7 – Clustering

Computational Intelligence for the Internet of Things (2019-20)

João Paulo Carvalho
joao.carvalho@inesc-id.pt

INESC-ID
Instituto Superior Técnico, Universidade de Lisboa

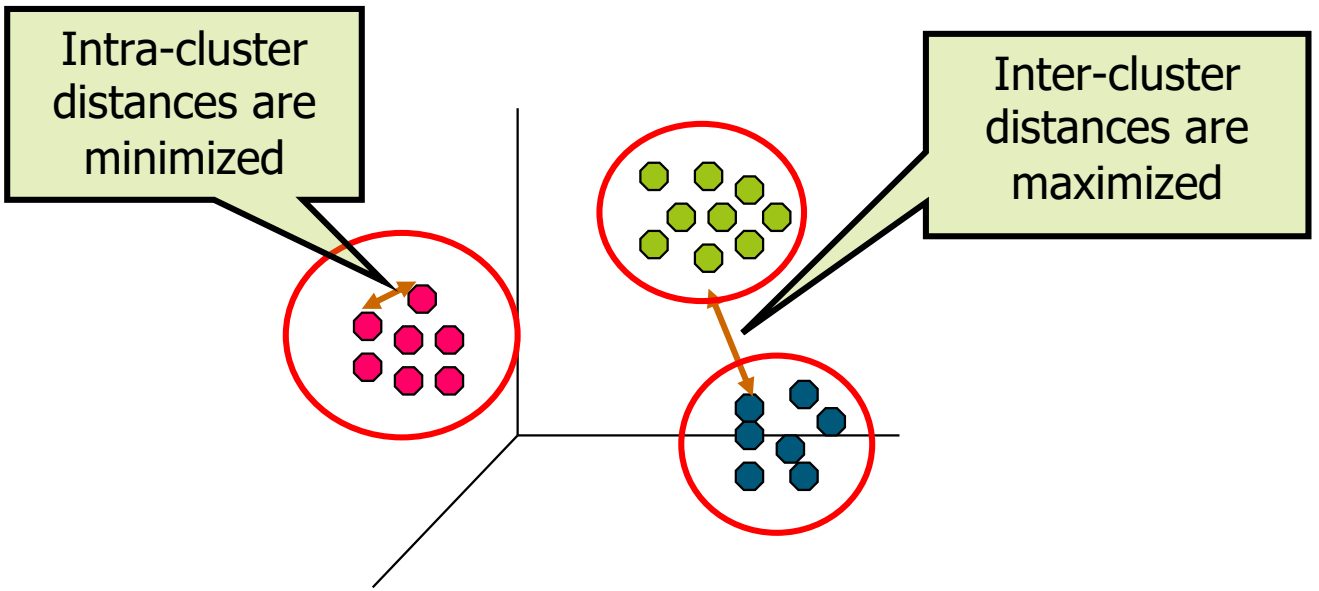
Clustering

- Introduction
- K-Means
- Fuzzy Clustering
- Hierarchical Clustering
- Density-Based Clustering

Introduction

What is Clustering?

- In general, a grouping of objects such that the objects in a group (cluster) are similar (or related) to one another and different from (or unrelated to) the objects in other groups



Cluster Analysis and Applications

- "Discovery" of new knowledge from data
  - Contrast with supervised classification (where labels are known)
  - Long history in the sciences of categories, taxonomies, etc.
  - Can be very useful for summarizing large data sets
    - For large *n* and/or high dimensionality
- Applications of clustering
  - Clustering of documents produced by a search engine
  - Segmentation of customers for an e-commerce store
  - Discovery of new types of galaxies in astronomical data
  - Clustering of data from different sensors
  - Cluster pixels in an image into regions of similar intensity…

Early Applications of Cluster Analysis

- Dr. John Snow's study of the 1854 London cholera outbreak is an historic example of a cluster analysis that suggested an effective intervention
- The outbreak of cholera was detected by Dr. Snow even before the bacterium that causes cholera had been identified
- He mapped mortality and found that most deaths occurred near the Broad Street water pump
- Once the handle of the pump was removed, the outbreak subsided

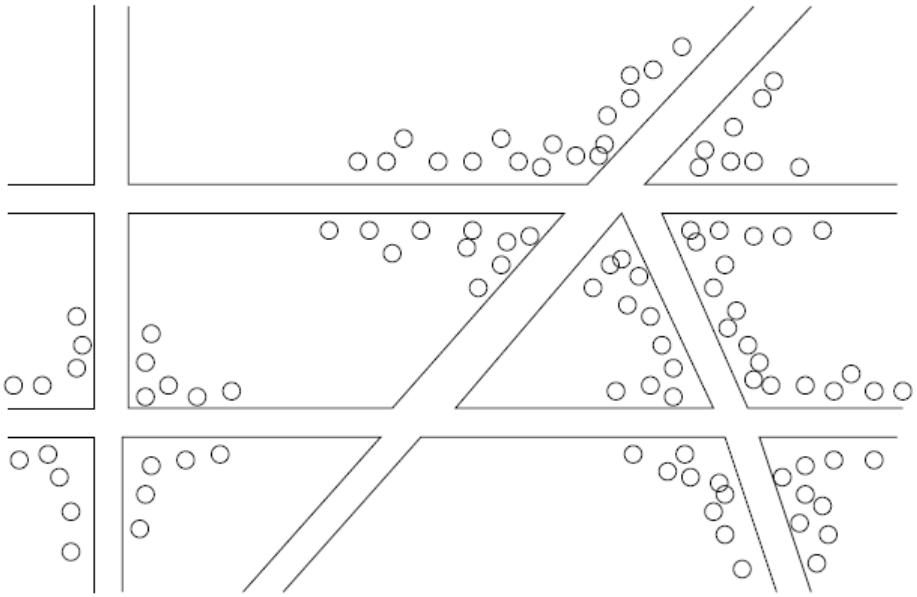


Figure 1.1: Plotting cholera cases on a map of London

Clusters can be Ambiguous...

How many clusters?

Six Clusters

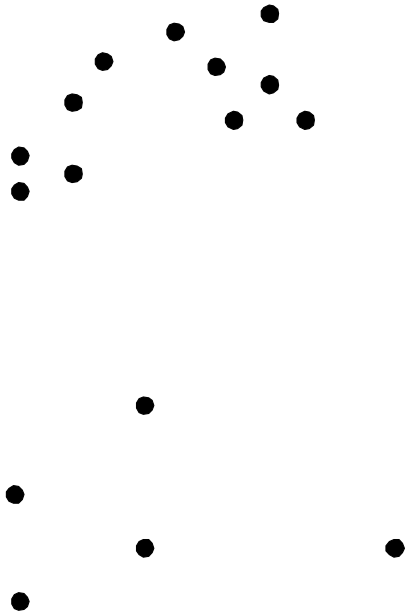
Two Clusters

Four Clusters

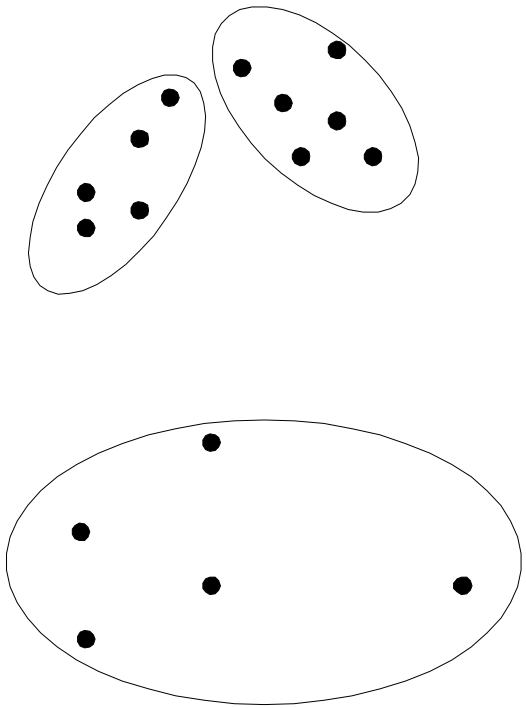
Clustering Approaches

- Clustering approaches:
  - Partitional Clustering
    - Divide data objects into subsets (clusters) such that each data object is in exactly one subset
  - Hierarchical clustering
    - A set of nested clusters organized as a hierarchical tree
- Type of subsets:
  - Hard clustering
  - Fuzzy clustering

Partitional Clustering

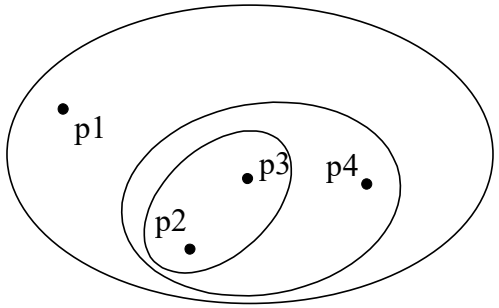


Original Points

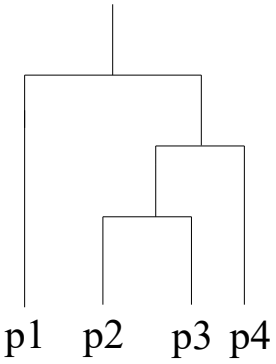


A Partitional Clustering

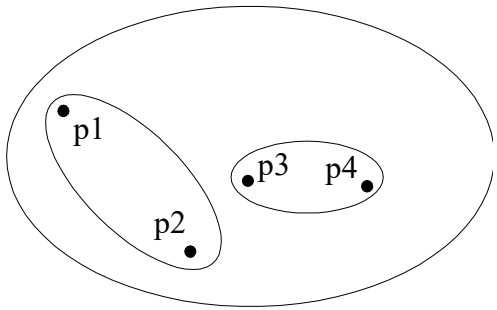
Hierarchical Clustering



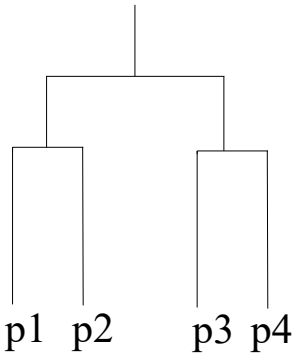
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



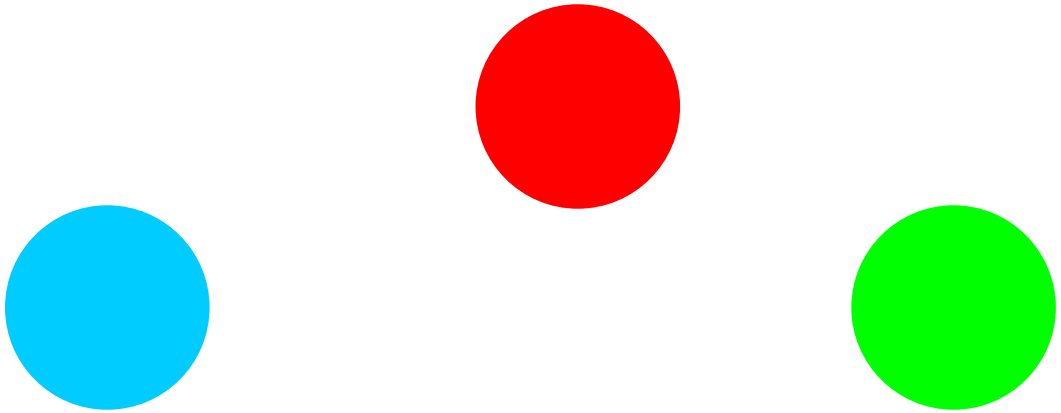
Non-traditional Dendrogram

Crisp vs. Fuzzy Clustering

- Crisp clustering algorithms
  - Partition the data set into disjoint groups, i.e., each data point belongs to one cluster only
  - Similarity is quantified using some metric
- Fuzzy clustering algorithms
  - Partition the data set into overlapping groups, i.e., each data point belongs to multiple clusters with varying degree of membership
  - Similarity is quantified using some metric which is modified by membership values

Types of Clusters: Well-Separated

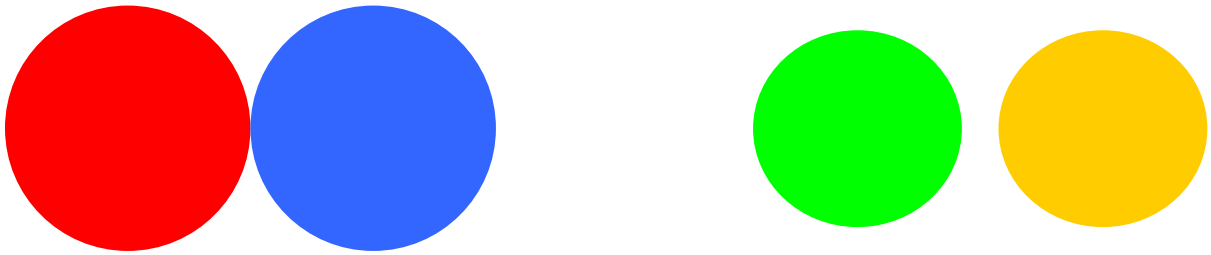
- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster



3 well-separated clusters

Types of Clusters: Center-Based

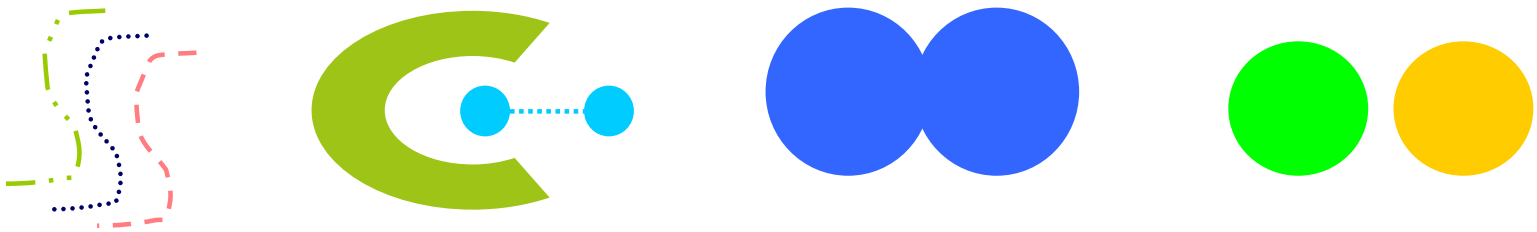
- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the minimizer of distances from all the points in the cluster, or a medoid,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



4 center-based clusters

Types of Clusters: Contiguity-Based

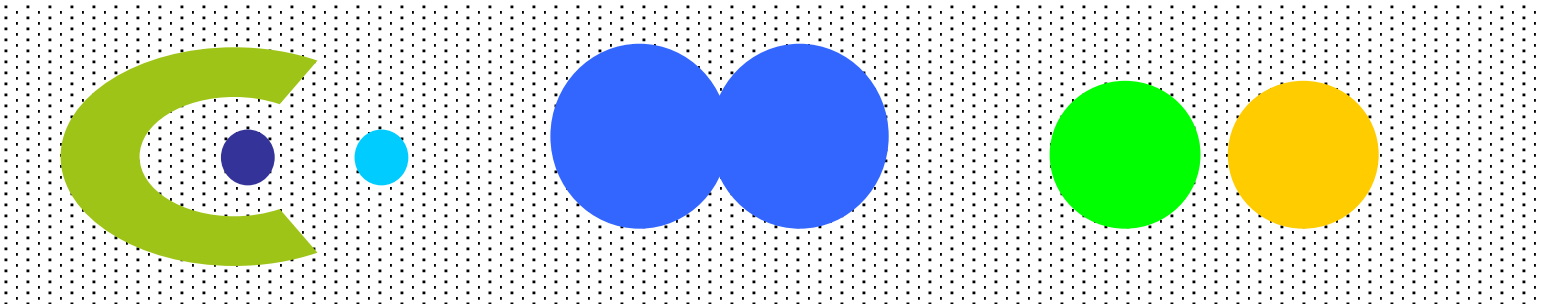
- Contiguous Cluster (Nearest Neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster



8 contiguous clusters

Types of Clusters: Density-Based

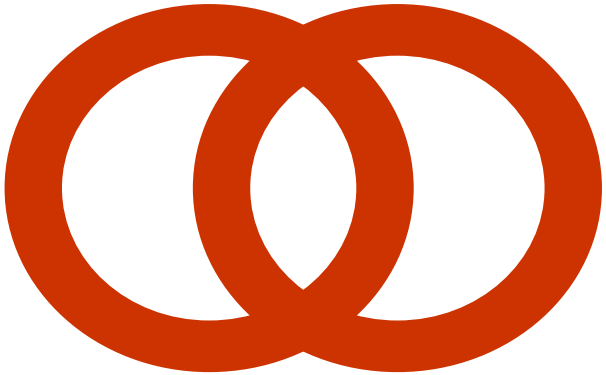
- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present



6 density-based clusters

Types of Clusters: Conceptual Clusters

- Shared Property or Conceptual Clusters
  - Finds clusters that share some common property or represent a particular concept



2 Overlapping Circles

Objective Function

- Clustering as an optimization problem
  - Find clusters that minimize or maximize an objective function
  - Enumerate all possible ways of dividing the points into clusters and evaluate the "goodness" of each potential set of clusters by using the given objective function (NP Hard)
  - Can have global or local objectives
    - Hierarchical clustering algorithms typically have local objectives
    - Partitional algorithms typically have global objectives

Objective Function

- Clustering as an optimization problem (cont.)
  - A variation of the global objective function approach is to fit the data to a parameterized model
    - The parameters for the model are determined from the data, and they determine the clustering
    - E.g., Mixture models assume that the data is a ‘mixture' of a number of statistical distributions

Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- DBSCAN

K-Means

K-Means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The objective is to minimize the sum of distances of the points to their respective centroid

K-Means Clustering

- Problem: Given a set X of n points in a d-dimensional space and an integer K, group the points into K clusters $C = \{C_1, C_2, \ldots, C_k\}$ such that

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} dist(x, c)$$

  is minimized, where c_i is the centroid of the points in cluster C_i

- Most common definition is with euclidean distance, minimizing the Sum of Squares Error (SSE) function

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} (x - c_i)^2$$

Complexity of the K-Means Problem

- NP-hard if the dimensionality of the data is at least 2 ($d>=2$)
  - Finding the best solution in polynomial time is infeasible
- For $d=1$ the problem is solvable in polynomial time
- A simple iterative algorithm works quite well in practice

