CIVIL-557

Decision Aid Methodologies In Transportation

Lecture 10:
Data Mining in Transport – Introduction & Clustering

Nikola Obrenovic

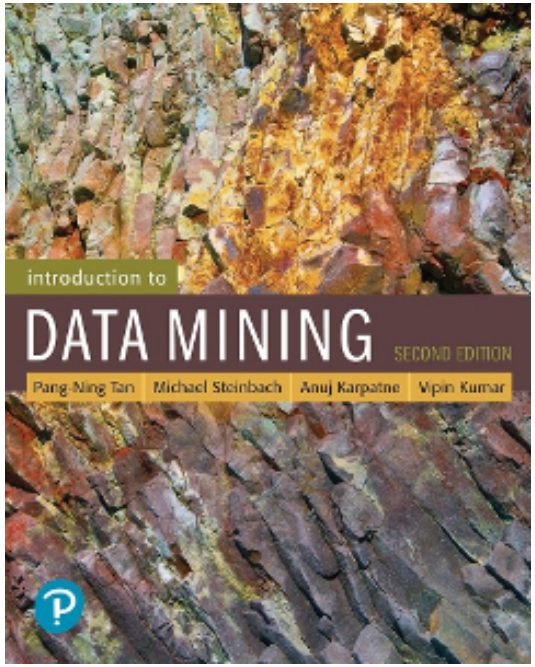
Transport and Mobility Laboratory TRANSP-OR
École Polytechnique Fédérale de Lausanne EPFL

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Acknowledgement

- The content of these slides has been partially taken over from the official slides accompanying the book: P.-N. Tan, M. Steinbach, A. Karpatne, V. Kumar: Introduction to Data Mining (2nd Edition)
- https://www-users.cs.umn.edu/~kumar001/dmbook/index.php



TRANSP-OR

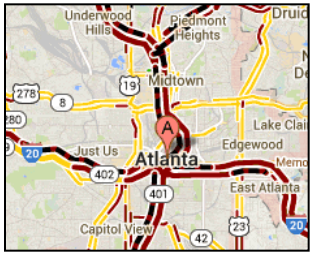
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Introduction to data mining

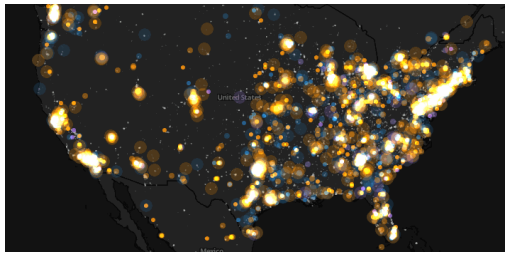
- There has been enormous data growth in both commercial and scientific databases due to advances in data generation and collection technologies
- Frequent strategy
  - Gather whatever data you can whenever and wherever possible.
- Expectations
  - Gathered data will have value either for the purpose collected or for a purpose not envisioned.

