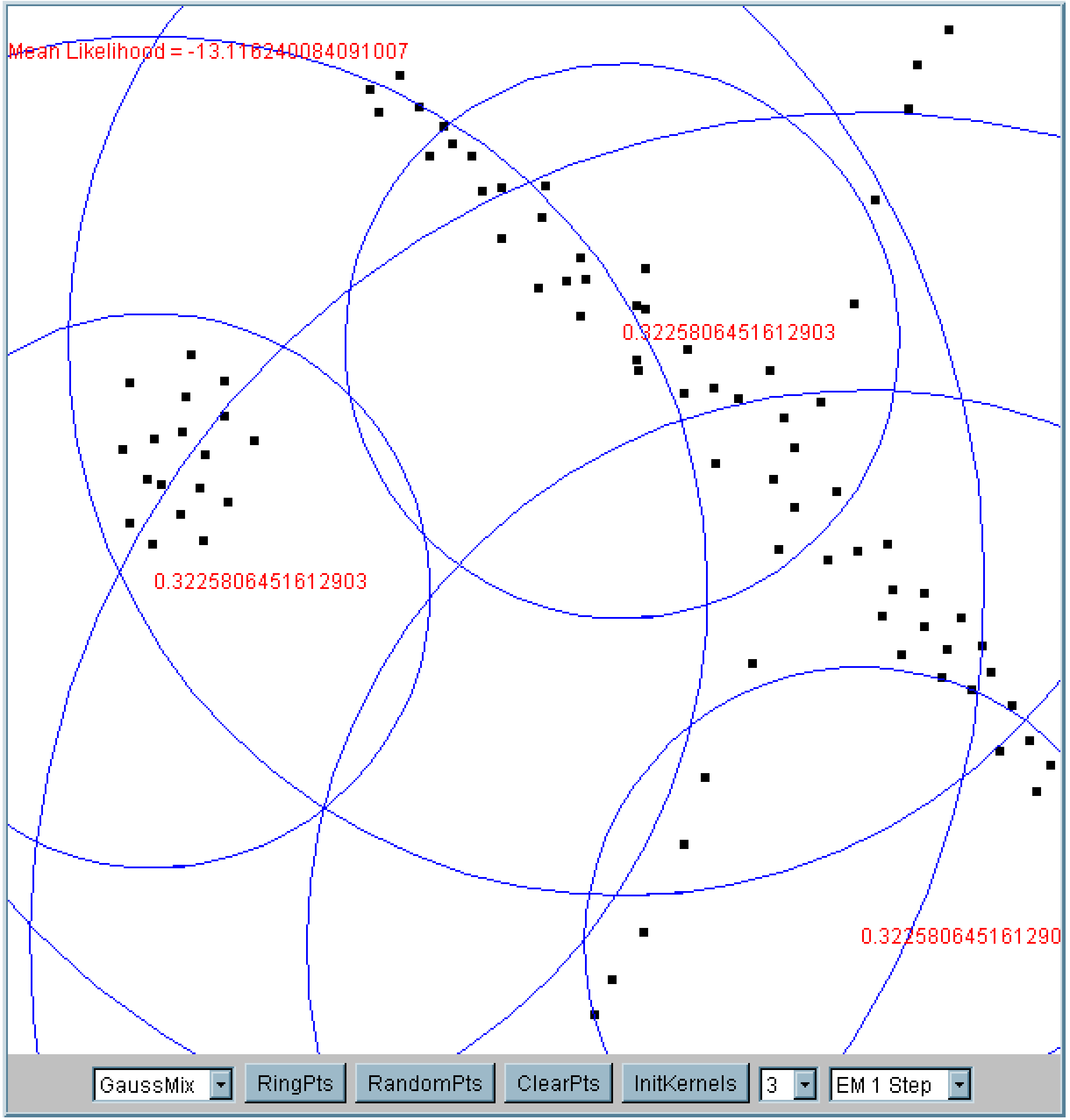
$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_t)^T \Sigma^{-1} (X_t-\mu_t)}$$