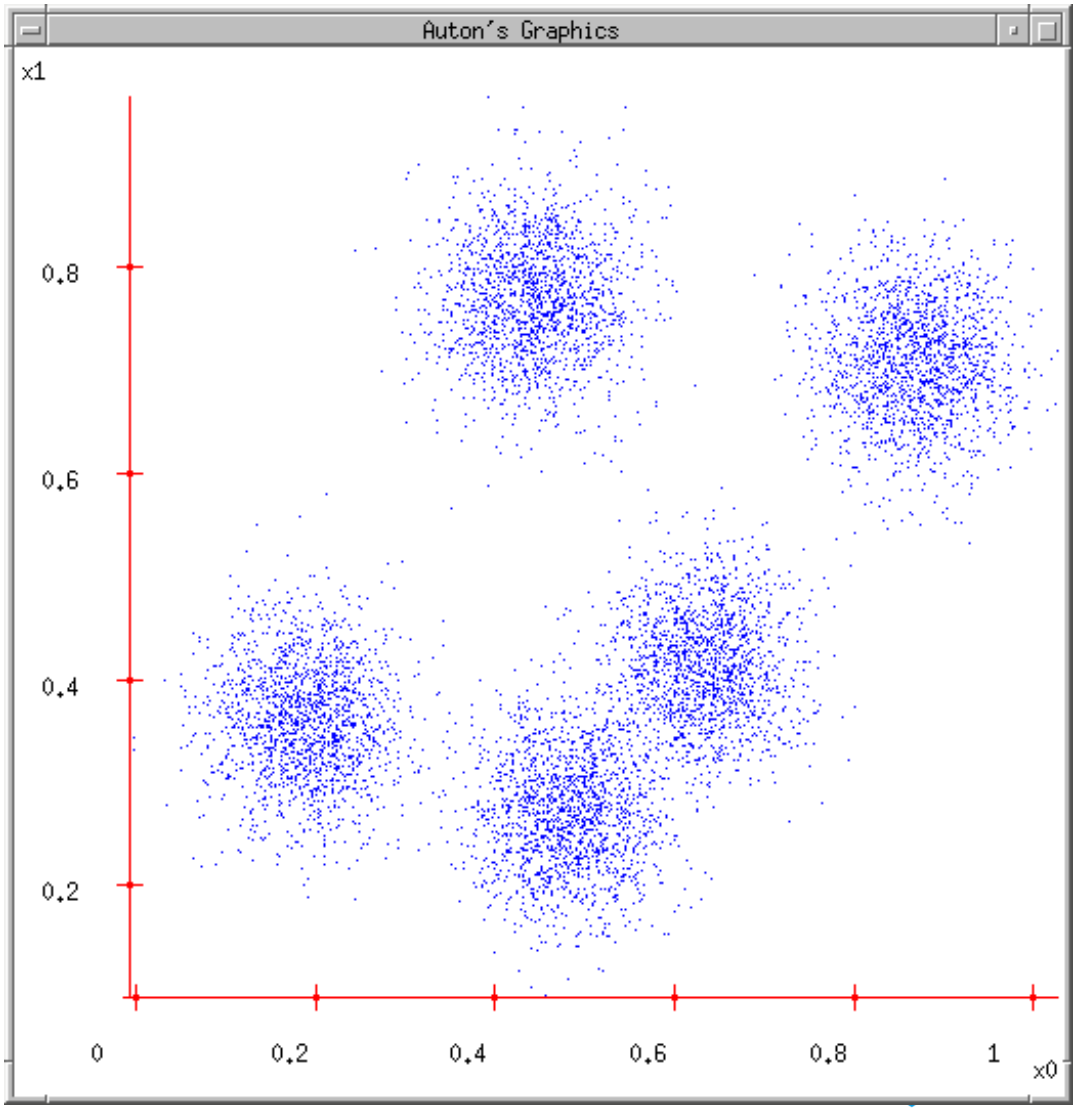
K-Means Algorithm

- Also known as Lloyd's algorithm
  - K-means is sometimes synonymous with this algorithm

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 Algorithm

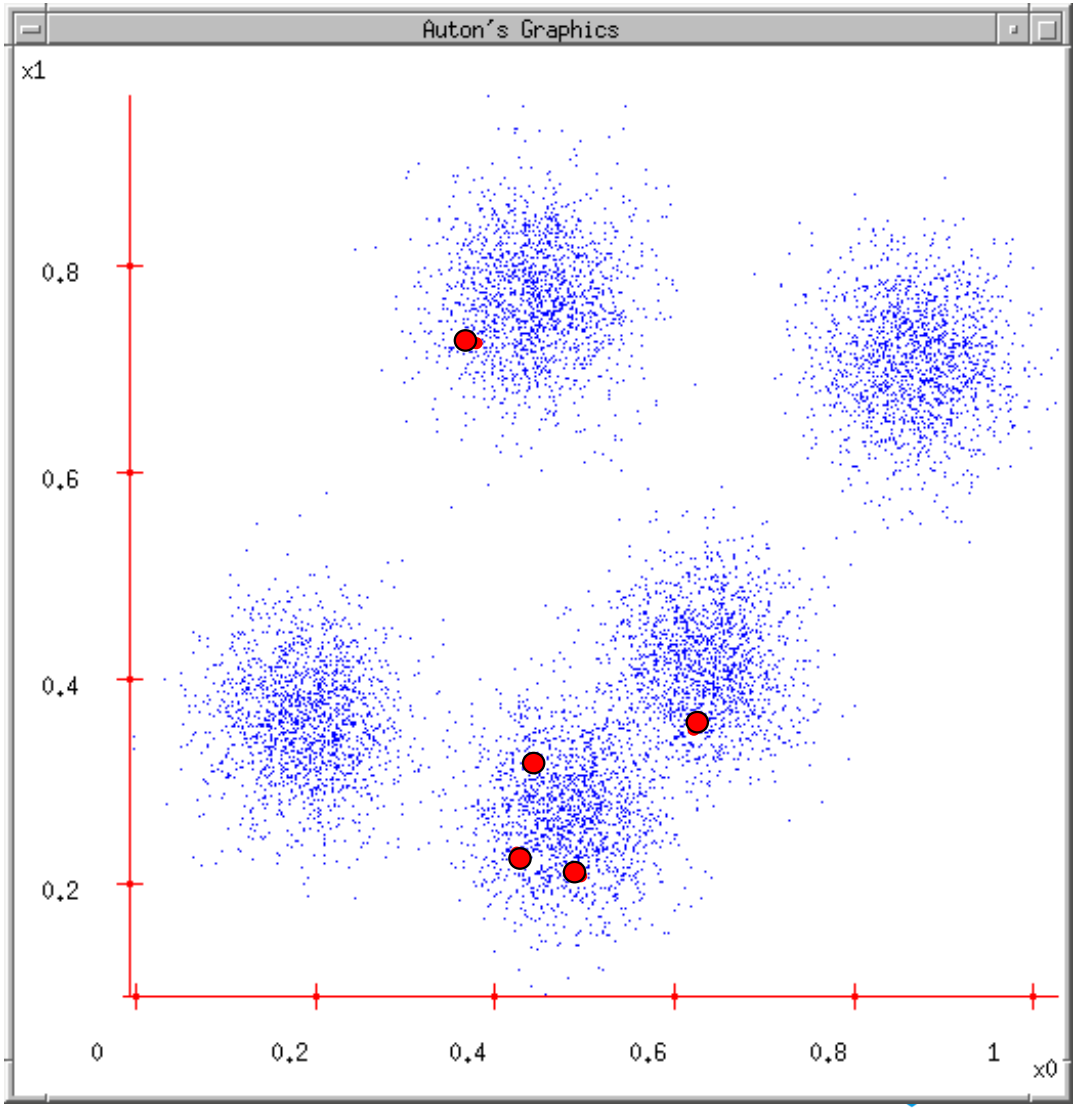
1. Ask user how many clusters they'd like. *(e.g. k=5)*



(Example is courtesy of Andrew Moore, CMU)

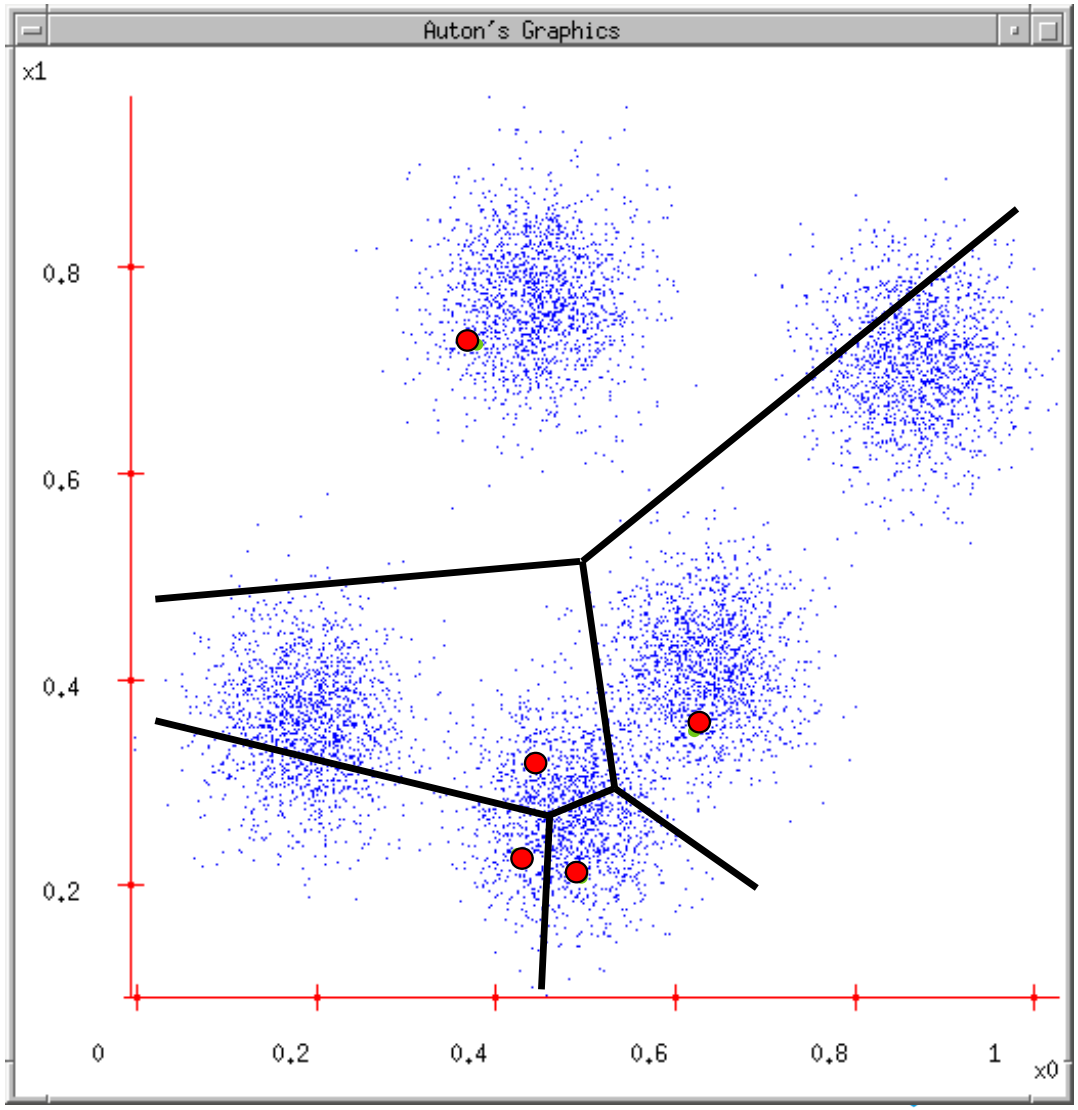
K-Means Algorithm

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



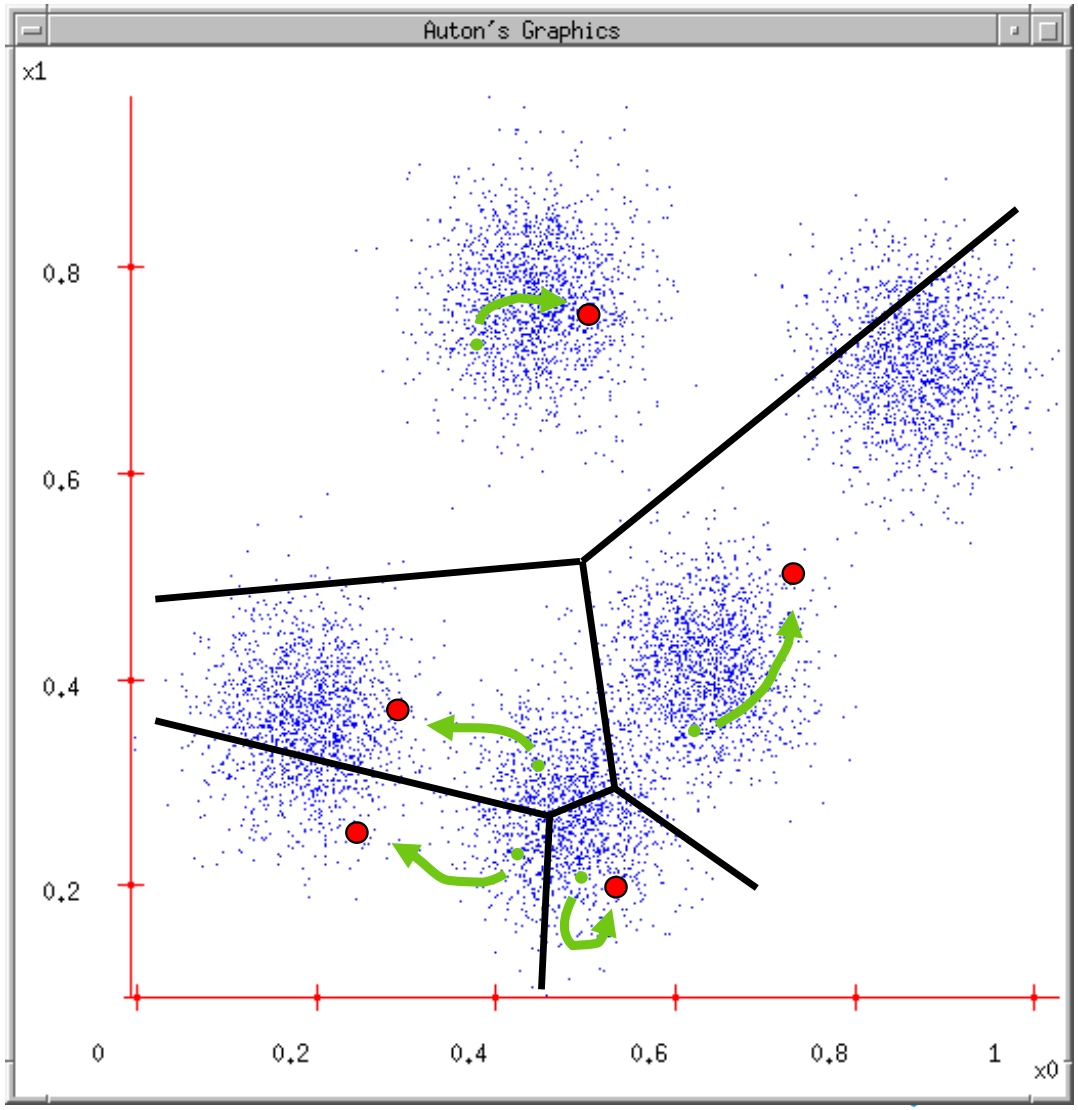
K-Means Algorithm

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.



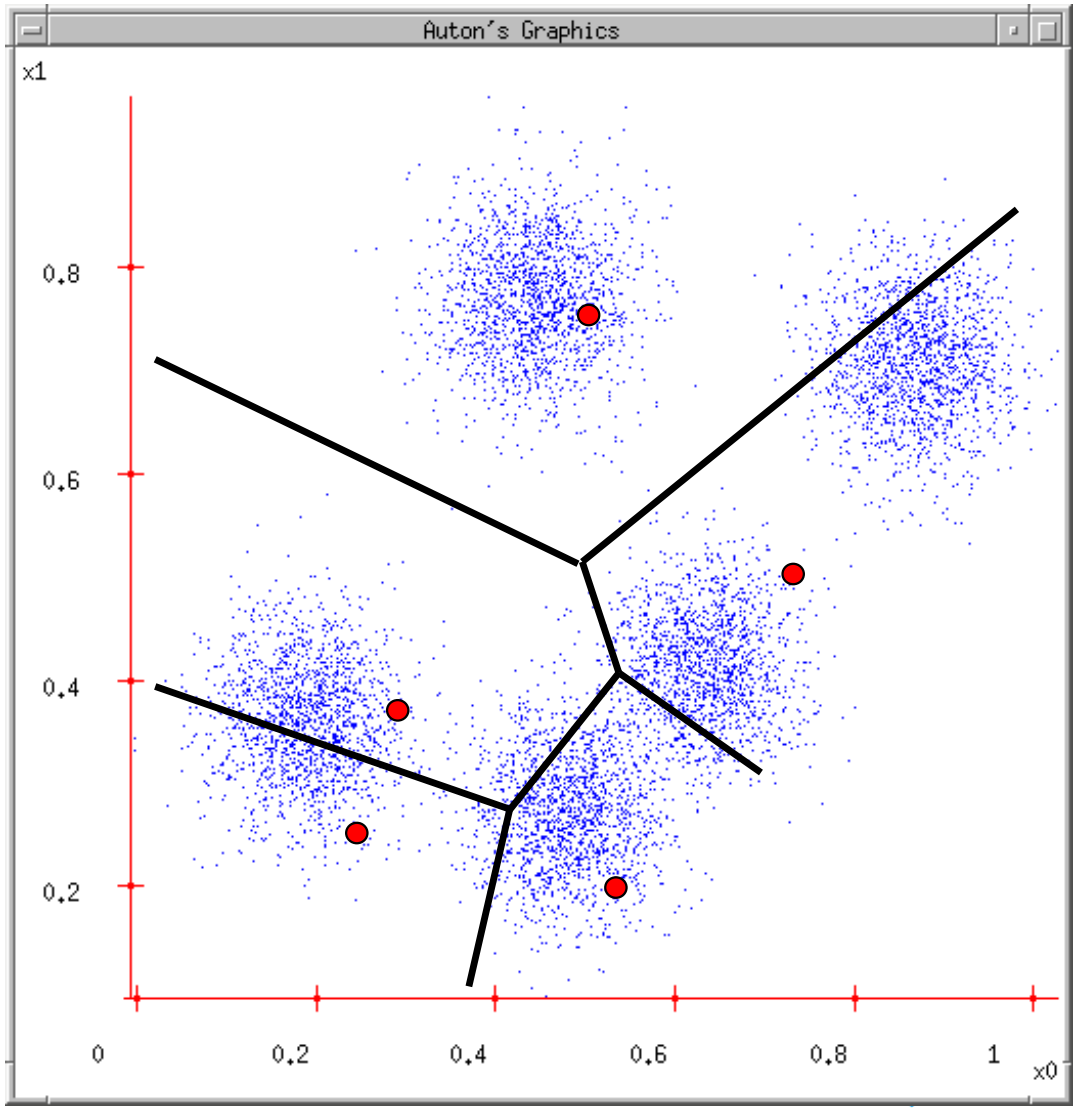
K-Means Algorithm

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns

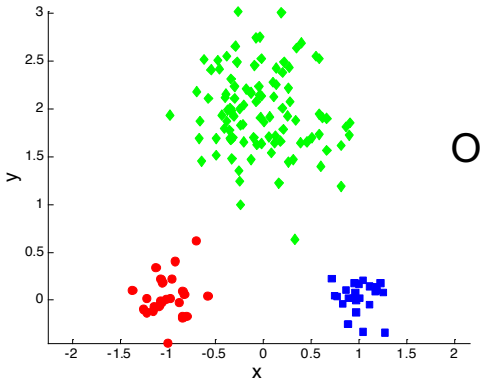


K-Means Algorithm

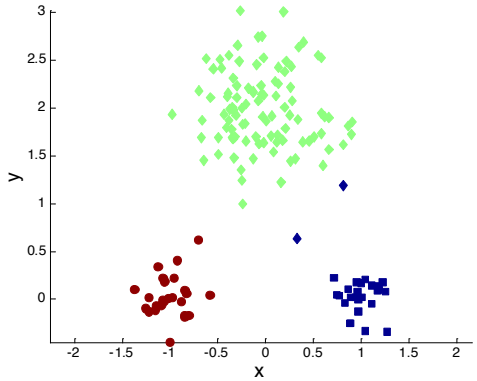
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns
5. New Centers => new boundaries
6. Repeat until no change