Cyber Security

E-Commerce

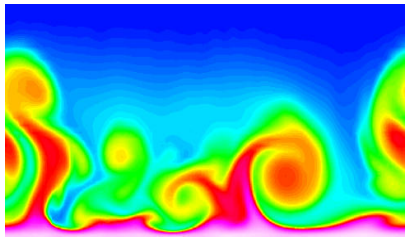


Traffic Patterns



Social Networking: Twitter

Sensor Networks



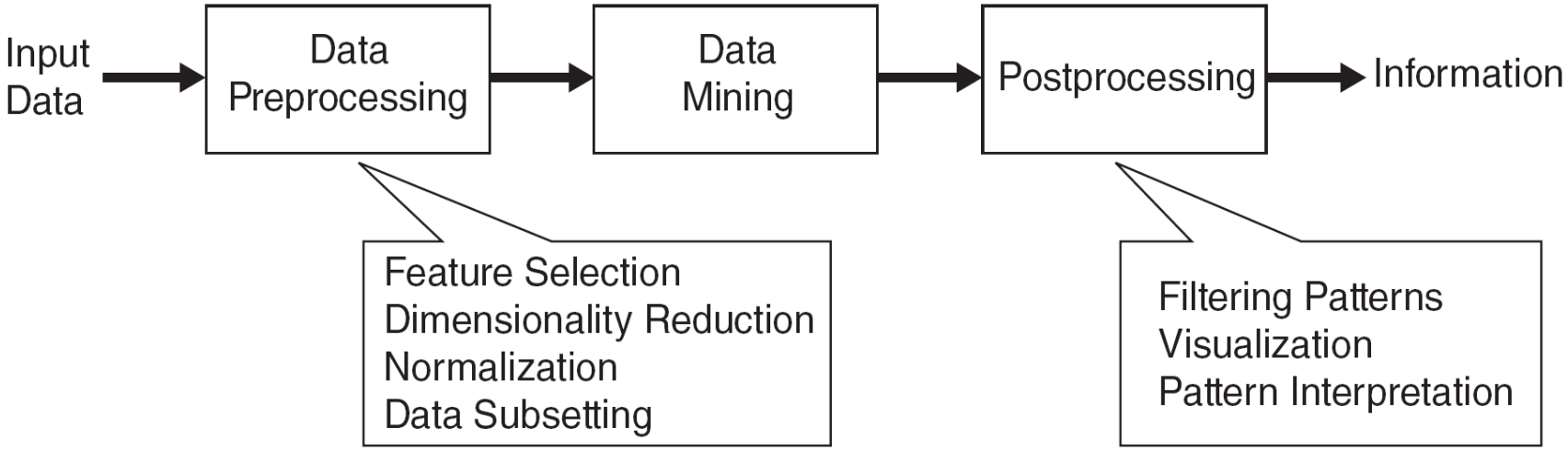
Computational Simulations

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Data Mining Definition

- Non-trivial extraction of implicit, previously unknown and potentially useful information from data
- Exploration & analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns



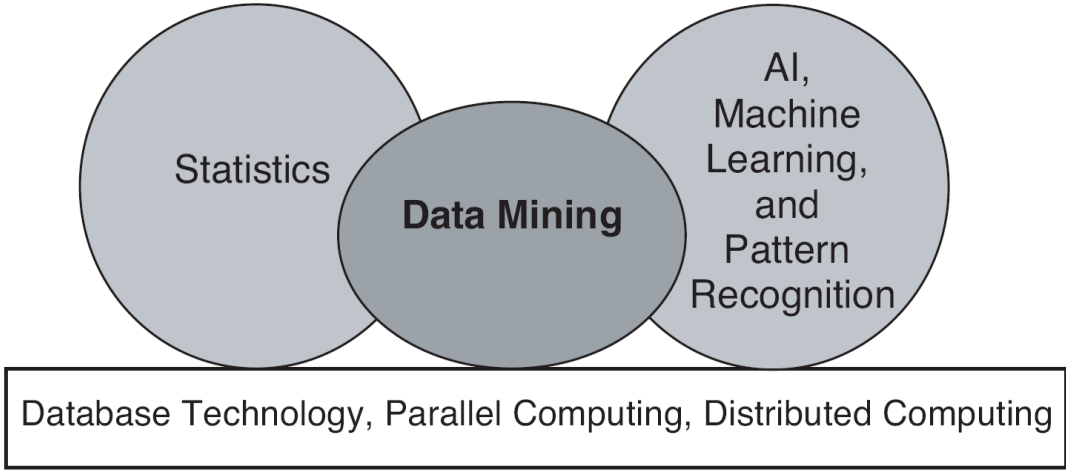
Data Mining Definition

- What is not Data Mining?
  - Look up phone number in phone directory
  - Query a Web search engine for information about "Amazon"
- What is Data Mining?
  - Certain names are more prevalent in certain US locations (O'Brien, O'Rourke, O'Reilly… in Boston area)
  - Group together similar documents returned by search engine according to their content and context (e.g., Amazon rainforest, Amazon.com)