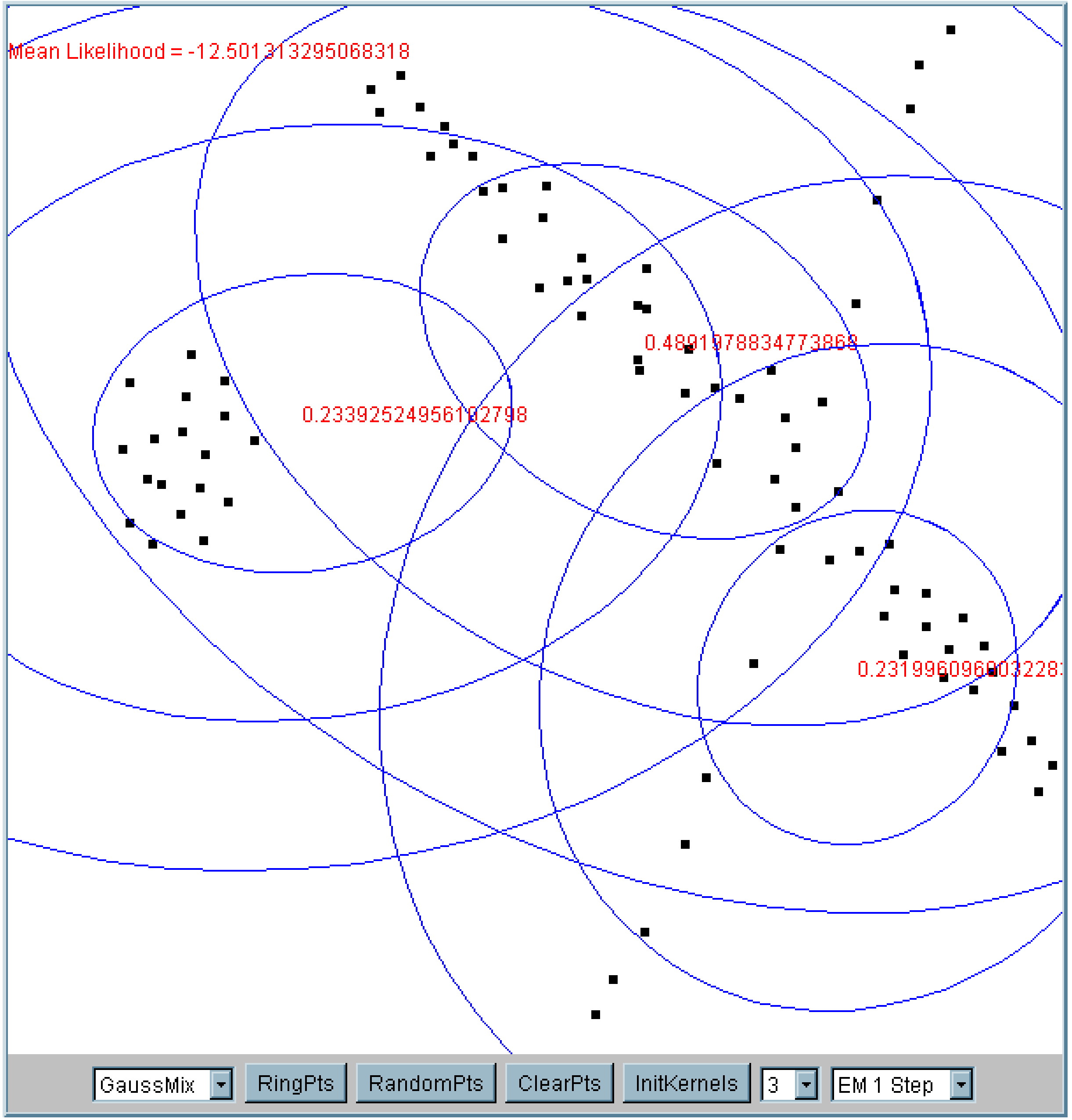

GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM Stop

Iteration 1