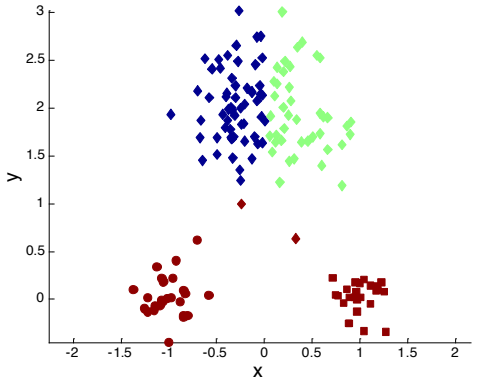
Different K-Means Clustering



Original Points

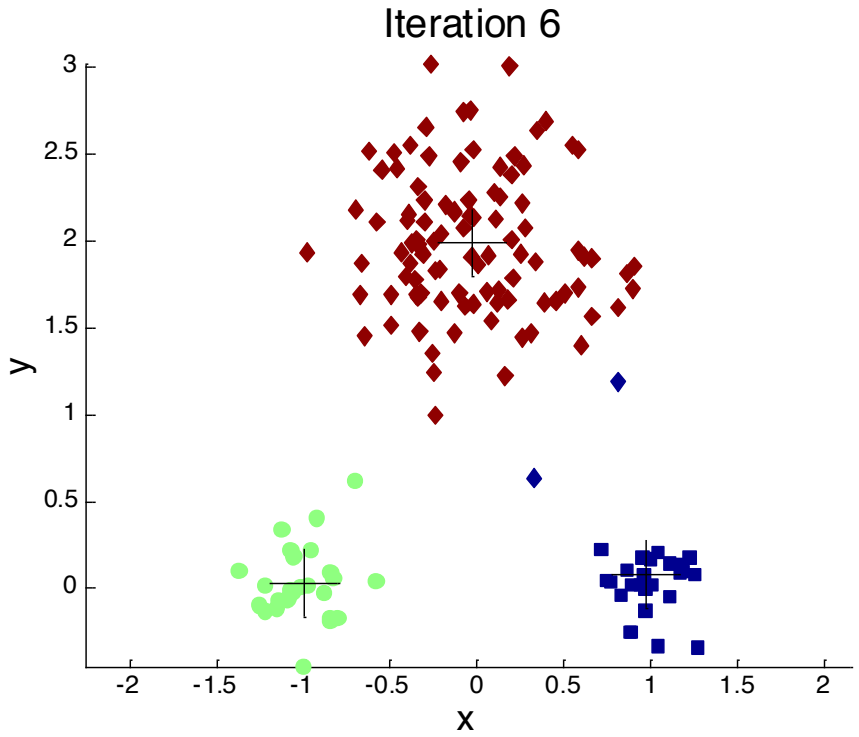


Optimal Clustering

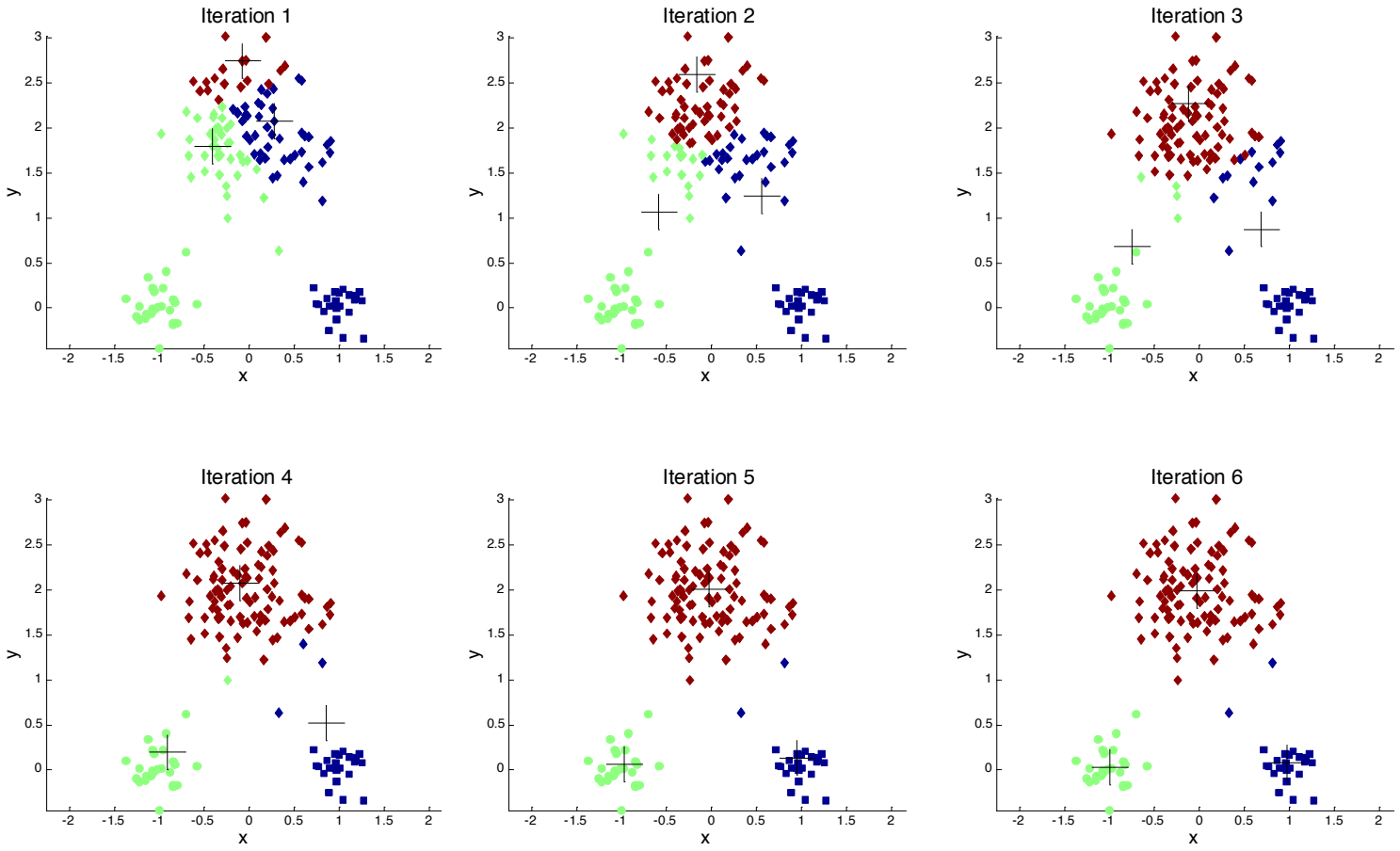


Sub-optimal Clustering

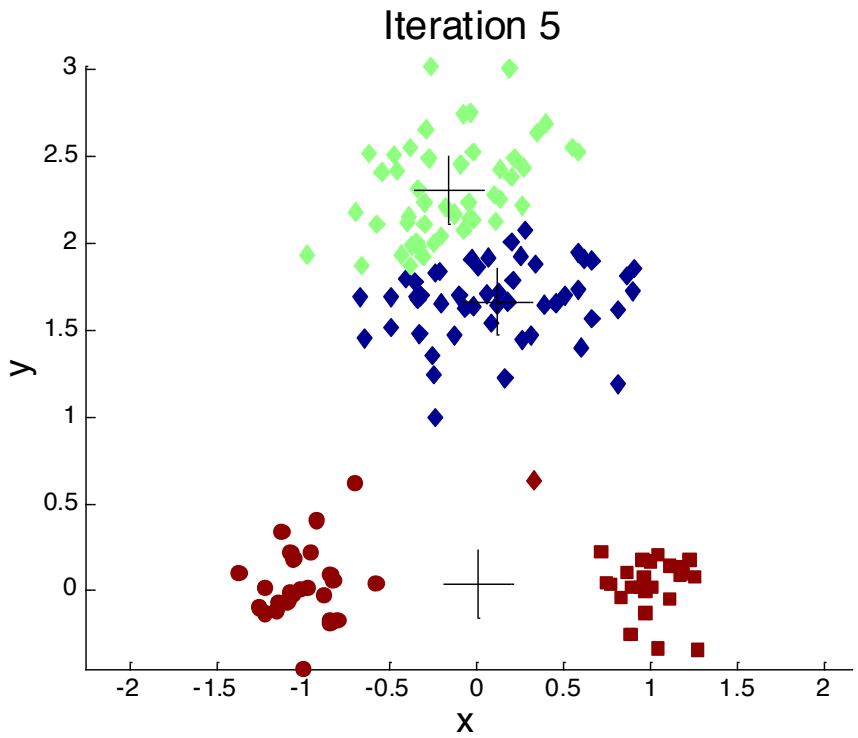
Importance of the Initial Centroids' Choice



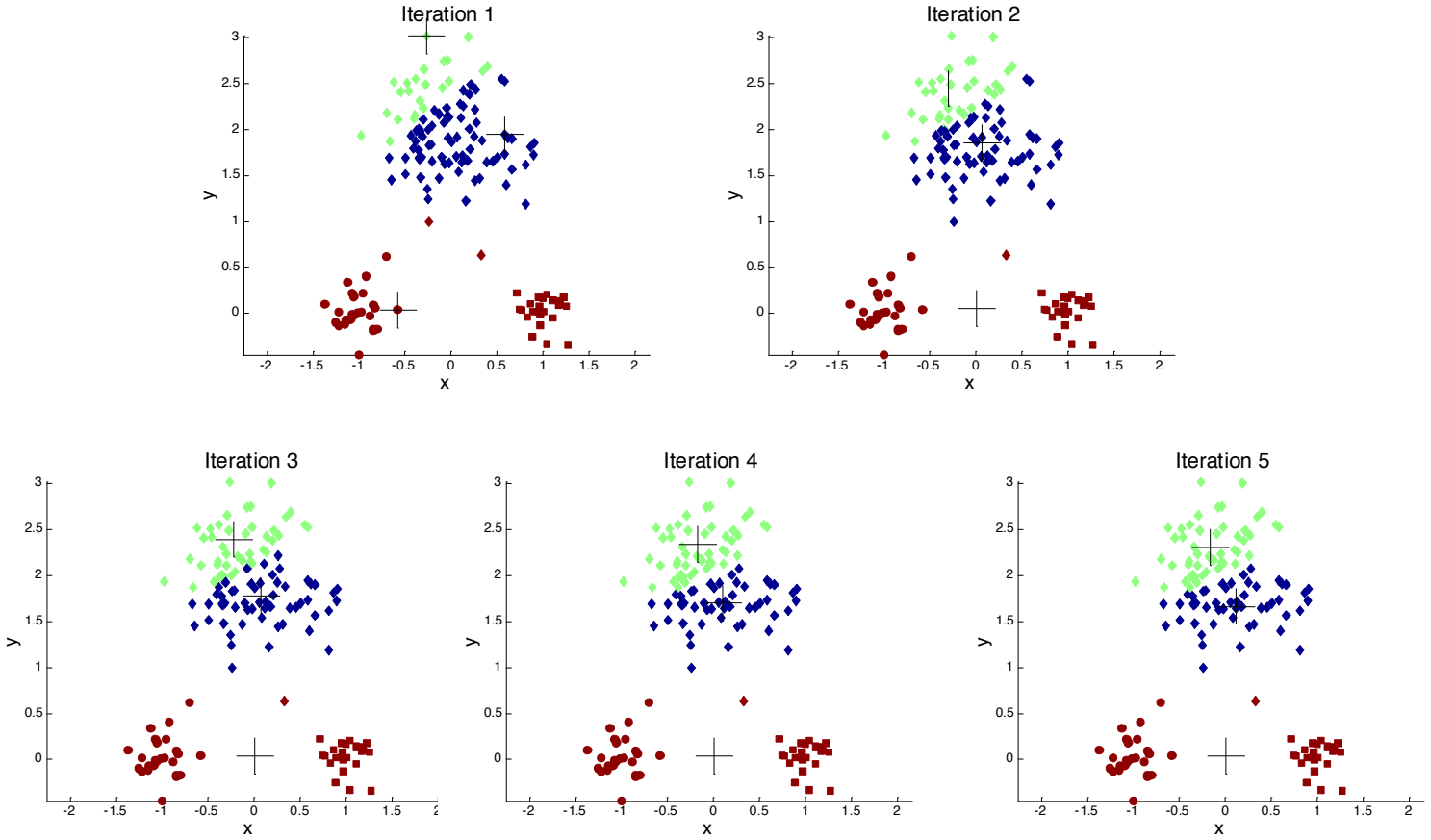
Importance of the Initial Centroids' Choice



Importance of the Initial Centroids' Choice



Importance of the Initial Centroids' Choice



Dealing with Initialization

- Do multiple runs and select the clustering with the smallest error
- Select original set of points by methods other than random. E.g., pick the most distant (from each other) points as cluster centers (K-means++ algorithm)

K-means Algorithm – Centroids

- The centroid depends on the distance function
  - The minimizer for the distance function
- 'Closeness' is measured by Euclidean distance (SSE), cosine similarity, correlation, etc.
- Centroid:
  - The mean of the points in the cluster for SSE, and cosine similarity
  - The median for Manhattan distance
- Finding the centroid is not always easy
  - It can be an NP-hard problem for some distance functions
    - E.g., median from multiple dimensions

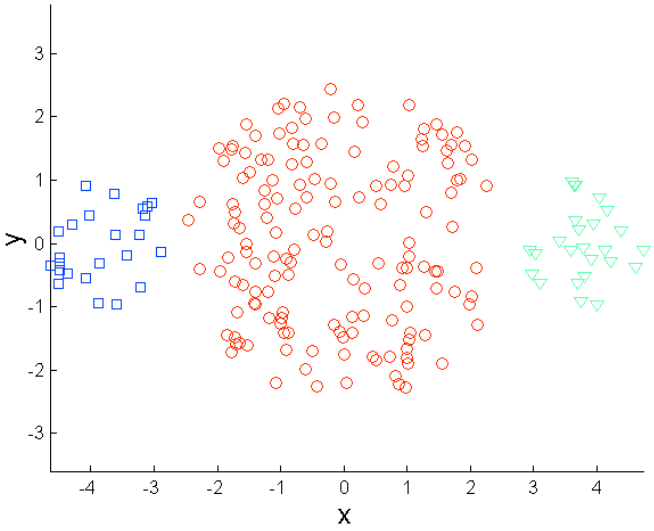
K-Means Algorithm – Convergence

- K-means will converge for common similarity measures
  - Most of the convergence happens in the first few iterations
  - Often the stopping condition is changed to “Until relatively few points change clusters”
- Complexity is $O(n.K.I.d)$
  - n = number of points, K = number of clusters, I = number of iterations, d = dimensionality
- In general, a fast and efficient algorithm

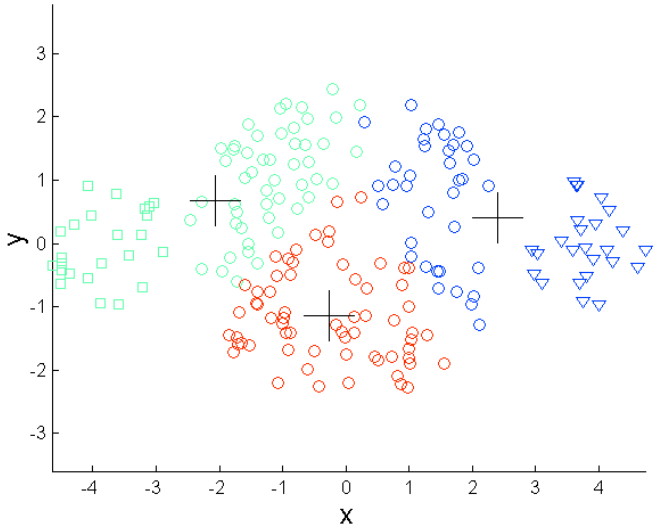
Limitations of K-means

- K-means has problems when clusters are of different
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers
- How to define the number of clusters?
  - (will deal with it in the Fuzzy Clustering section)

Limitations of K-Means: Differing Sizes

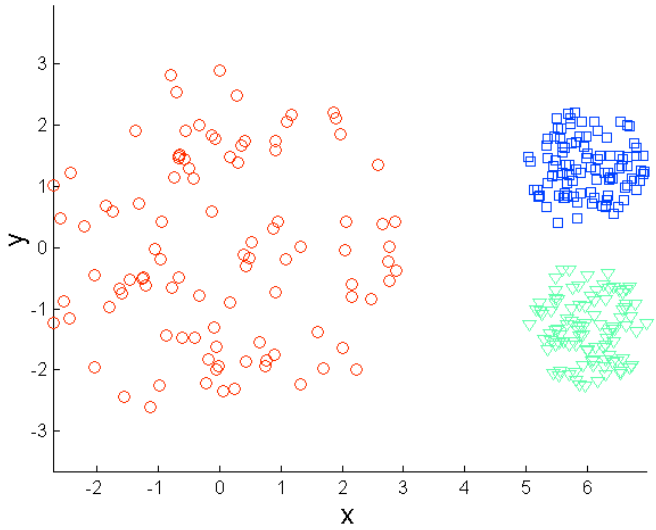


Original Points

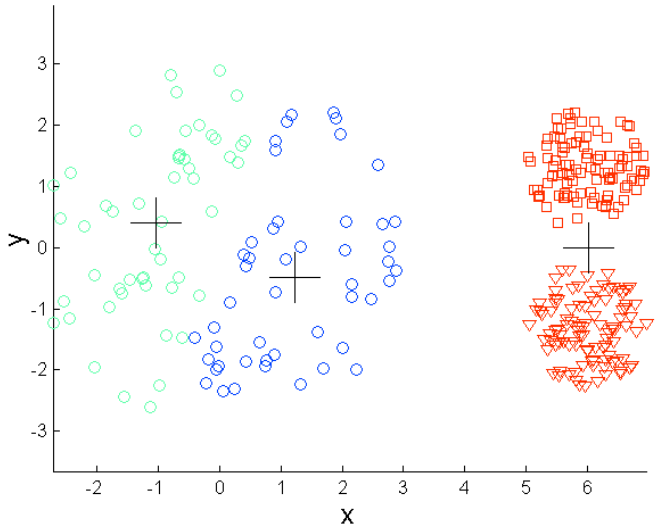


K-means (3 Clusters)

Limitations of K-Means: Differing Density

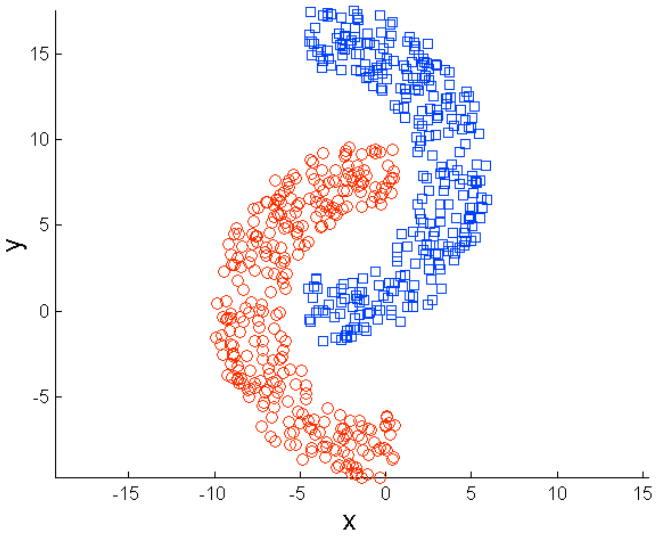


Original Points

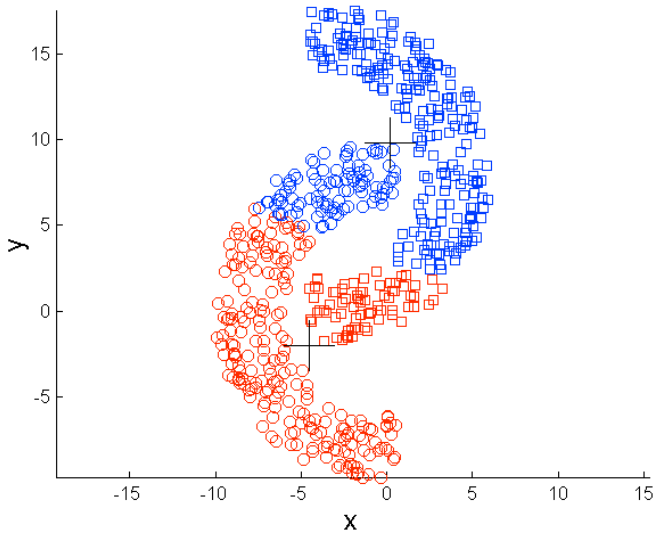


K-means (3 Clusters)