Data Mining Origins

- Draws ideas from machine learning/AI, pattern recognition, statistics, and database systems

- Traditional techniques may be unsuitable due to data that is
  - Large-scale
  - High dimensional
  - Heterogeneous
  - Complex
  - Distributed

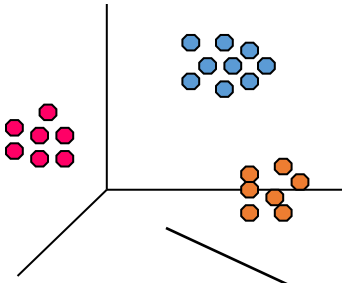


- A key component of the emerging field of data science and data-driven discovery and analysis

Data Mining Tasks

- Prediction Methods
  - Use some variables to predict unknown or future values of other variables.
- Description Methods
  - Find human-interpretable patterns that describe the data.

Data Mining Tasks

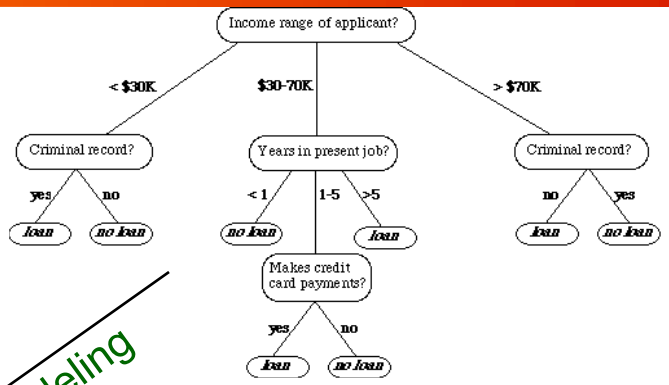


Clustering

Data

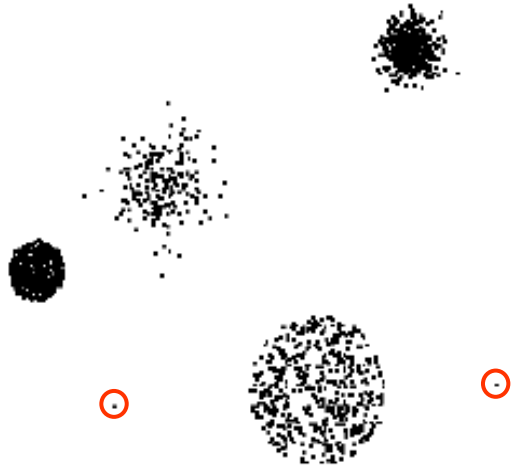
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes
11	No	Married	60K	No
12	Yes	Divorced	220K	No
13	No	Single	85K	Yes
14	No	Married	75K	No
15	No	Single	90K	Yes

Predictive Modeling



Association Rules

Anomaly Detection



TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Clustering Task Examples

- Finding similar energy consumption users
- Finding similar transport paths
- Identifying congestion patterns