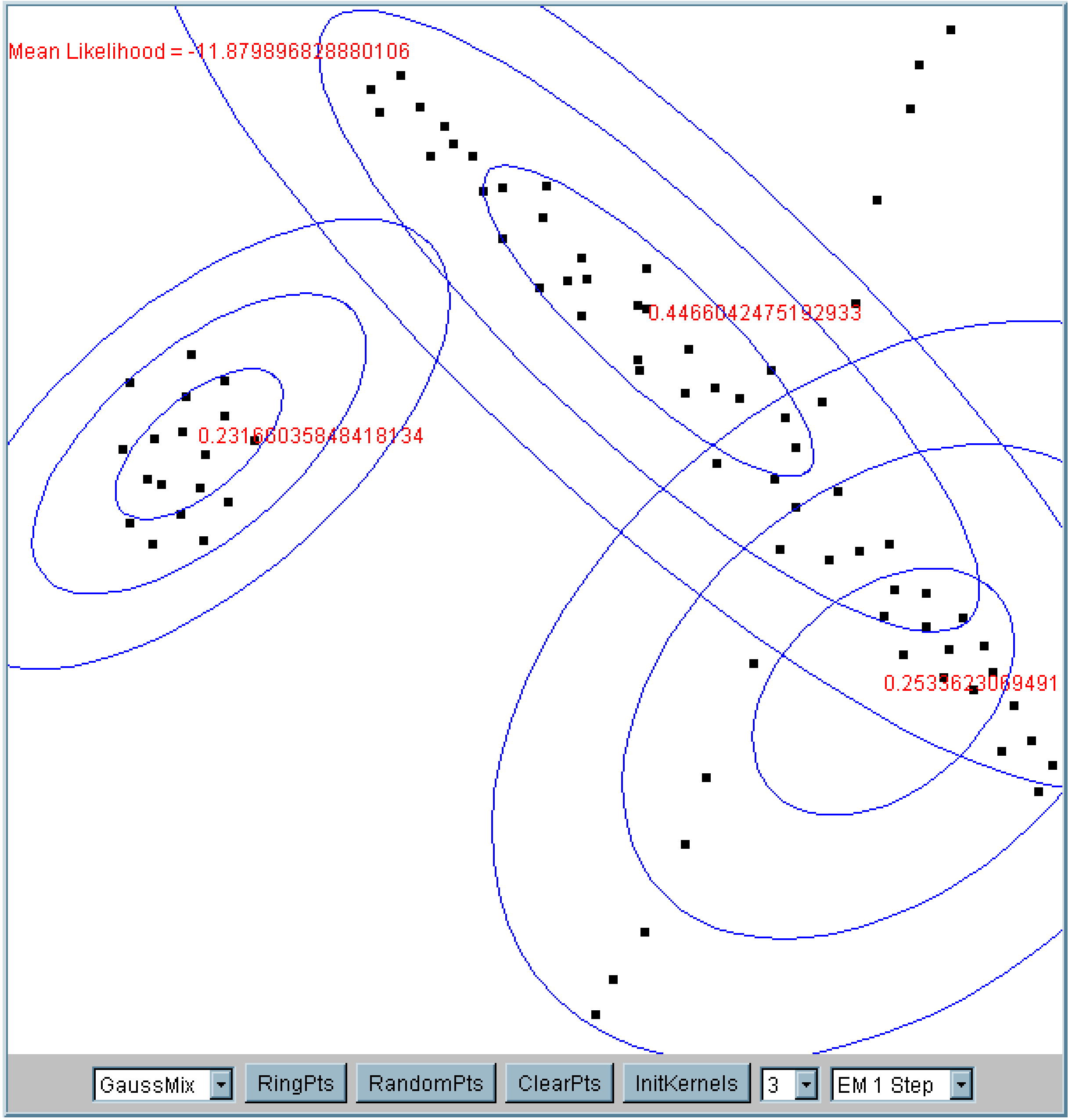
The cluster means are randomly assigned



Iteration 2

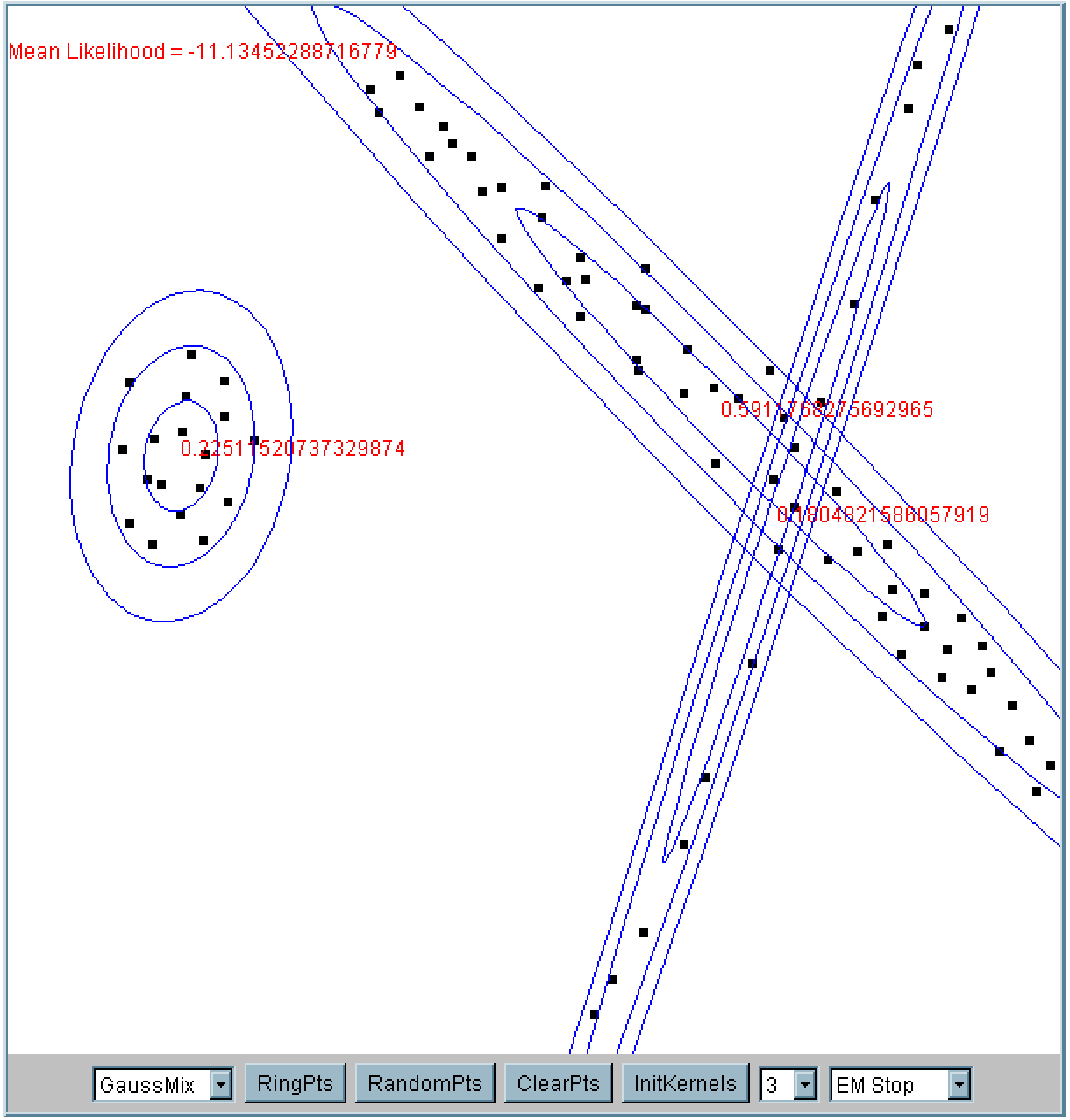

Mean Likelihood = -12.501313295068318
0.4891978834773862
0.2339252495610279
0.2319960960322
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM 1 Step