Limitations of K-Means: Non-Globular Shapes

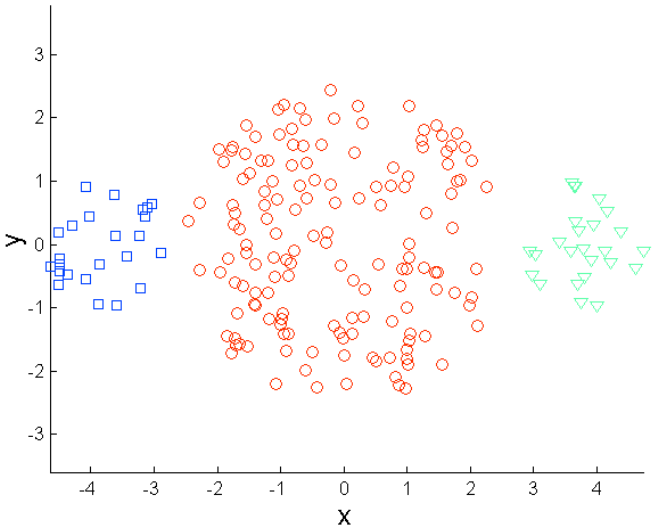


Original Points

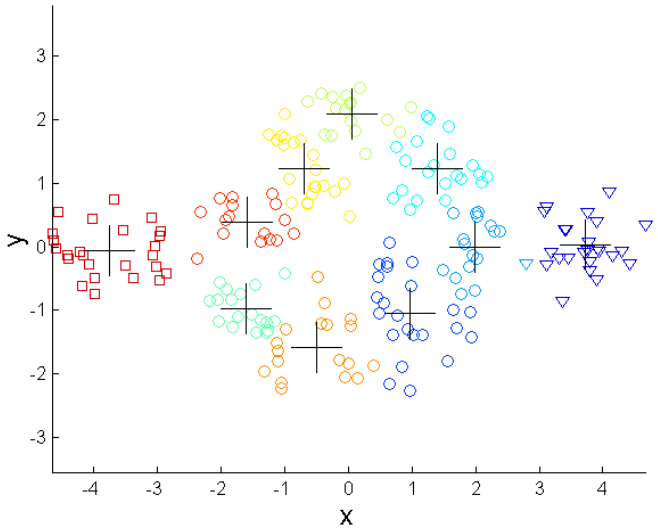


K-means (2 Clusters)

Overcoming K-Means Limitations



Original Points



K-means Clusters

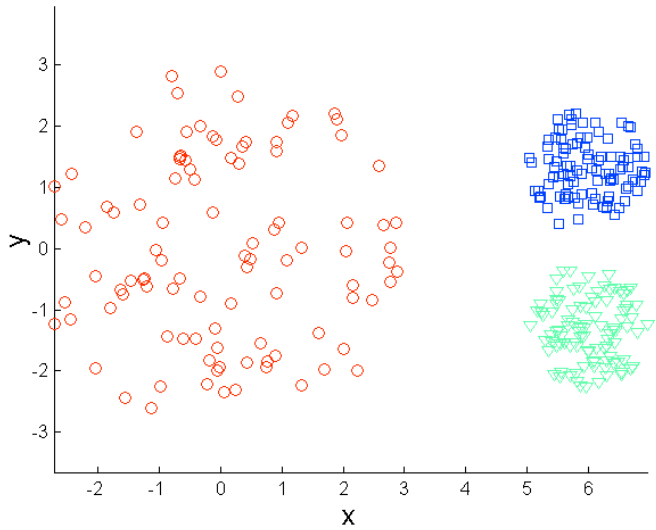
A solution is to use many clusters, and find parts of clusters… but joining the smaller clusters ain't trivial

Cluster Merging

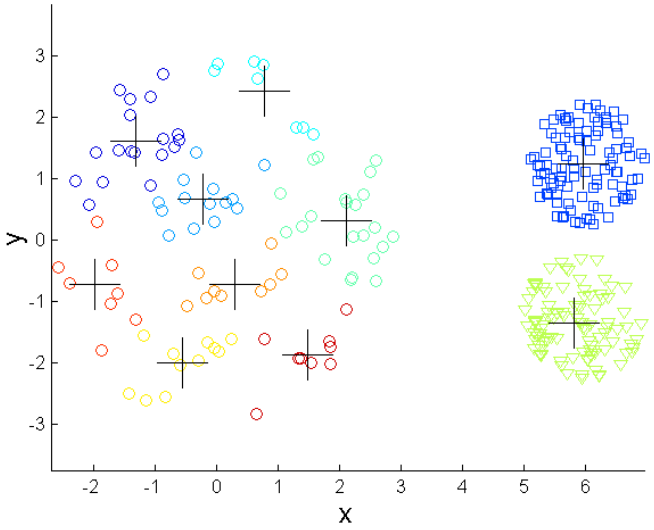
1. Select number of clusters larger than needed and do clustering
2. Merge clusters that are compatible
3. Cluster again and continue merging until there are no compatible clusters

- Cluster compatibility measured by:
  - Compatibility criteria
    - How close are clusters?
    - How similar are their characteristics?
    - Etc.
  - c.f. Similarity measures

Overcoming K-means Limitations

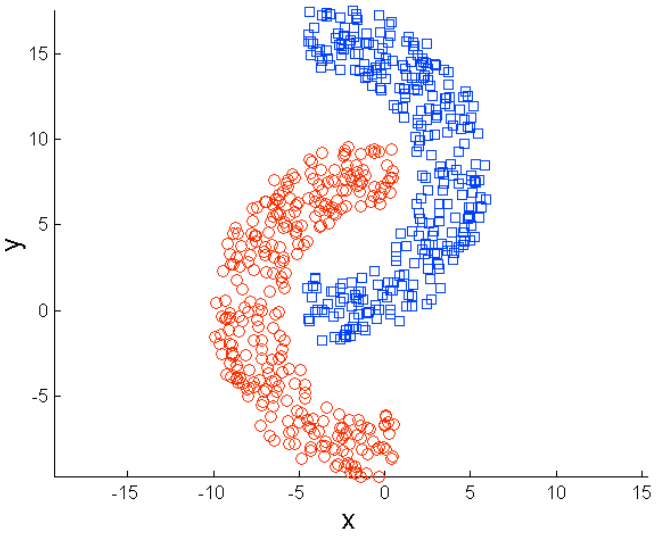


Original Points

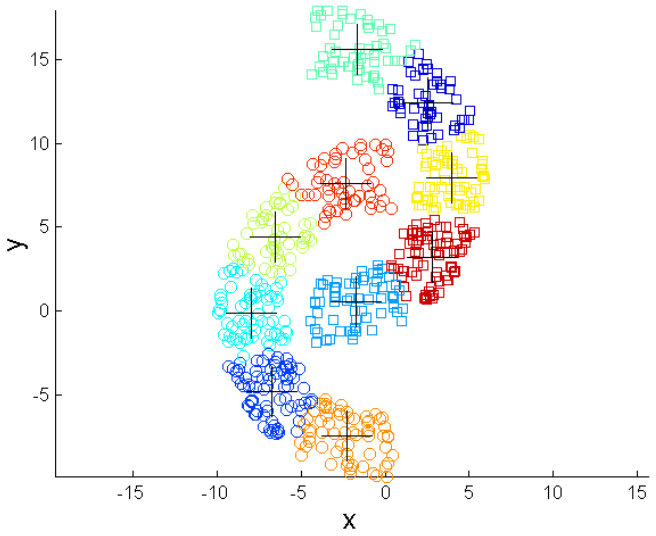


K-means Clusters

Overcoming K-means Limitations



Original Points



K-means Clusters

Variations

- K-medoids: Similar problem definition as in K-means, but the centroid of the cluster is defined to be one of the points in the cluster (the medoid)
- K-centers: Similar problem definition as in K-means, but the goal now is to minimize the maximum diameter of the clusters (diameter of a cluster is maximum distance between any two points in the cluster).

Fuzzy Clustering

Fuzzy C-Means
Probabilistic Clustering
GK Clustering

(Interlude) Let's generalize the Data into *n* dimensions…

- Pattern recognition terminology:
  - Columns are called patterns or objects
  - Rows are called the features or attributes

$$Z = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,N} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n,1} & z_{n,2} & \cdots & z_{n,N} \end{bmatrix}$$

In this example, there are *N* data points, each consisting of *n* attributes

Fuzzy Clustering: Fuzzy C-Means

- Partition data into overlapping sets based on similarity amongst patterns

Given the data $\mathbf{x}_k = [x_{1k}, x_{2k}, \ldots, x_{nk}]^T \in \Re^n, \quad k = 1, \ldots, N$

find the fuzzy partition matrix

$$\mathbf{U} = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1N} \\ \vdots & \ddots & \vdots \\ \mu_{C1} & \cdots & \mu_{CN} \end{bmatrix}, \quad \mu_{ij} \in [0,1]$$

and the cluster centres $\mathbf{V} = \{\mathbf{v}_1, \ldots \mathbf{v}_C\}, \quad \mathbf{v}_i \in \Re^n$

that minimize objective function

$$J(\mathbf{X}, \mathbf{U}, \mathbf{V}) = \sum_{i=1}^{C} \sum_{k=1}^{N} \mu_{ik}^{m} d^2(\mathbf{x}_k, \mathbf{v}_i)$$

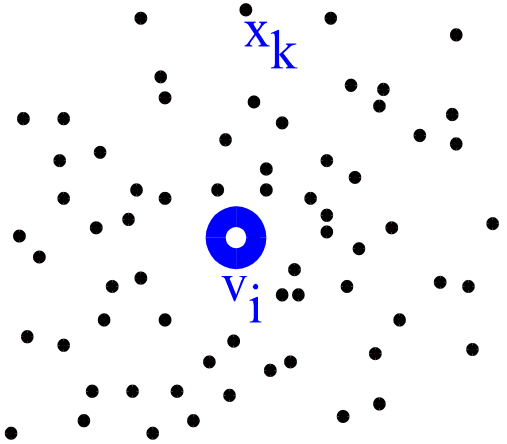
Divides N objects into C (overlapping) groups (clusters), indicating the degree of membership of each point to each cluster

This is a generalization of hard *k*-means!

Fuzzy Clustering: Fuzzy C-Means (II)

- Minimise objective function

$$J(\mathbf{X},\mathbf{U},\mathbf{V}) = \sum_{i=1}^{C}\sum_{k=1}^{N} \mu_{ik}^{m} d^2(\mathbf{x}_k,\mathbf{v}_i)$$



  - subject to:

$$0 \le \mu_{ik} \le 1, \quad i = 1,\ldots,C,\ k = 1,\ldots,N$$ membership degree

$$\sum_{i=1}^{C} \mu_{ik} = 1, \quad k = 1,\ldots,N$$ total membership

$$0 < \sum_{k=1}^{N} \mu_{ik} < N, \quad i = 1,\ldots,C$$ no cluster empty

$m \in (1,\infty)$ is the fuzziness parameter; usually m=2

Fuzzy C-Means Algorithm

- Repeat:
  1. Compute cluster centers

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} \mu_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{N} \mu_{ik}^m}$$

Assumes partition matrix is fixed

  2. Calculate distances

$$d_{ik}^2 = (\mathbf{x}_k - \mathbf{v}_i)^T (\mathbf{x}_k - \mathbf{v}_i)$$

  3. Update partition matrix

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{C} (d_{ik}^2 / d_{jk}^2)^{1/(m-1)}}$$

Assumes cluster centers are fixed

until $\|\Delta \mathbf{U}\| < \varepsilon$

(other stopping criteria are possible)

Initialization can be done either by initializing V or by initializing U

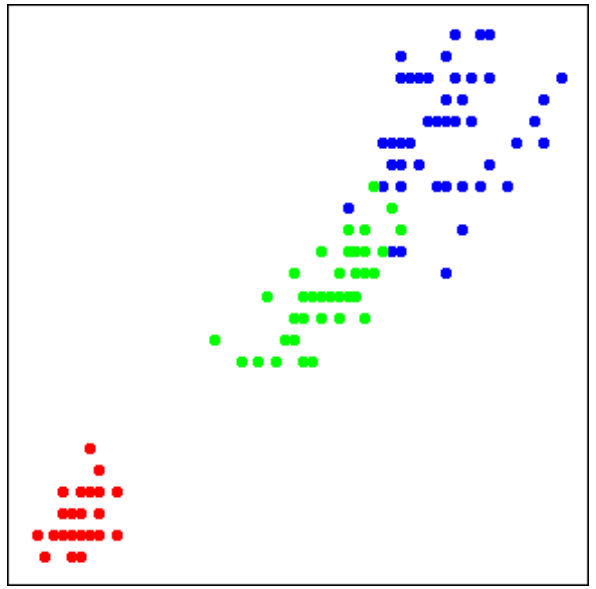
Example: The Iris Data

- © Iris Species Database http://www.badbear.com/signa/
  - Collected by Ronald Aylmer Fischer (famous statistician)
  - 150 cases in total, 50 cases per Iris flower type
  - Measurements: sepal length/width, petal length/width (cm)
- Probably the most famous dataset in pattern recognition and data analysis

Iris setosa

Iris versicolor

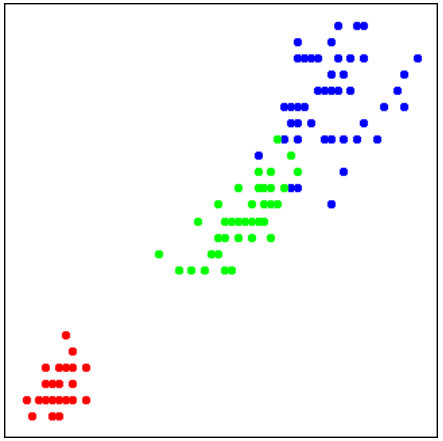
Iris virginica

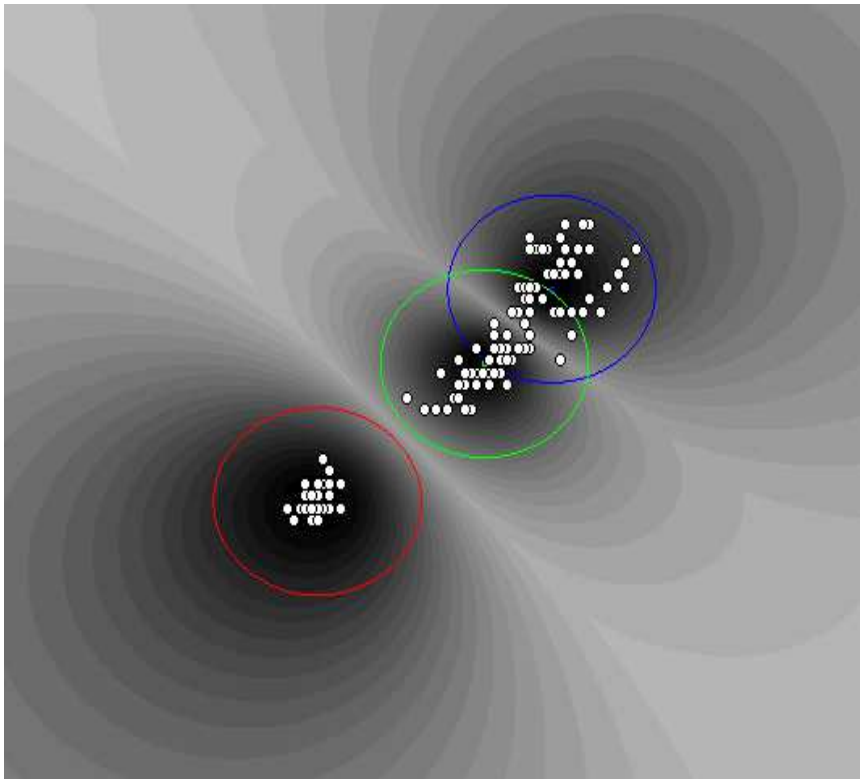


Iris setosa (red)
Iris versicolor (green)
Iris virginica (blue)
(Shown: sepal length and petal length)

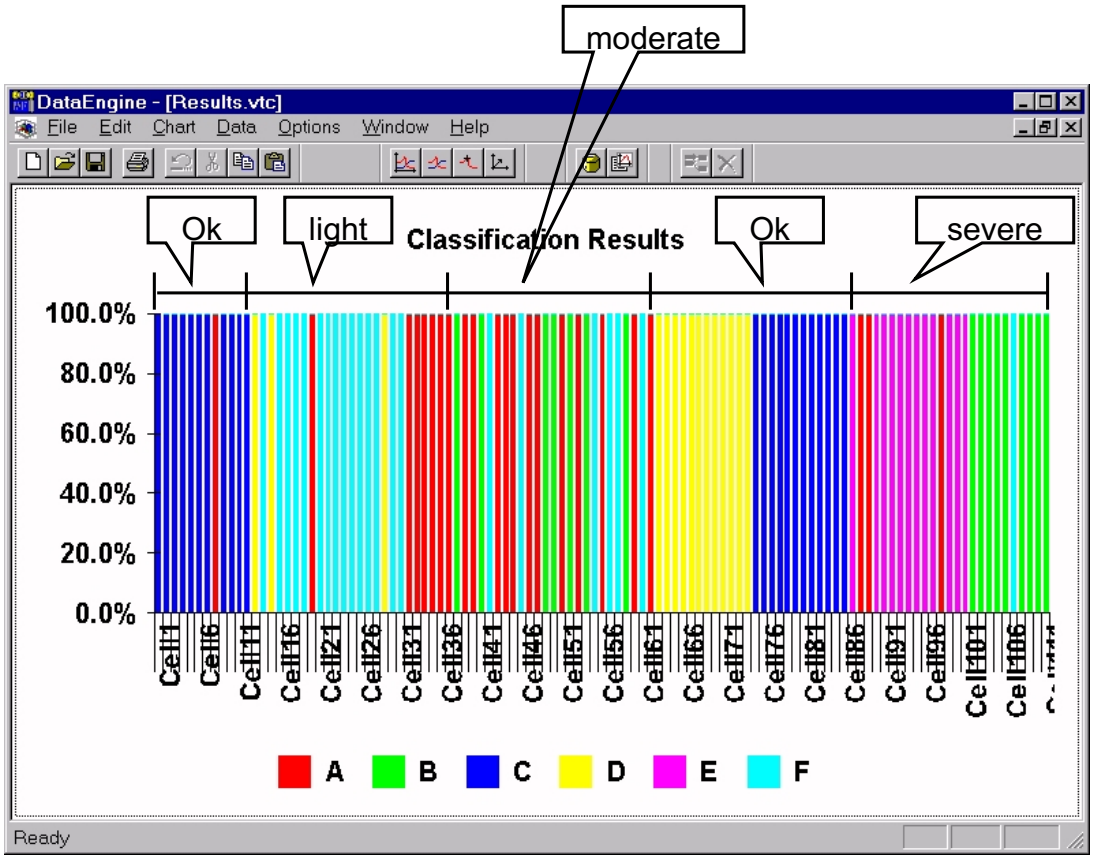
Example: The Iris Data – Fuzzy C-Means

Iris setosa (red), Iris versicolor (green), Iris virginica (blue)
(Shown: sepal length and petal length)