Classification Task Examples

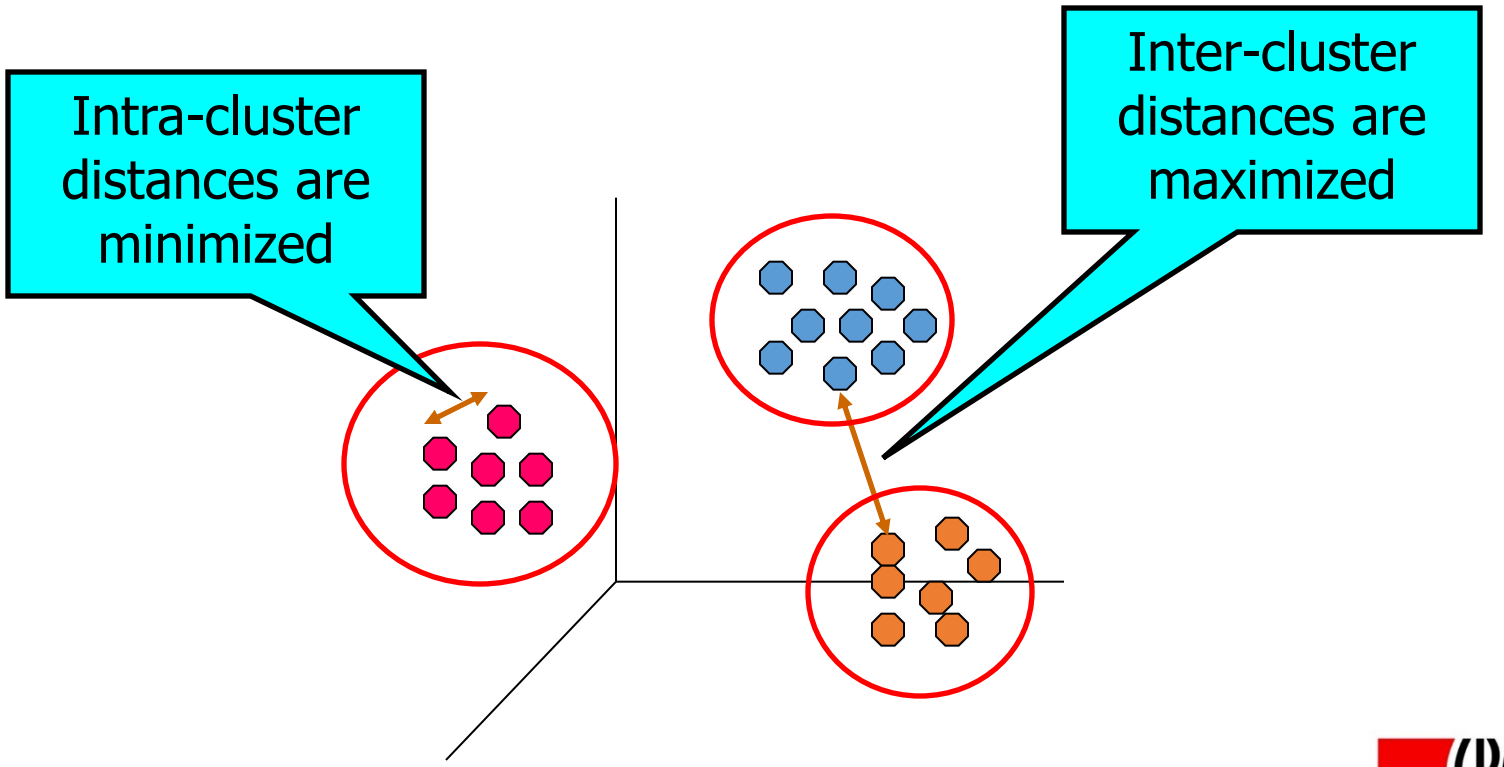
- Classifying credit card transactions as legitimate or fraudulent
- Classifying land covers (water bodies, urban areas, forests, etc.) using satellite data
- Categorizing news stories as finance, weather, entertainment, sports, etc
- Identifying intruders in the cyberspace
- Predicting tumor cells as benign or malignant

Forecasting Task Examples

- Forecasting traffic flows and congestions
- Forecasting stock market prices
- Forecasting electricity demand

What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



What is not Cluster Analysis?

- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Results of a query
  - Groupings are a result of an external specification
  - Clustering is a grouping of objects based on the data
- Supervised classification
  - Have class label information

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

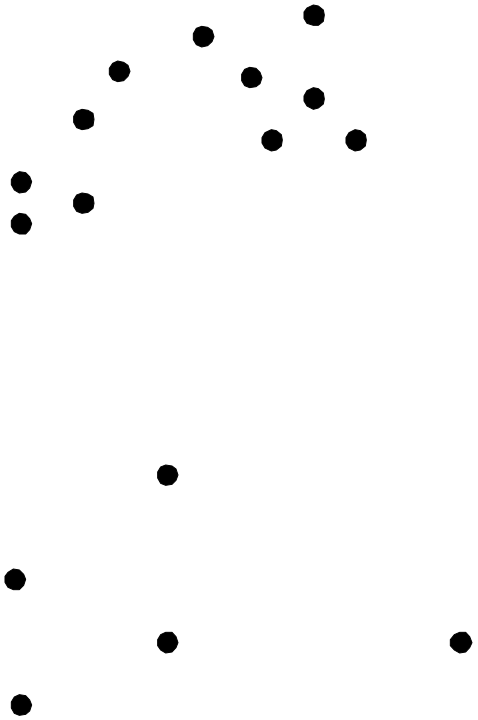
Two Clusters

Four Clusters

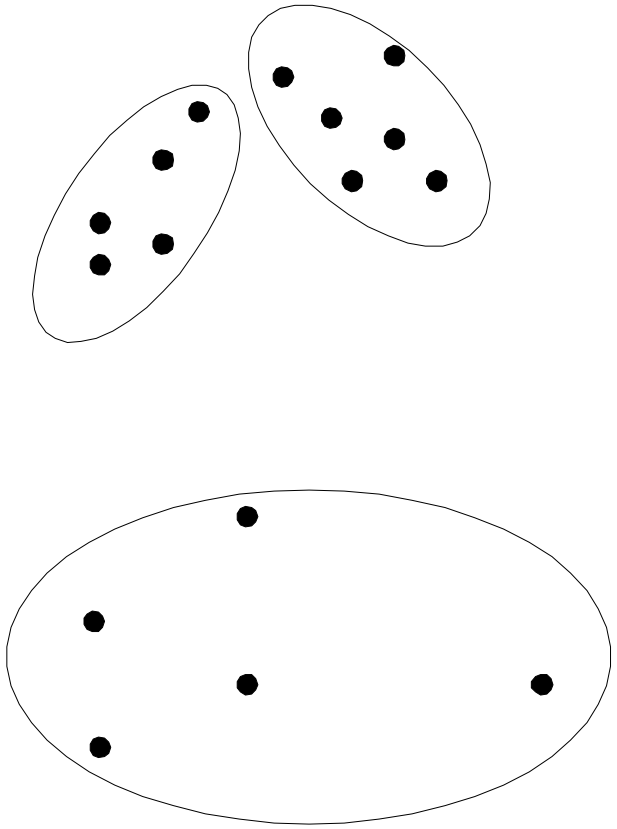
Types of Clusterings

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

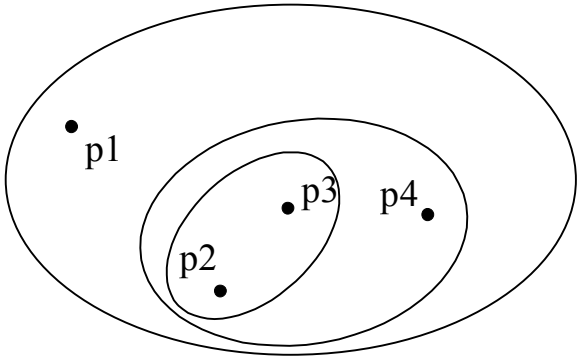


Original Points

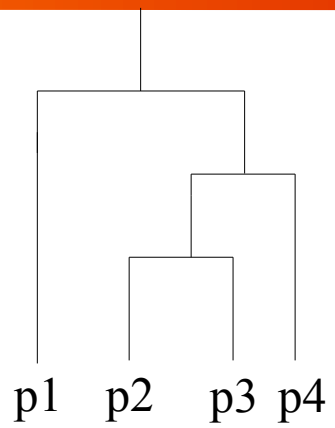