Iteration 5



Mean Likelihood = -11.879896828880106
0.4466042475192933
0.23166035848418134
0.2533623069491
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM 1 Step

Iteration 25



EM Algorithm

Algorithm 9.2 EM algorithm.

1: Select an initial set of model parameters.
(As with K-means, this can be done randomly or in a variety of ways.)
2: **repeat**
3: **Expectation Step** For each object, calculate the probability that each object belongs to each distribution, i.e., calculate $prob(distribution\ j|\mathbf{x}_i, \Theta)$.
4: **Maximization Step** Given the probabilities from the expectation step, find the new estimates of the parameters that maximize the expected likelihood.
5: **until** The parameters do not change.
(Alternatively, stop if the change in the parameters is below a specified threshold.)

Application

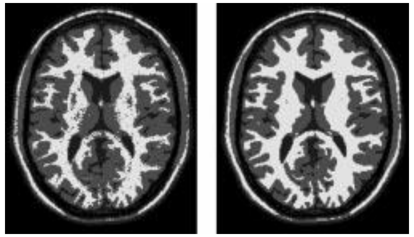
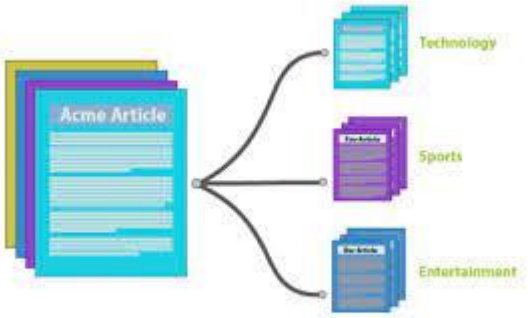
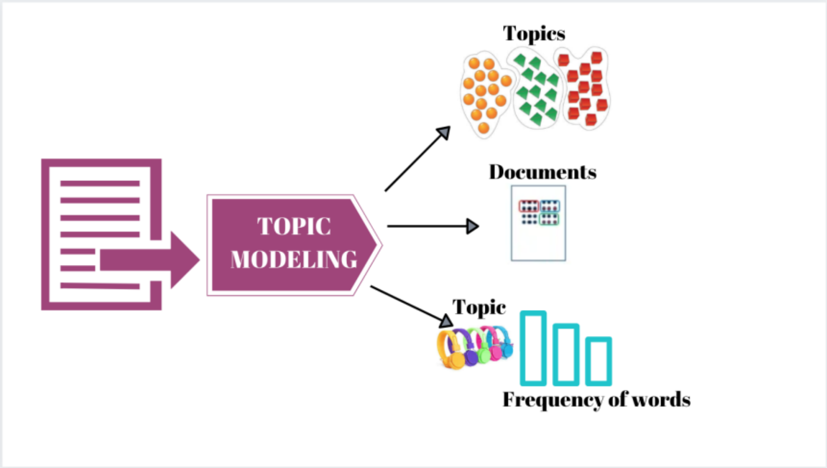


Image segmentation



Document clustering



Topic modeling

Clustering high-dimensional data

- Why cluster high-dimensional data?
  - Many applications, such text documents or DNA micro-array data, may need to handle tens of thousands of dimensions
  - Many clustering algorithms may not work well when the number of dimensions grows to 20.

Clustering high-dimensional data

- Challenges:
  - Many irrelevant dimensions may mask clusters
  - Distance measures becomes meaningless
  - Clusters may exist only in some subspaces

Clustering high-dimensional data

- Subspace clustering approaches
- Dimensionality reduction approaches

End of Class 4