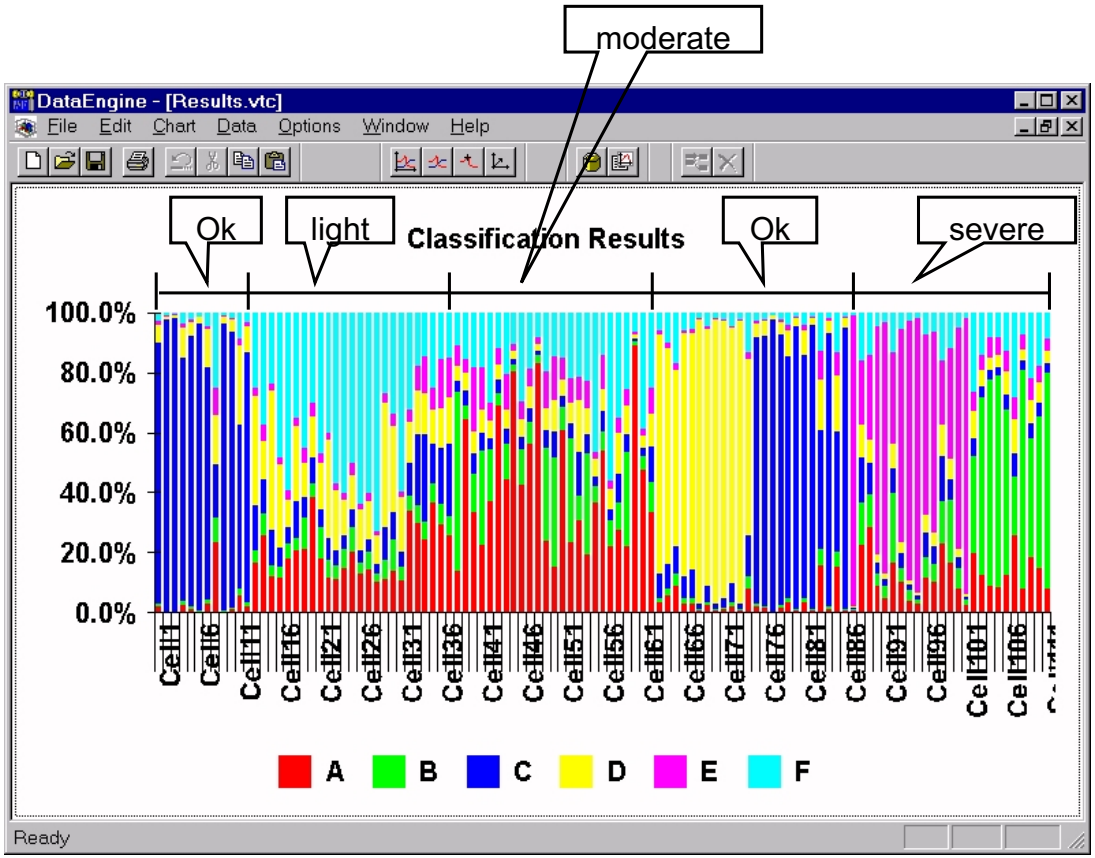


Example: K-Means vs. Fuzzy C-Means – Cancer Cells Hard Clustering classifier



A cell is either one or the other class defined by a colour

Example: K-Means vs. Fuzzy C-Means – Cancer Cells Fuzzy Clustering Classifier



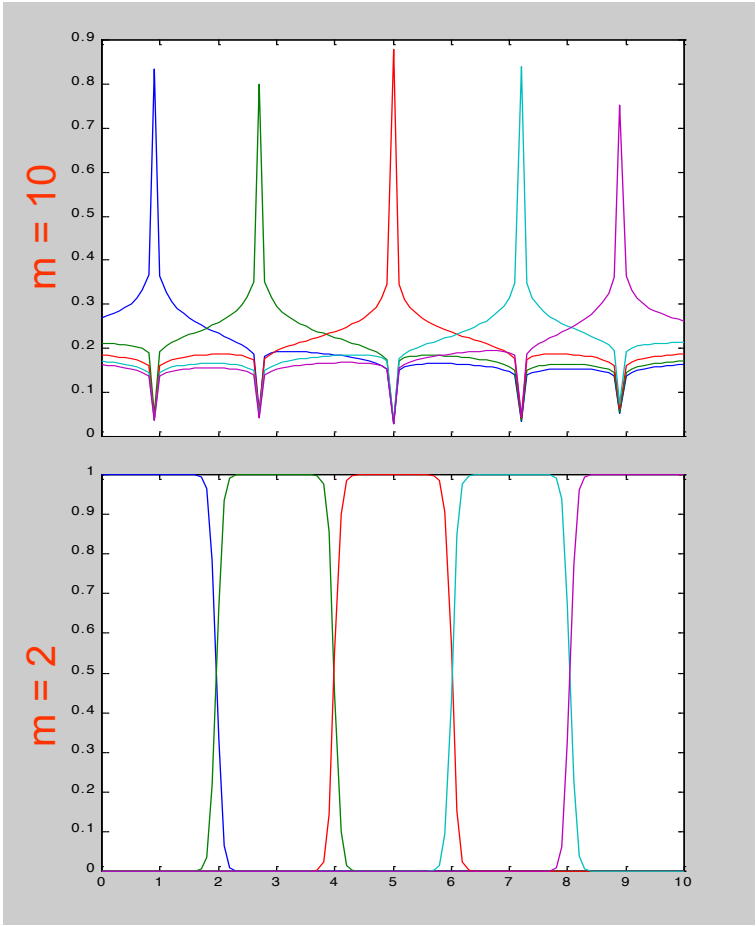
A cell can belong to several classes to a Degree, i.e., one column may have several colours

Initialization

- How to avoid local minima during the optimization?
  - Fuzzy C-Means is quite robust to initialization!
- Therefore:
  - Randomly select a set of cluster prototypes $\mathbf{V}$
  - Randomly initialize the partition matrix $\mathbf{U}$
- Improving the intialization:
  - Use information (e.g. cluster center locations) from a separate clustering step
  - Initialize centers far away from data

Effect of the Fuzziness Index m

- As m increases, clusters overlap more; their centers become more isolated
- As m decreases, clusters overlap less; they become crisp
- Often m=2 is selected
- With m=1, we have hard clustering



Cluster Validity

- Q: How good are the clustering results?
  - Desirable: summarize information by a single criterion indicating how well a data point is classified by clustering
  - Note: Each data point has c memberships
- Criteria for definition of “optimal partition” based on:
  - Clear separation between resulting clusters
  - Minimal volume of clusters
  - Maximal number of points concentrated close to cluster centroid
- Cluster validity: average of each criteria over the entire data set
  - Note: “Good” clusters are actually not very fuzzy!
- Cluster validity measures (CVI) try to quantify these criteria

CVI (Cluster Validity Index)

- Dozens of CVI have been proposed
  - None can be said to be better than others
- Best procedure:
  - Select several CVI
  - Compare results
  - Try to use consensus
- CVI examples:

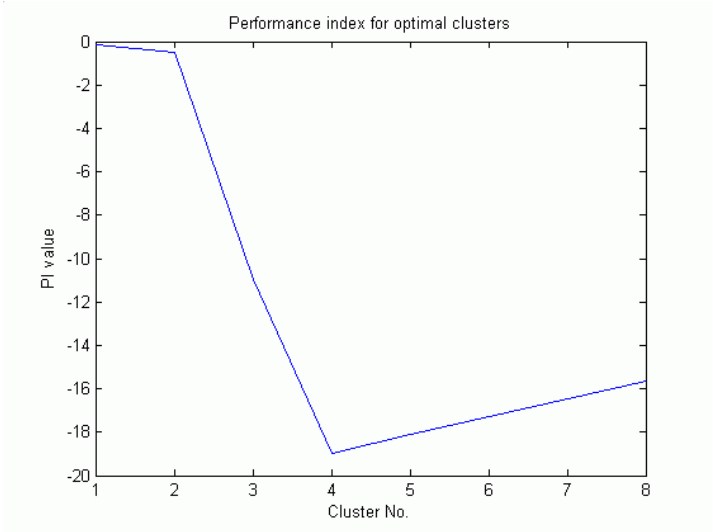
Gath and Geva index

$$S_G = \sum_{i=1}^{C} \sqrt{\left| \frac{\sum_{k=1}^{N} (\mu_{ik})^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m} \right|} + \beta C$$

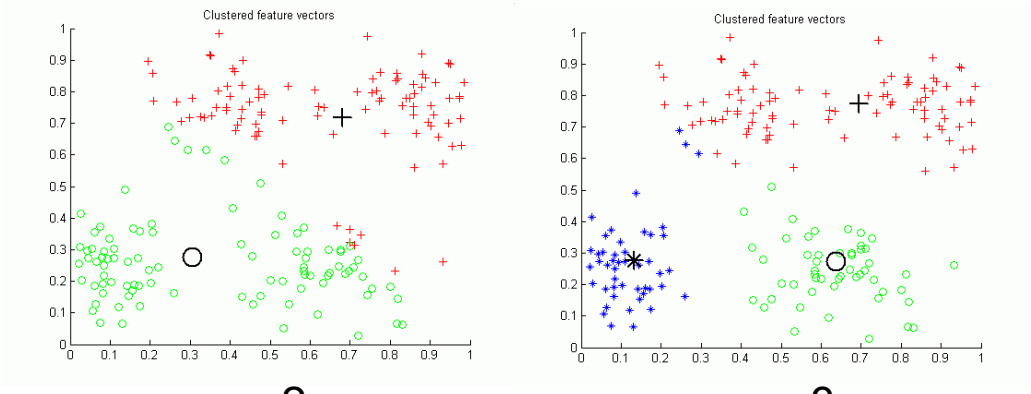
Xie-Beni index

$$S_X = \frac{\sum_{i=1}^{C} \sum_{k=1}^{N} \mu_{ik}^m d^2(\mathbf{x}_k, \mathbf{v}_i)}{N\left(\min_{i,j,i \neq j} d^2(\mathbf{v}_i - \mathbf{v}_j) \right)}$$

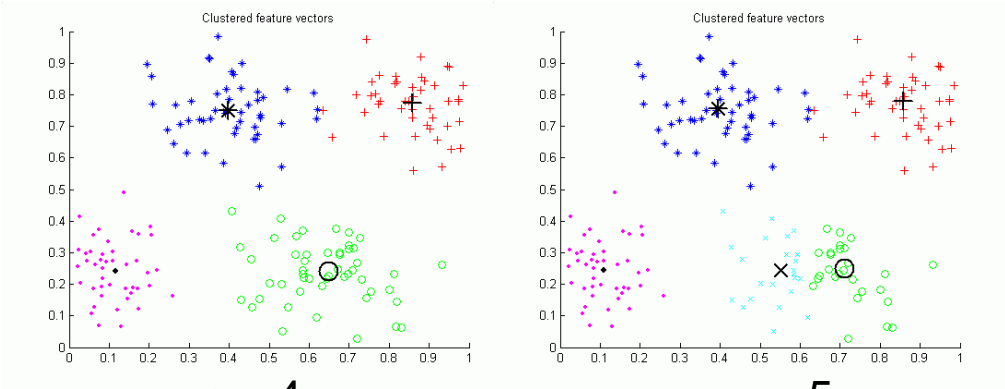
Cluster Validity Example



CVI
(is minimum for c = 4)



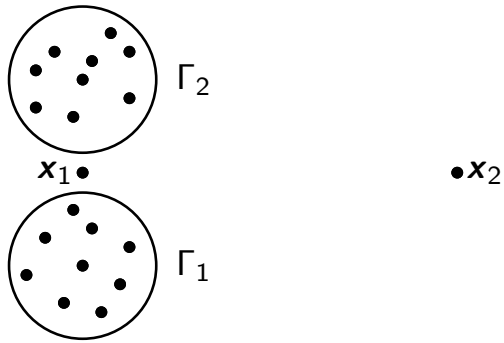
c = 2 c = 3



c = 4 c = 5

Fuzzy C-Means: Discussion

- ☺ It is initialized with randomly placed cluster centers
- ☺ Is stable and robust. Compared to hard *K*-means, it is:
  - quite insensitive to initialization;
  - not likely to get stuck in local minimum.
- ☺ FCM converges in a saddle point or minimum
- ☹ The normalization of memberships is a problem for noise and outliers:
  - Noisy data contribute as much as any other point even though they are a large distance from the bulk of the data
    - E.g.: x_1 and x_2 have the same membership degrees assigned to each cluster
  - Possibilistic C-Means avoids this issue



Fuzzy C-Means “Made Easy”…

- Use the Force, Luke...
- There are numerous libraries implementing Fuzzy C-Means
- E.g: Scikit-fuzzy
  - https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_cmeans.html#example-plot-cmeans-py

Possibilistic Fuzzy Clustering

- Minimise objective function

$$J(\mathbf{X},\mathbf{U},\mathbf{V},\boldsymbol{\eta}) = \sum_{i=1}^{C}\sum_{k=1}^{N} \mu_{ik}^{m} d^2(\mathbf{x}_k,\mathbf{v}_i) + \sum_{i=1}^{C} \eta_i \sum_{k=1}^{N} (1-\mu_{ik})^m$$

  - where:

    $m \in (1,\infty)$ is the fuzziness parameter

    η determines the size of the clusters
    (suitable values taken from average inter-cluster distance)

$$\eta_i = \frac{\sum_{k=1}^{N} \mu_{ik}^{m} d_{ik}^2}{\sum_{k=1}^{N} \mu_{ik}^{m}}$$

The optimization problem can now be decomposed into *C* independent optimization problems

PFCM: Possibilistic Clustering Algorithm

- Repeat:
  1. Compute cluster centers

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} \mu_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{N} \mu_{ik}^m}$$

Assumes partition matrix is fixed

  2. Calculate distances

$$d_{ik}^2 = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A} (\mathbf{x}_k - \mathbf{v}_i)$$

  3. Update partition matrix

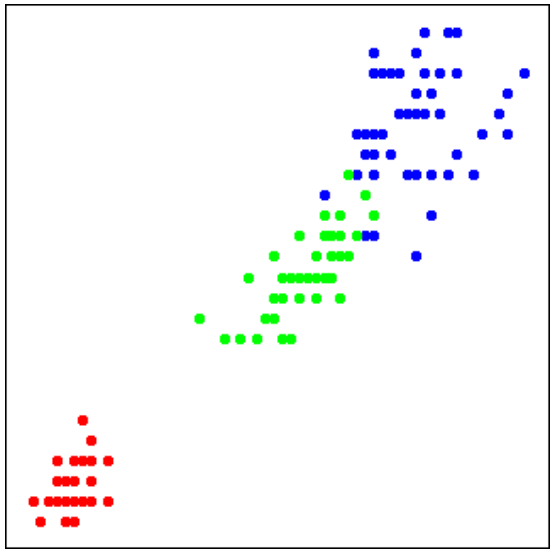
$$\mu_{ik} = \frac{1}{1 + \left(\frac{d_{ik}^2}{\eta_i^2} \right)^{\frac{1}{m-1}}}$$

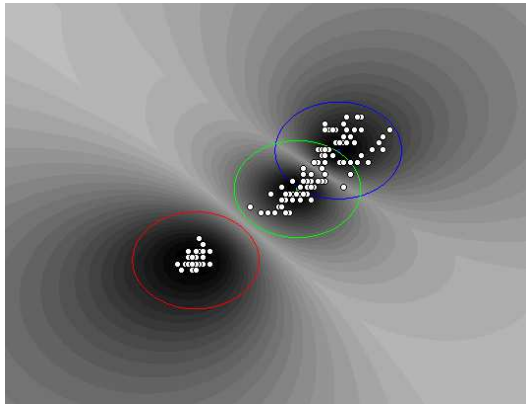
Membership value does not depend on the membership to other clusters (sum of the membership to each cluster no longer needs to be 1)

until $\|\Delta \mathbf{U}\| < \varepsilon$

PFCM vs. FCM: Iris Data Example

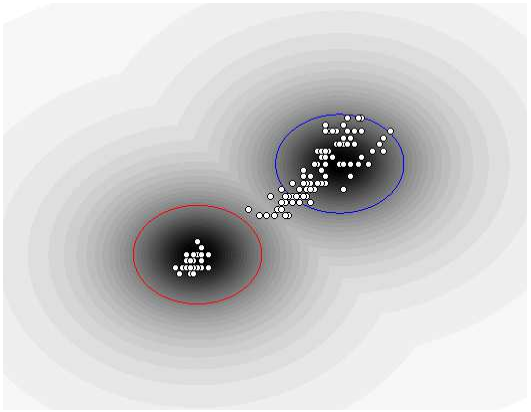
Iris setosa (red), Iris versicolor (green), Iris virginica (blue)
(Shown: sepal length and petal length)





FCM divides space (separates data into 3 clusters)

PCM depends on typicality to closest clusters (separates data into 2 clusters)



PCM vs. FCM

Characteristic	FCM	PCM
Data Partition	Enforced	Not Enforced
Membership Degree	Distributed	Determined by data
Cluster Interaction	Covers whole data	Doesn't
Intra-Cluster Distribution	High	Low
Cluster Number C	Enforced	Upper bound

PCM: Final Remarks

- PCM tends to interpret low membership data as outliers
- Cluster coincidence:
  - PCM clusters can coincide and might not even cover the whole data
- To obtain better coverage:
  - use FCM to initialize PCM (i.e. prototypes, η_i, C);
  - after 1st PCM run, re-estimate η_i again;
  - use improved estimates for 2nd PCM run as final solution.

Gustafson-Kessel Clustering

- Clusters adapt themselves to the shape and location of data
- Clusters are constrained by volume
- Uses an adaptive distance metric (Malahanobis)

$$d^2(\mathbf{x}_k - \mathbf{v}_i) = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A}_i (\mathbf{x}_k - \mathbf{v}_i)$$

$$\mathbf{A}_i = \left|\mathbf{F}_i\right|^{1/n} \mathbf{F}_i^{-1}$$

- Fuzzy covariance matrix

$$\mathbf{F}_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m}$$

GK Algorithm

- Repeat:
  1. Compute cluster centers

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} \mu_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{N} \mu_{ik}^m}$$

Assumes partition matrix is fixed

  2. Calculate distances and covariance matrices

$$d_{ik}^2 = |\mathbf{F}_i|^{1/n} (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{F}_i^{-1} (\mathbf{x}_k - \mathbf{v}_i) \qquad \mathbf{F}_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m}$$

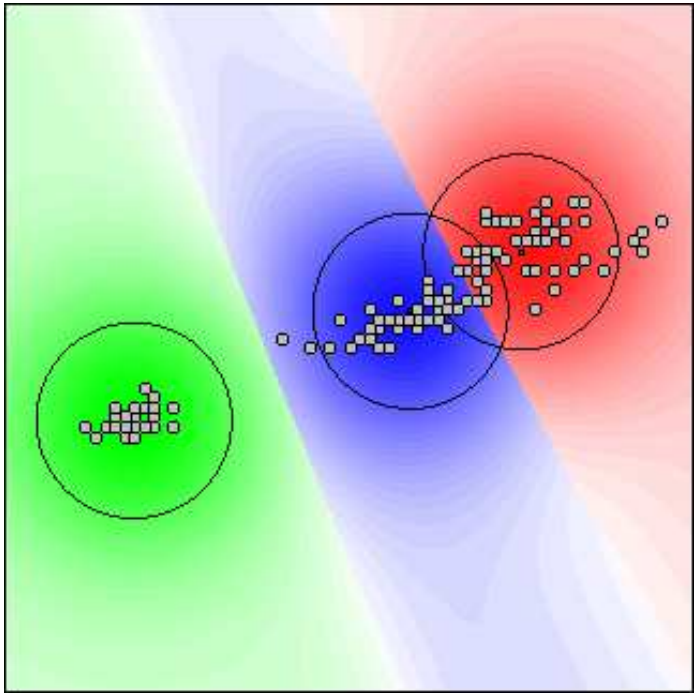
  3. Update partition matrix

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{C} (d_{ik}^2 / d_{jk}^2)^{1/(m-1)}}$$

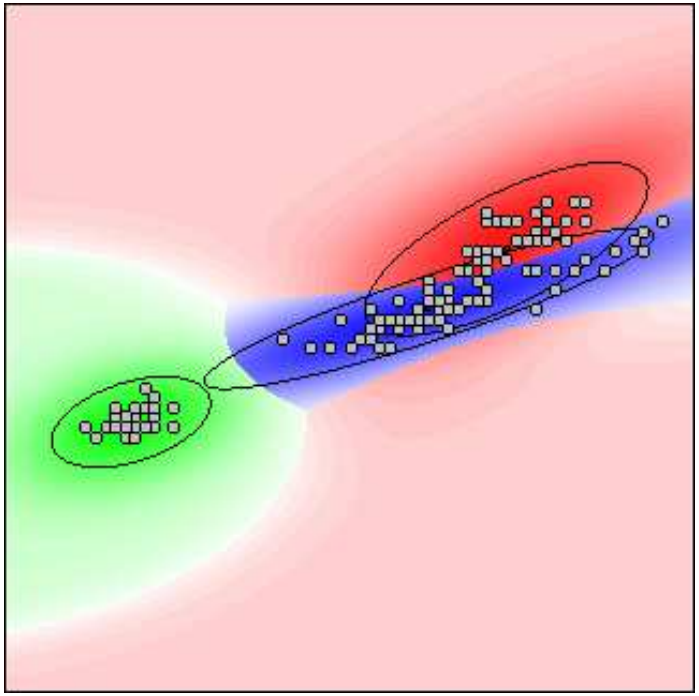
Assumes cluster centers are fixed

until $\|\Delta \mathbf{U}\| < \varepsilon$

GK Algorithm Example (Iris Data, 2 clusters)



Fuzzy C-Means



Fuzzy GK

Gustafson-Kessel discussion

- GK extracts more information than standard FCM and PCM
- More sensitive to initialization
  - Recommended initialization: few runs of FCM or PCM
- When compared to FCM or PCM, GK is:
  - Hard to apply to huge datasets
  - Computationally more intensive due to matrix inversions
    - Restriction to axis-parallel clusters reduces computational costs

Distance Variants

- Euclidean norm:

$$d^2(\mathbf{x}_k, \mathbf{v}_i) = (\mathbf{x}_k - \mathbf{v}_i)^T (\mathbf{x}_k - \mathbf{v}_i)$$

Spherical clusters only!