A Partitional Clustering

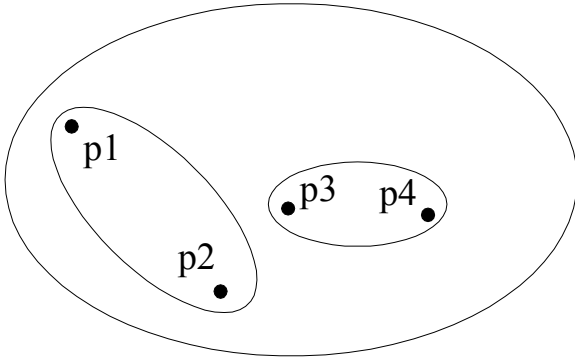
Hierarchical Clustering



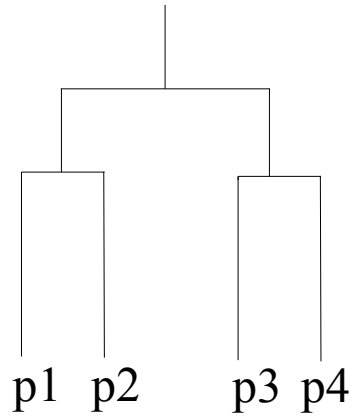
Standard Hierarchical Clustering



Standard Dendrogram



Binary Hierarchical Clustering



Binary Dendrogram

Other Distinctions Between Sets of Clusters

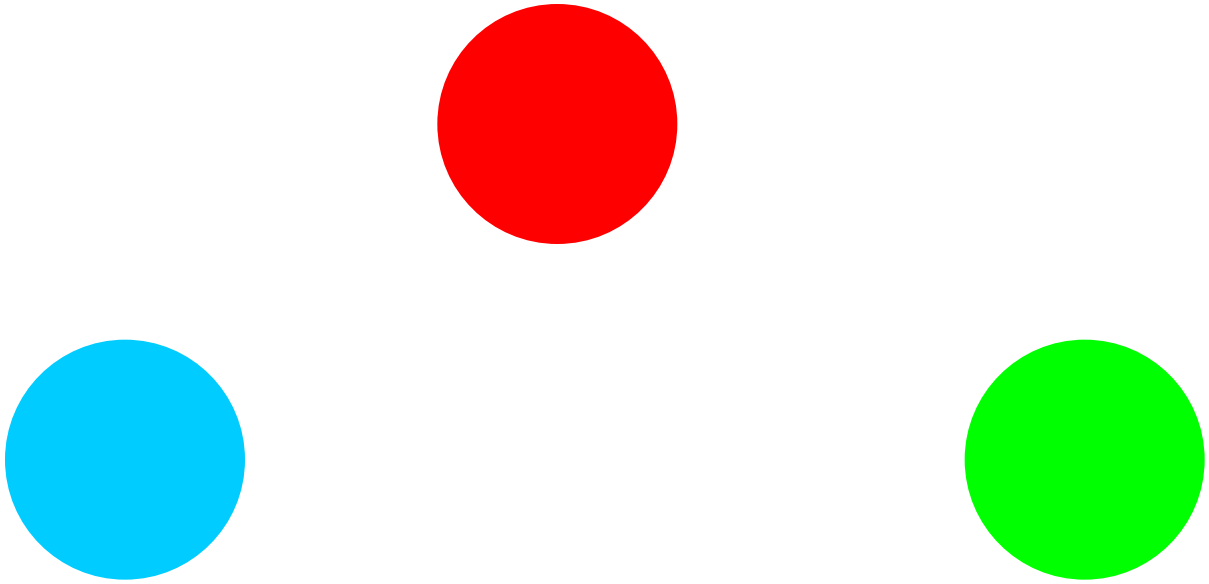
- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or ‘border’ points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
- Partial versus complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
  - Clusters of widely different sizes, shapes, and densities