- Inner-product norm:

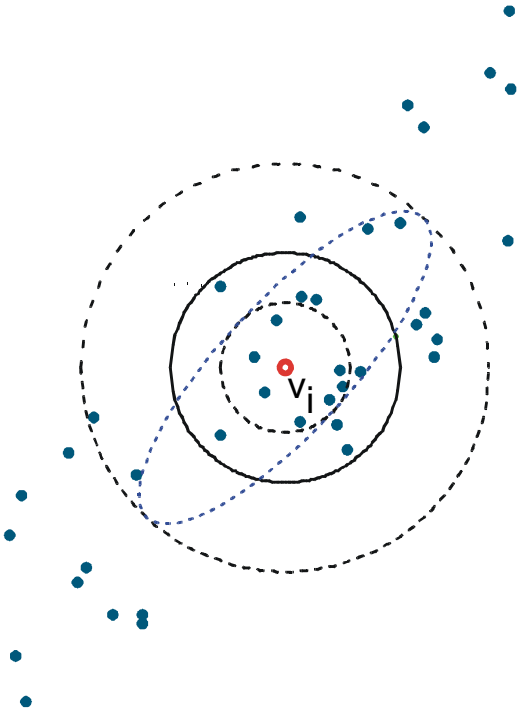
$$d^2(\mathbf{x}_k, \mathbf{v}_i) = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A} (\mathbf{x}_k - \mathbf{v}_i)$$

A is diagonal

- Mahalanobis norm:

$$d^2(\mathbf{x}_k, \mathbf{v}_i) = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{F}_i^{-1} (\mathbf{x}_k - \mathbf{v}_i)$$

Rotated clusters



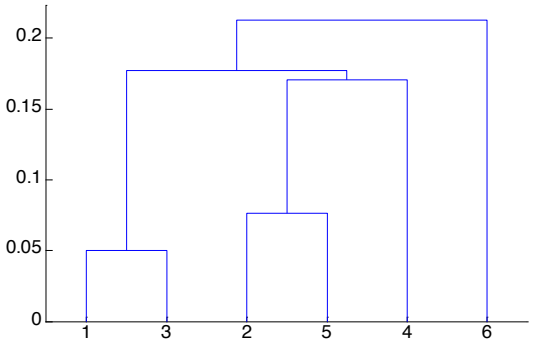
Hierarchical Clustering

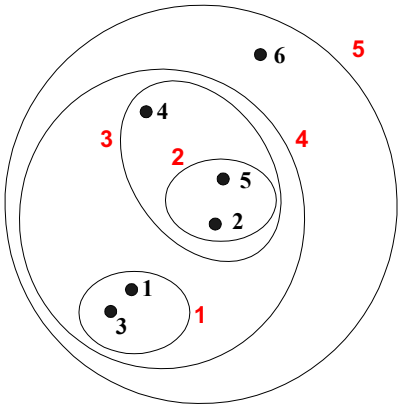
Hierarchical Clustering

- Two main types of 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) is 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)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

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





Strengths of Hierarchical Clustering

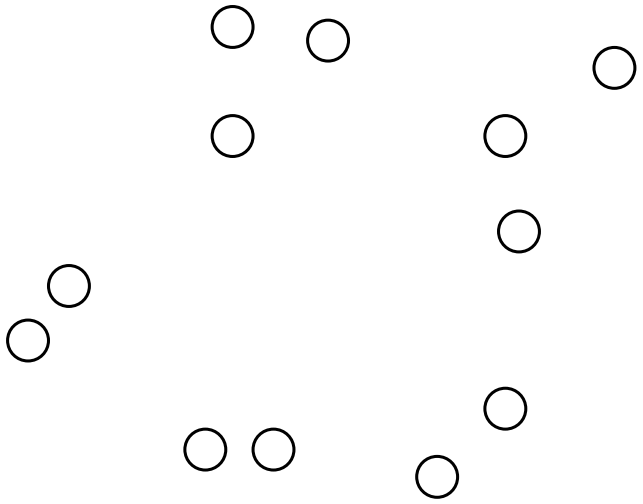
- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward:
  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
- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

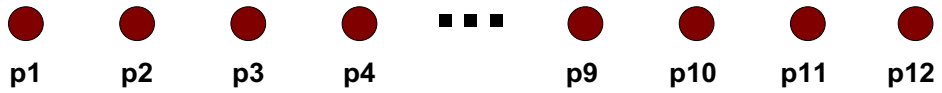
Agglomerative Clustering Example: Starting Situation

- Start with clusters of individual points and a proximity matrix



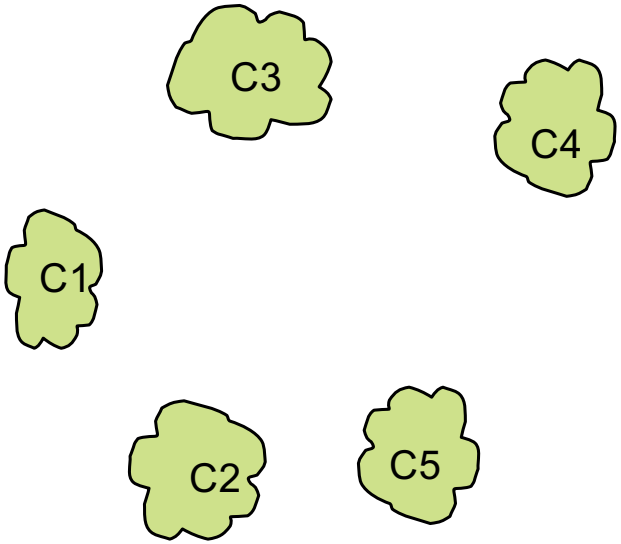
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix



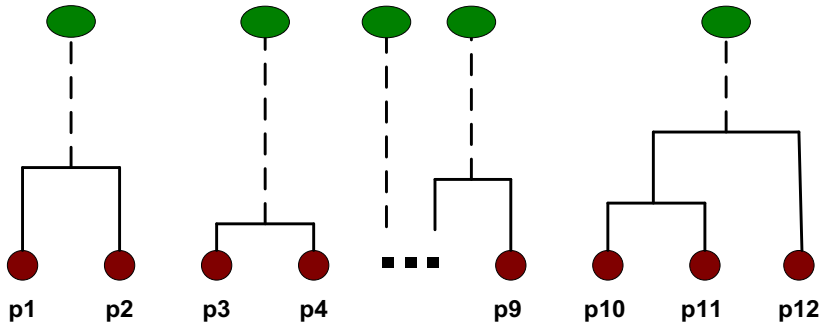
Agglomerative Clustering Ex: Intermediate Situation

- After some merging steps, we obtain some clusters



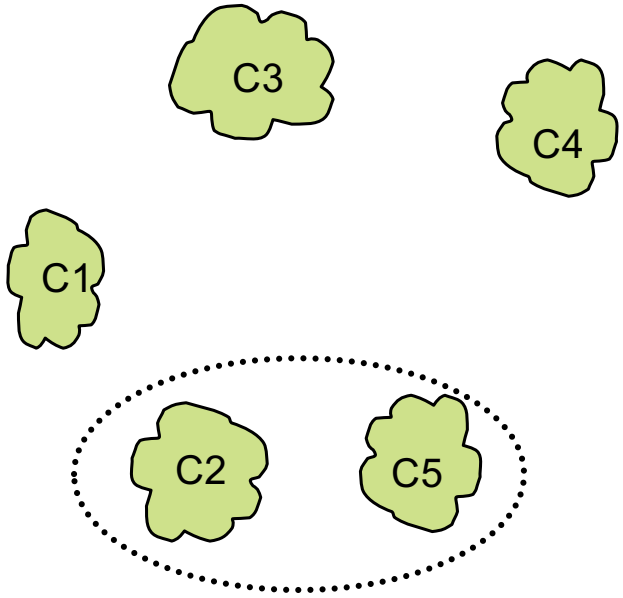
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



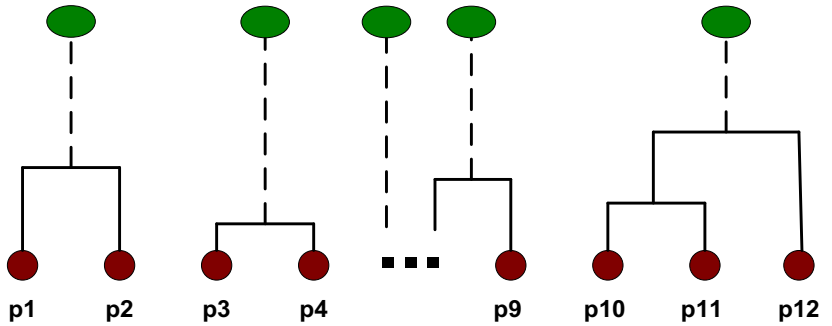
Agglomerative Clustering Ex: Intermediate Situation (II)

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix



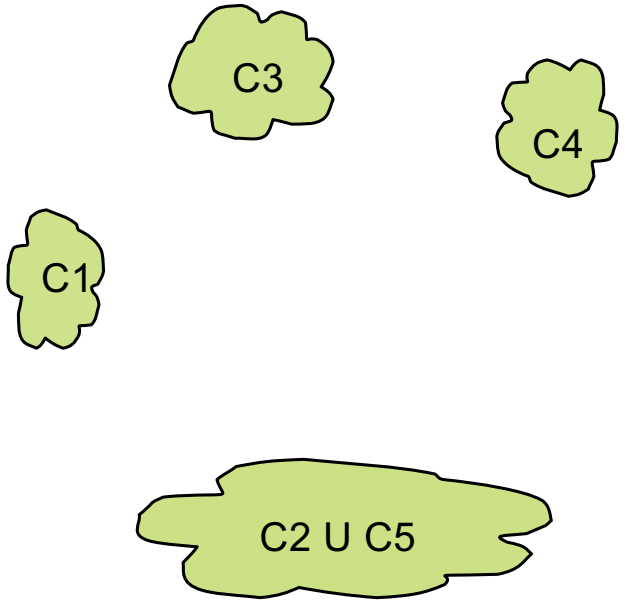
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



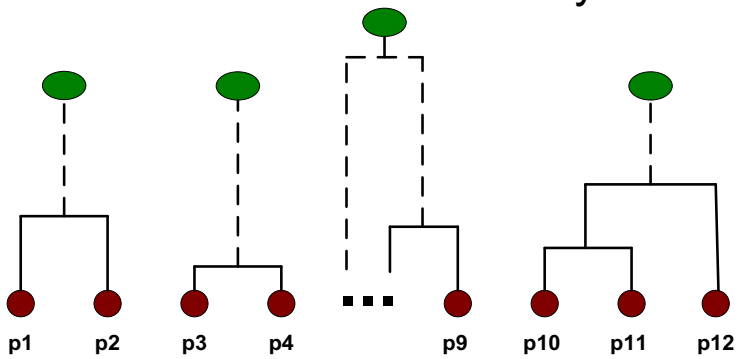
Agglomerative Clustering Ex: After Merging

- The question is “How do we update the proximity matrix?”

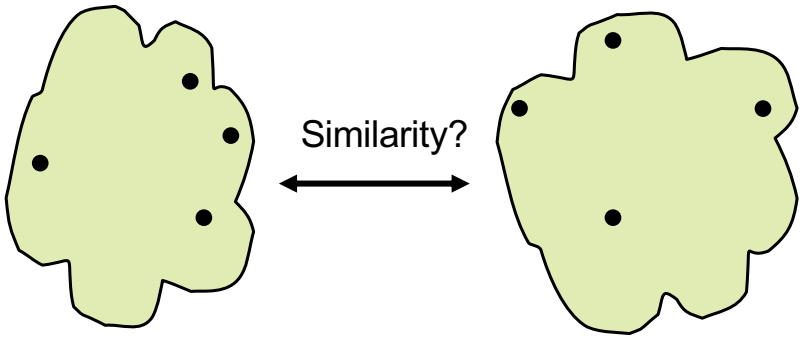


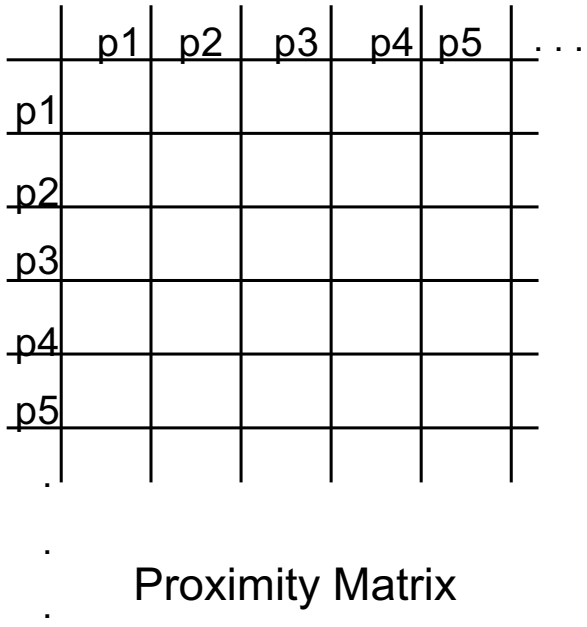
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity?

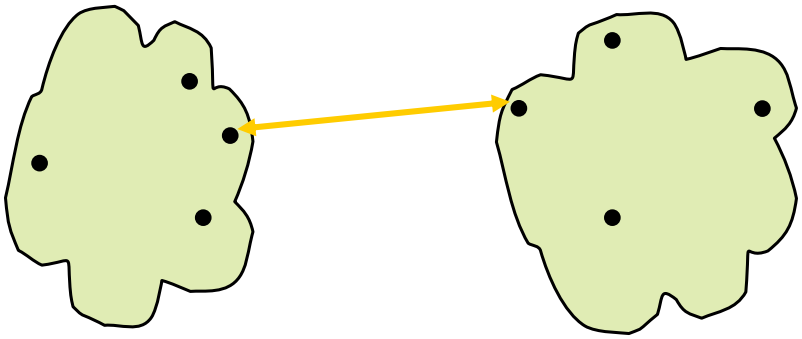


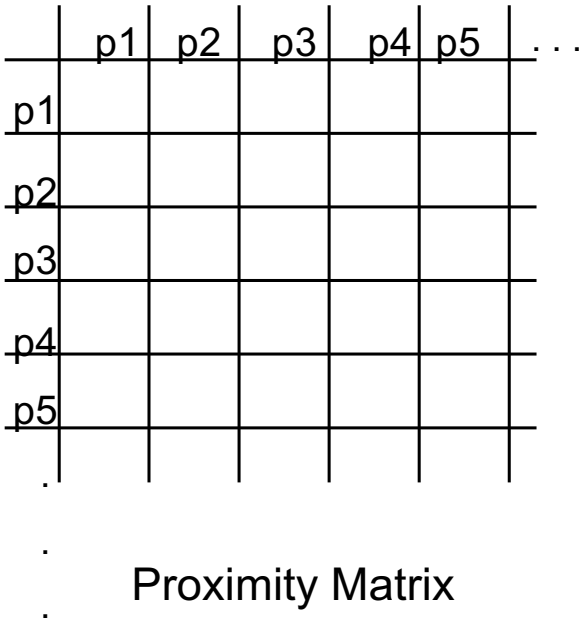


Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity?

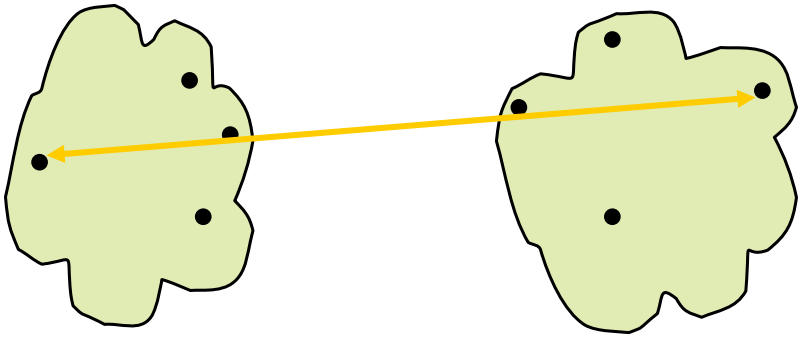


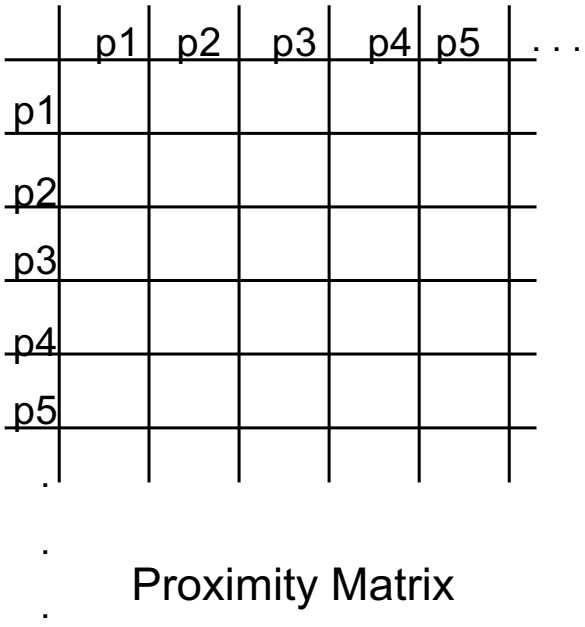


Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity?

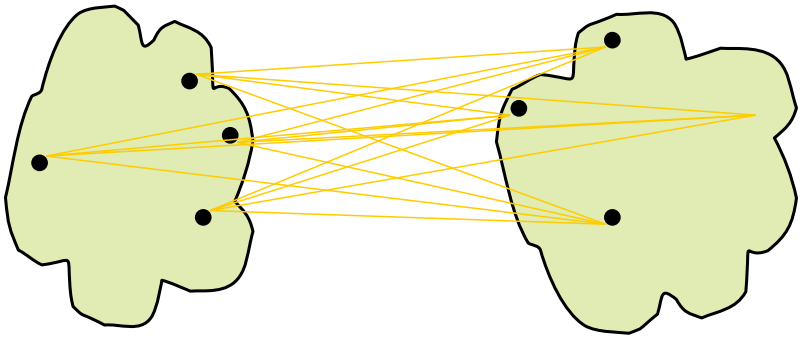


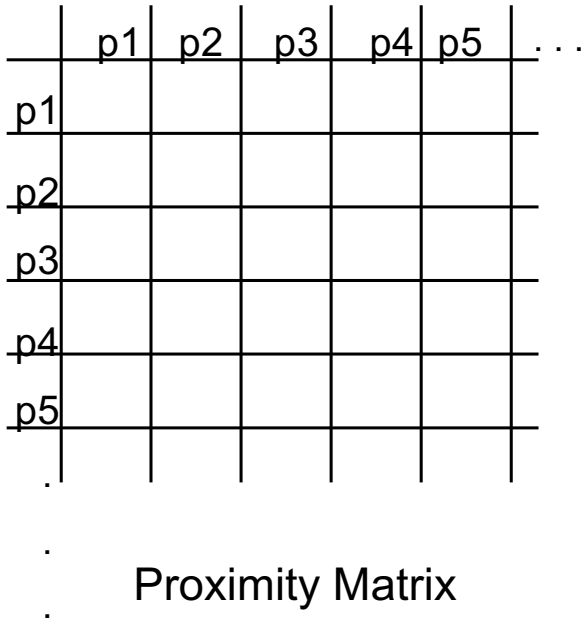


Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity?

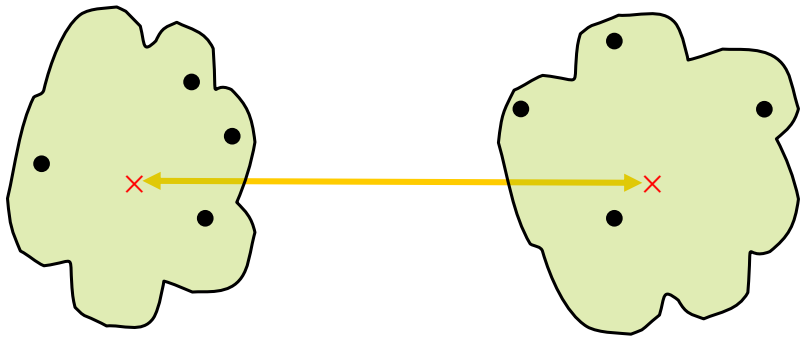


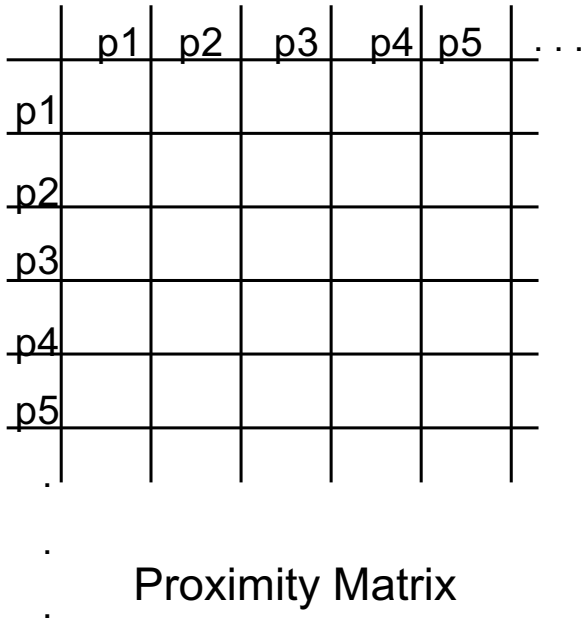


Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity?





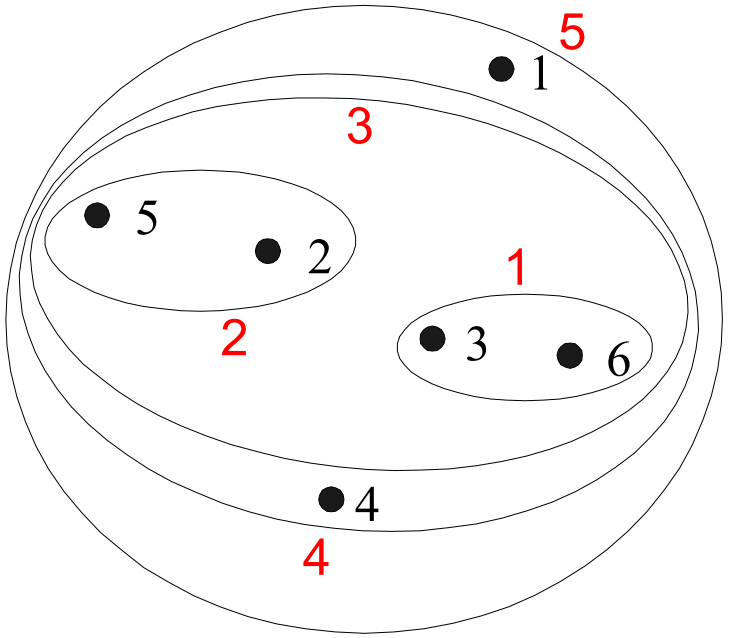
Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

Single Link vs. Complete Link

- Another way to view the processing of the hierarchical algorithm is to create links between their elements in order of increasing distance
  - Single Link, will merge two clusters when a single pair of elements is linked (equivalent to MIN)
  - Complete Linkage will merge two clusters when all pairs of elements have been linked (equivalent to MAX).

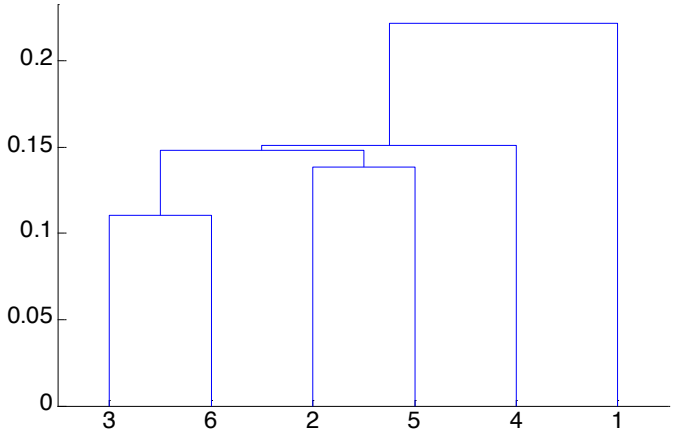
Hierarchical Clustering Similarity: MIN



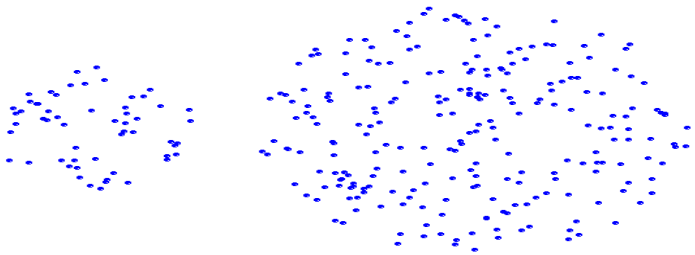
Nested Clusters

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0

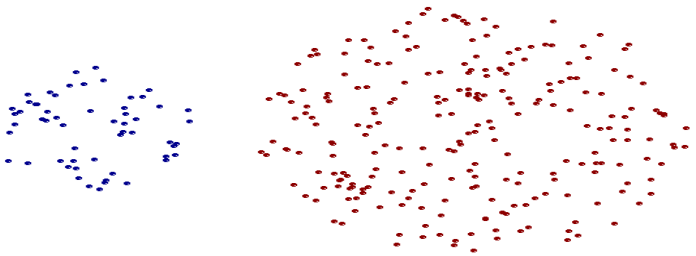
Dendrogram:



☺ Strength of MIN



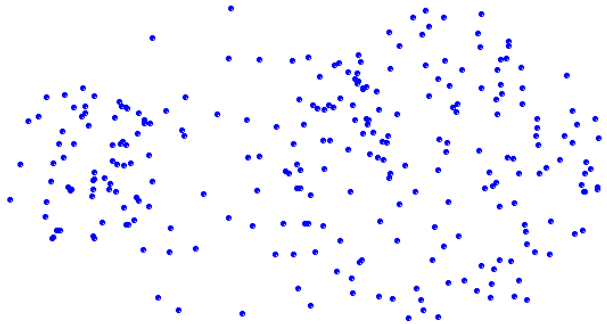
Original Points



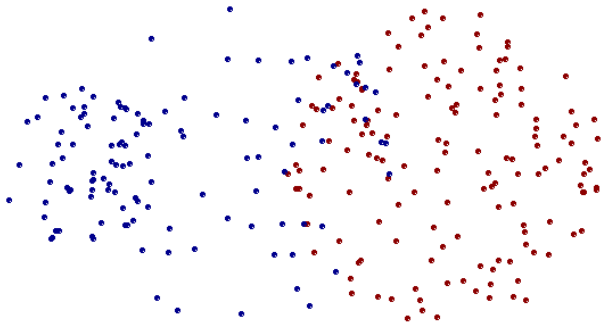
Two Clusters

Can handle non-elliptical shapes

☹ Limitations of MIN



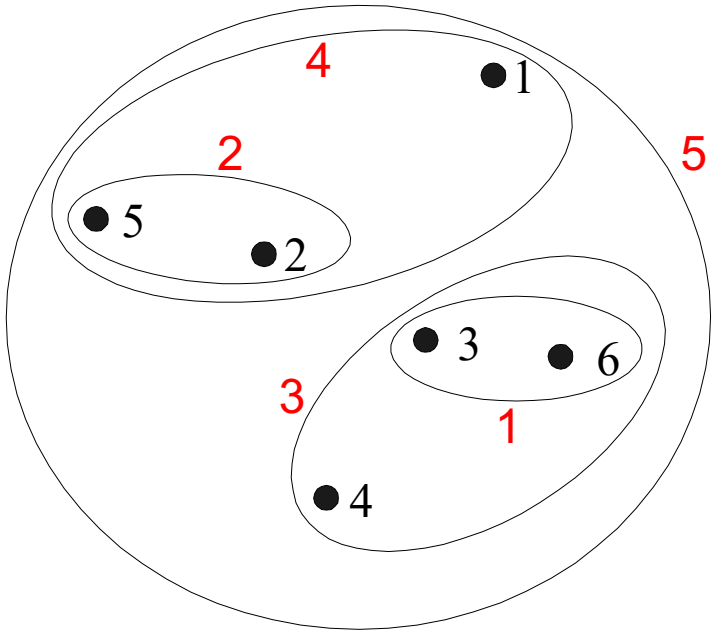
Original Points



Two Clusters

Sensitive to noise and outliers

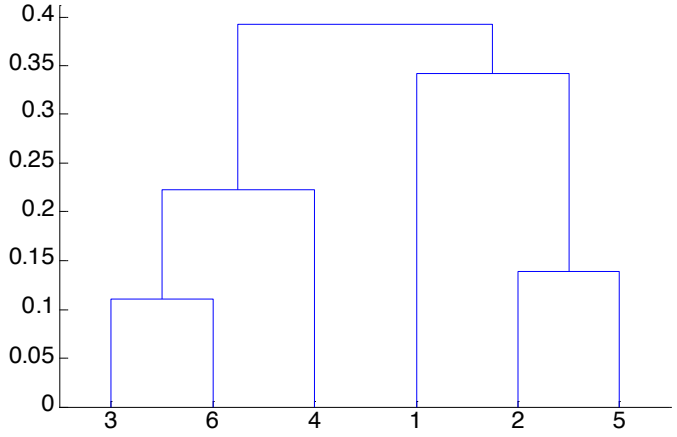
Hierarchical Clustering Similarity: MAX



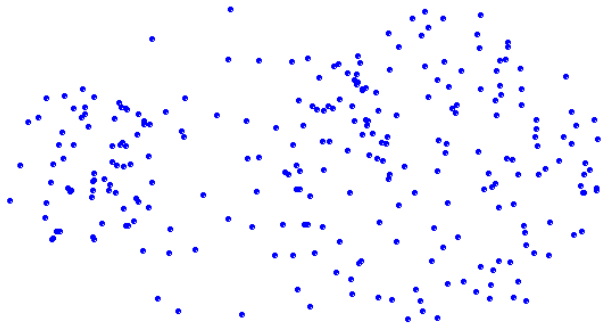
Nested Clusters

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0

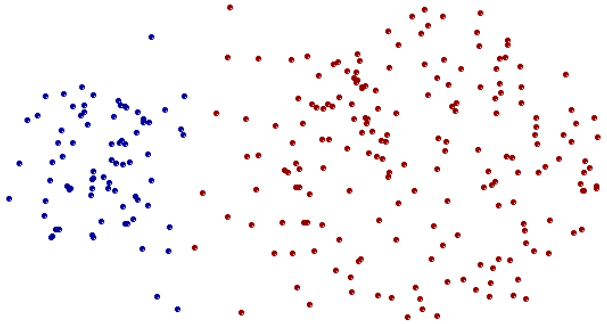
Dendrogram:



☺ Strength of MAX



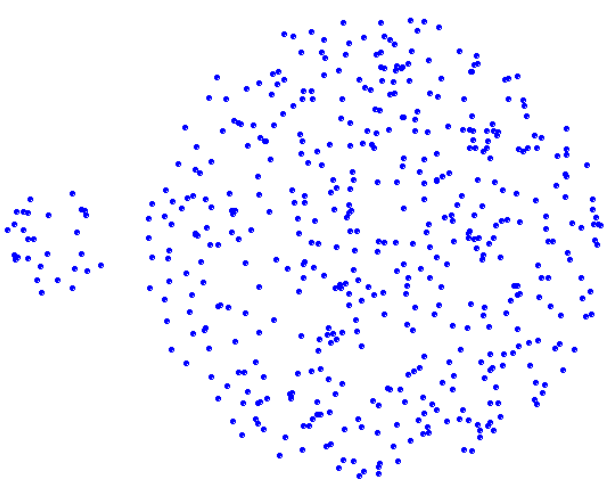
Original Points



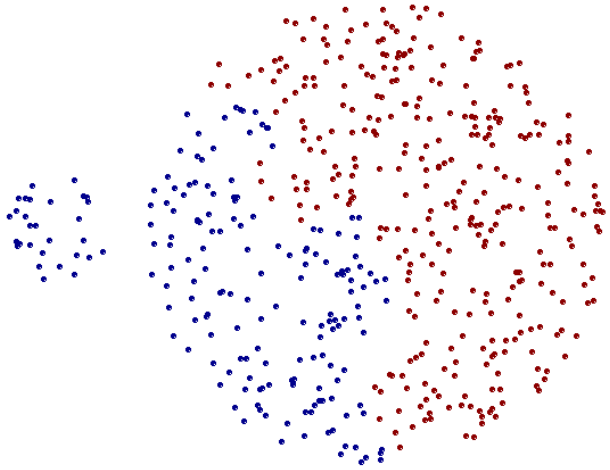
Two Clusters

Less susceptible to noise and outliers

☹ Limitations of MAX



Original Points



Two Clusters

Tends to break large clusters

Biased towards globular clusters

Hierarchical Clustering Similarity: Group Average

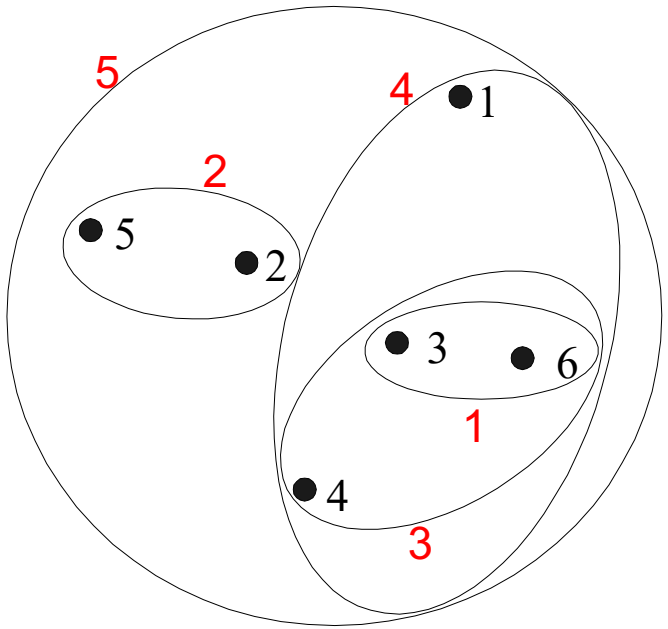
- Proximity of two clusters is the average of pairwise proximity between points in the two clusters:

$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

- Need to use average connectivity for scalability since total proximity favors large clusters

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0

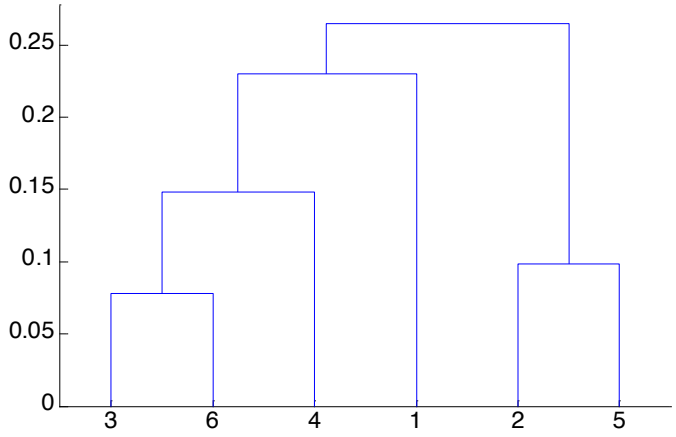
Hierarchical Clustering Similarity: Group Average



Nested Clusters

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0

Dendrogram:



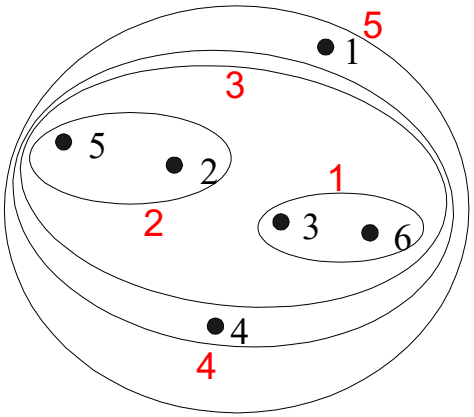
Hierarchical Clustering Similarity: Group Average

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

Cluster Similarity: Ward's Method

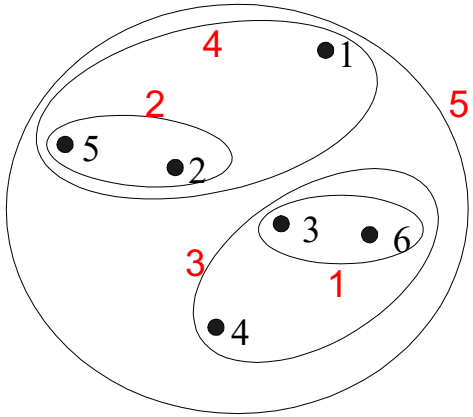
- Similarity of two clusters is based on the increase in squared error (SSE) when two clusters are merged
  - Similar to group average if distance between points is distance squared
- ☺ Less susceptible to noise and outliers
- ☹ Biased towards globular clusters
- Hierarchical analogue of K-means
  - Can be used to initialize K-means

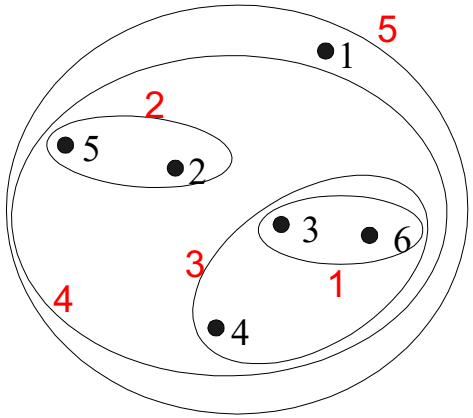
Hierarchical Clustering: Comparison



MIN

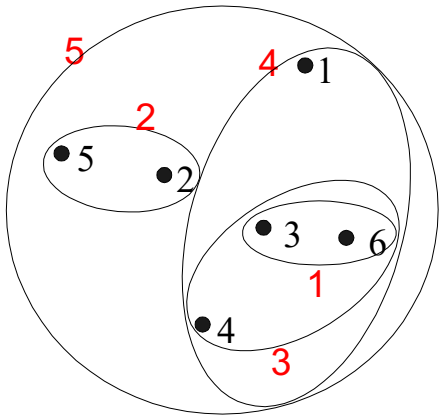
MAX





Group Average

Ward's Method



Hierarchical Clustering: Time and Space Requirements

- $O(N^2)$ space since it uses the proximity matrix
  - N is the number of points.
- $O(N^3)$ time in many cases
  - There are N steps and at each step the size, N^2 proximity matrix must be updated and searched
  - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

Hierarchical Clustering: Problems and Limitations

- ☹ Computational complexity in time and space
- ☹ Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - ☹ Sensitivity to noise and outliers
  - ☹ Difficulty handling different sized clusters and convex shapes
  - ☹ Breaking large clusters

Density-Based Clustering

DBSCAN

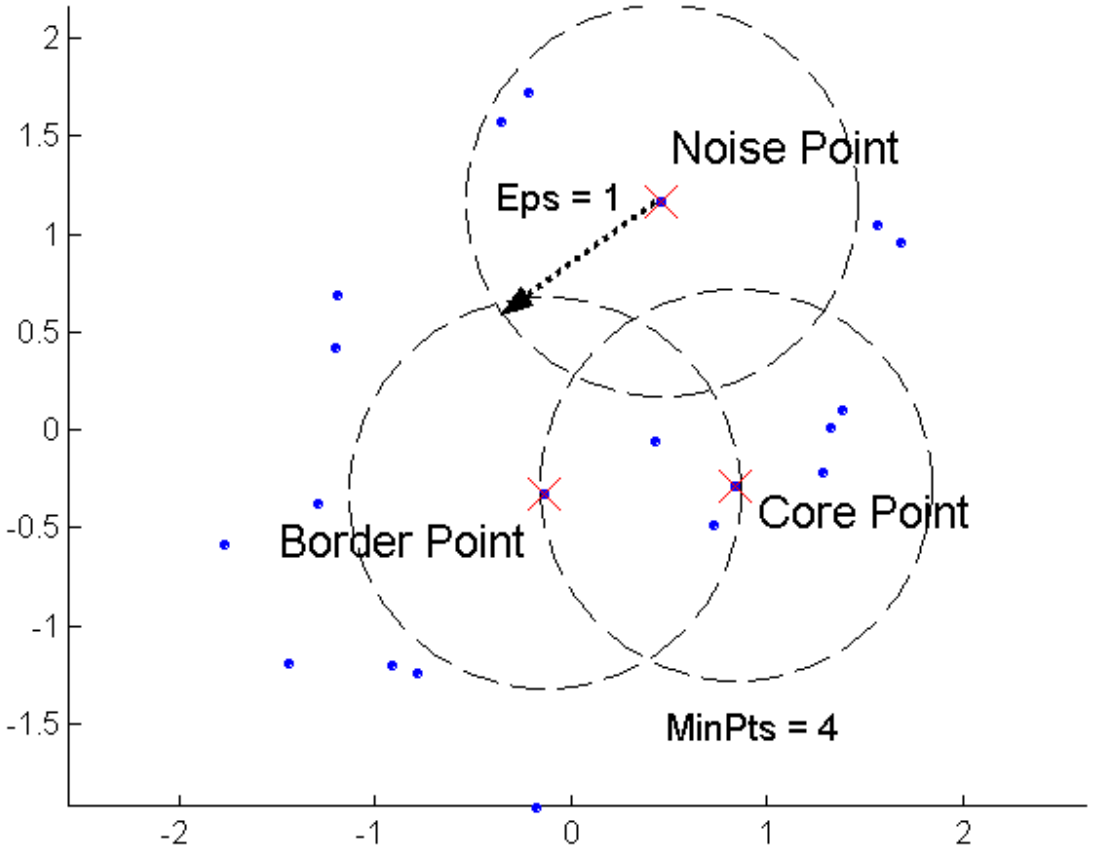
DBSCAN: Density-Based Clustering

- DBSCAN is a Density-Based Clustering algorithm
- In density-based clustering we partition points into dense regions separated by not-so-dense regions
- Important Questions:
  - How do we measure density?
  - What is a dense region?
- DBSCAN:
  - Density at point p: number of points within a circle of radius Eps
  - Dense Region: A circle of radius Eps that contains at least MinPts points

DBSCAN: Characterization of Points

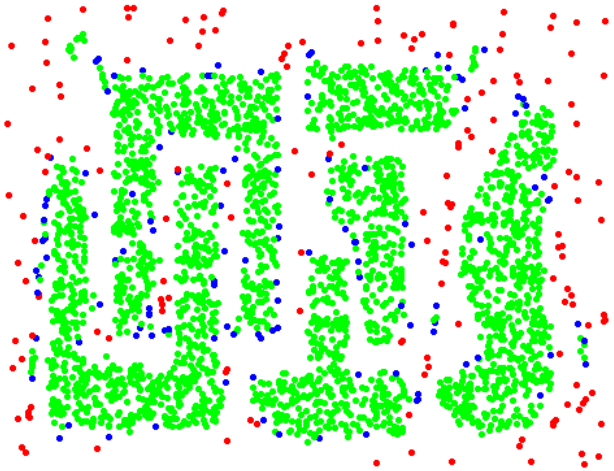
- A point is a core point if it has more than a specified number of points (MinPts) within Eps
  - These points belong in a dense region and are at the interior of a cluster
- A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.
- A noise point is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points



DBSCAN: Core, Border, and Noise Points

Original Points

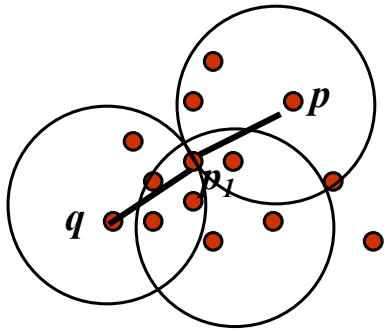


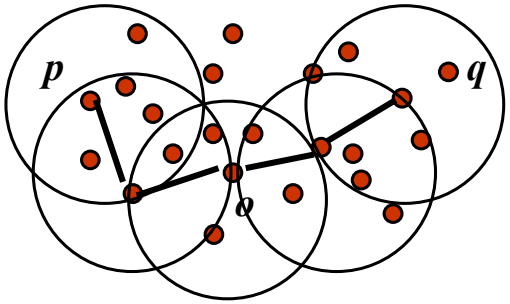
Point types: core, border and noise

Eps = 10, MinPts = 4

Density-Connected Points

- Density edge
  - We place an edge between two core points q and p if they are within distance Eps
- Density-connected
  - A point p is density-connected to a point q if there is a path of edges from p to q



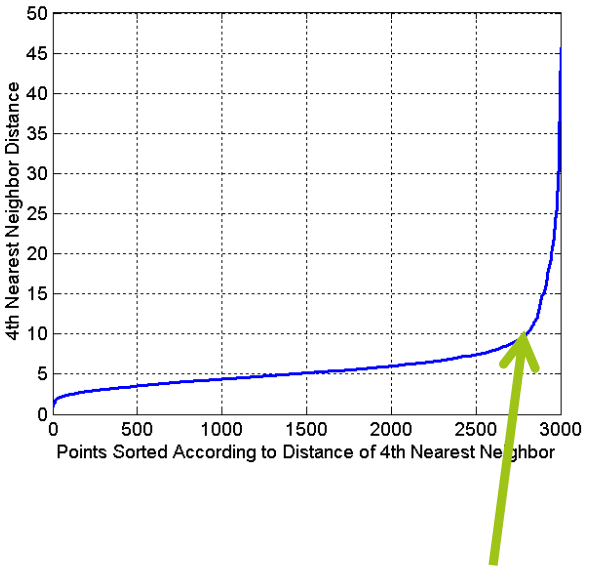


DBSCAN Algorithm

- Label points as core, border and noise
- Eliminate noise points
- For every core point p that has not been assigned to a cluster:
  - Create a new cluster with the point p and all the points that are density-connected to p
- Assign border points to the cluster of the closest core point

DBSCAN: Determining Eps and MinPts

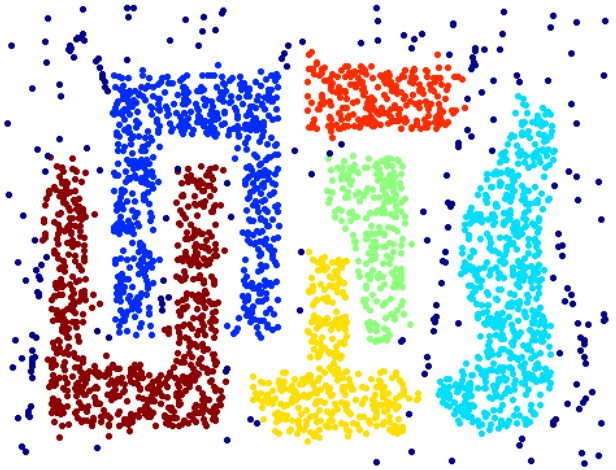
- Rationale: for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- Plot the sorted distance of every point to its k^{th} nearest neighbor
- Find the distance d where there is a "knee" in the curve
  - Eps = d, MinPts = k



Eps ~ 7-10
MinPts = 4

When Does DBSCAN Work Well?

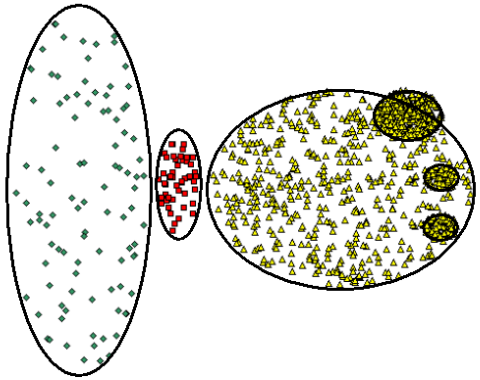
Original Points



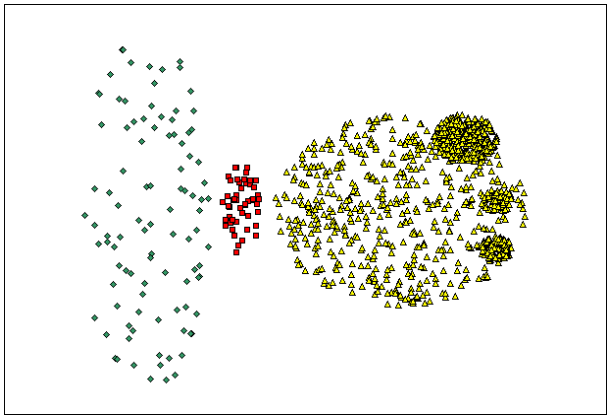
Clusters

- ☺ Resistant to Noise
- ☺ Can handle clusters of different shapes and sizes

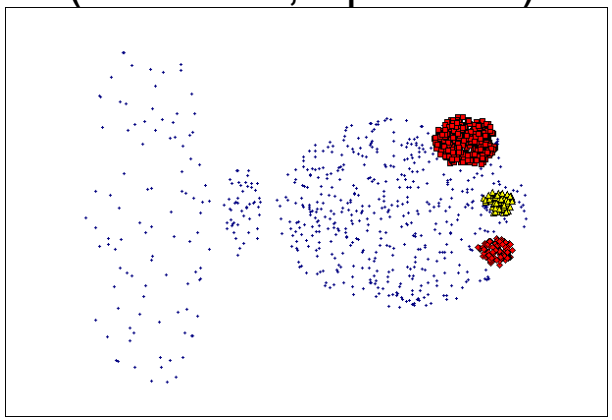
When Does DBSCAN Not Work?



Original Points



(MinPts=4, Eps=9.75)



(MinPts=4, Eps=9.92)

- ☹ Small variation in density can generate very different clusters
- ☹ High-dimensional data

Sensitivity to Parameters

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

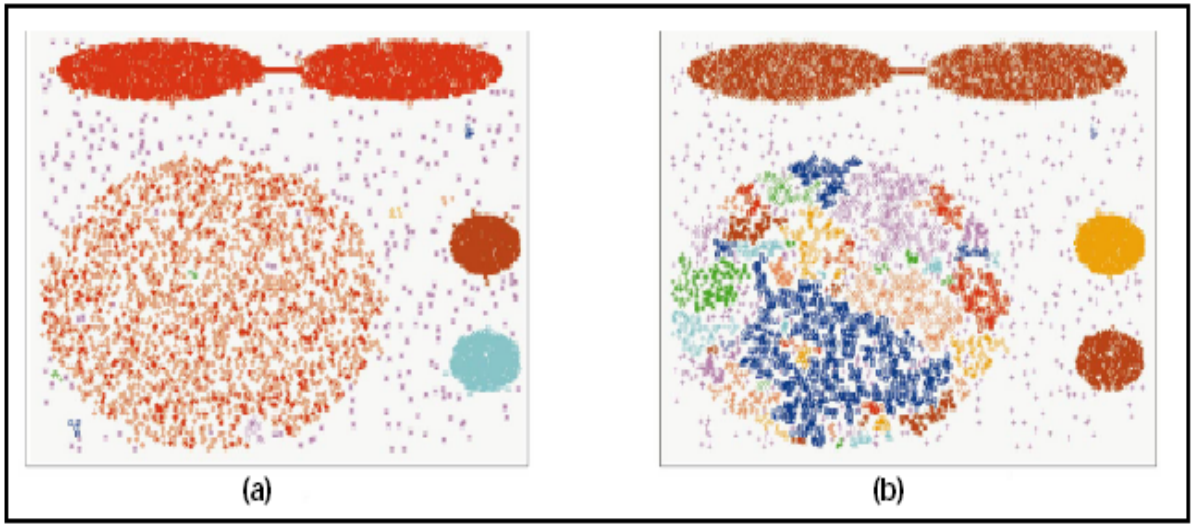
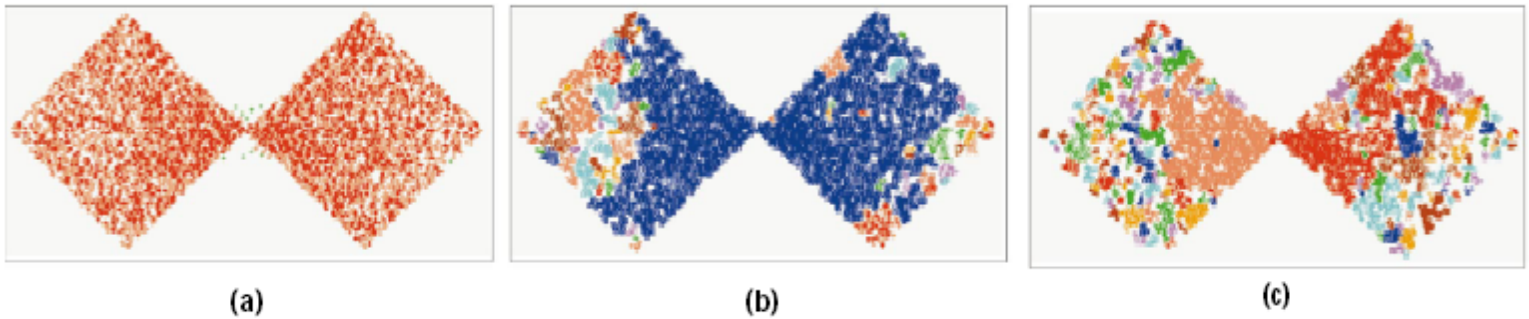


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



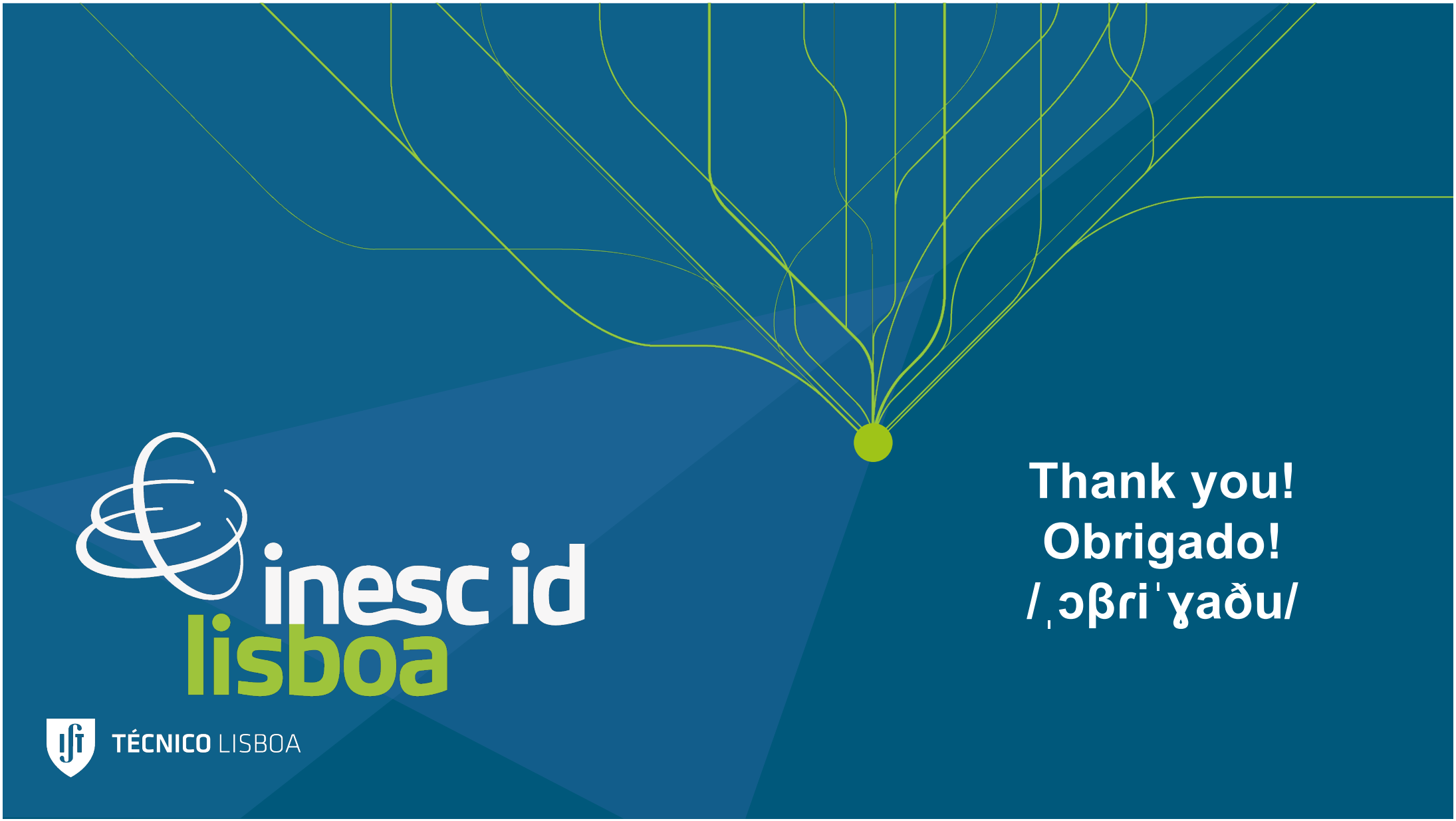
Remarks and Conclusions

Remarks

- ☺ Clustering is a very useful approach to extract knowledge from “Big Unlabeled Data”
- There is a huge variety of Clustering algorithms:
  - ☹ None is ”universal”, i.e., the choice of the best clustering algorithm is highly dependent on the data...
  - ☹ ...and most of the times it is not feasible to analyze the data in order to know which algorithm to choose
- ☹ The evaluation (or validation) of the clustering results is even more difficult than the process of clustering

Other Clustering Algorithms

- PAM, CLARANS: Solutions for the k-medoids problem
- BIRCH: Constructs a hierarchical tree that acts a summary of the data, and then clusters the leaves
- MST: Clustering using the Minimum Spanning Tree
- ROCK: clustering categorical data by neighbor and link analysis
- LIMBO, COOLCAT: Clustering categorical data using information theoretic tools.
- CURE: Hierarchical algorithm uses different representation of the cluster
- CHAMELEON: Hierarchical algorithm uses closeness and interconnectivity for merging


inesc id
lisboa
TÉCNICO LISBOA
Thank you!
Obrigado!
/ˌɔβriˈɣaðu/