Types of Clusters

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters
- Described by an Objective Function

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

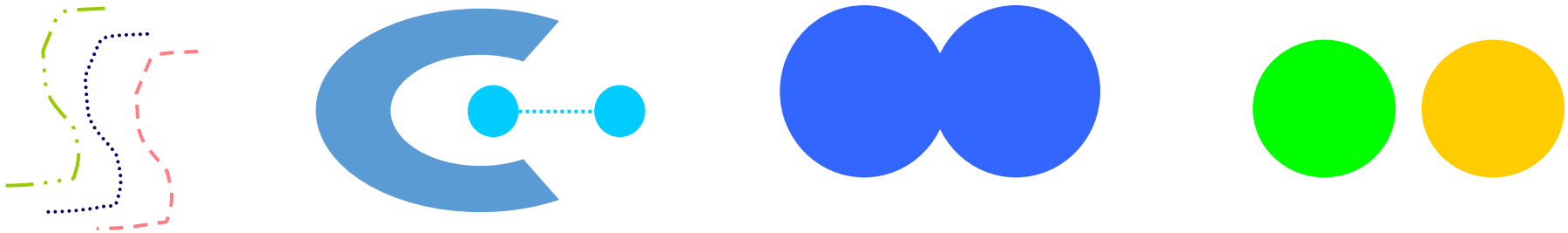
Types of Clusters: Center-based

- 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:
    - Centroid = the average of all the points in the cluster, or
    - Medoid = the most “representative” point of a cluster

4 center-based clusters

Types of Clusters: Contiguity-Based

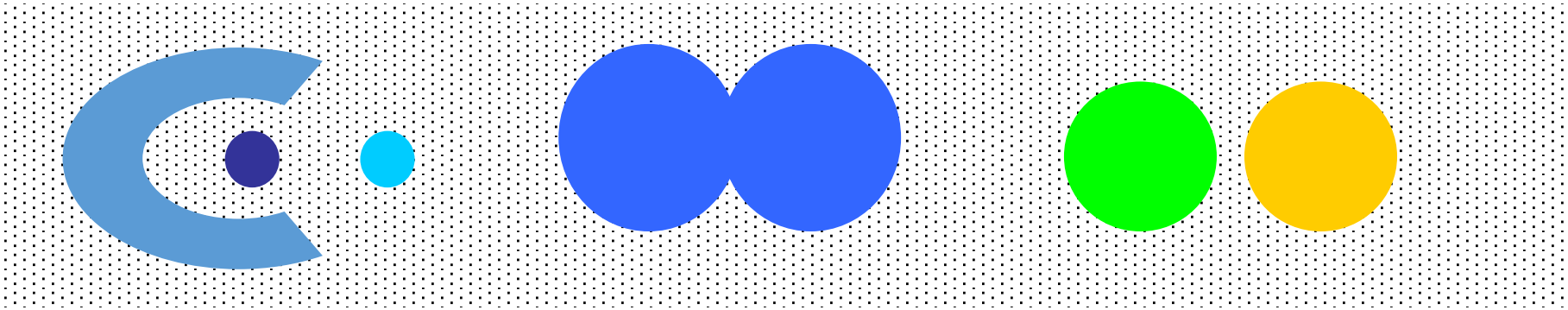
- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



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: Objective function

- Clusters Defined by an Objective Function
  - Finds clusters that minimize or maximize an objective function.
    - Objective function is usually some form of error measure
  - 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 problem)

Important Characteristics of the Input Data

- Similarity or density measure
  - Central to clustering
  - Depends on data and application
- Data characteristics that affect algorithms
  - Dimensionality (Sparseness)
  - Attribute type
  - Special relationships in the data (autocorrelation)
  - Distribution of the data
- Noise and Outliers
  - Often interfere with the operation of the clustering algorithm

Clustering Algorithms

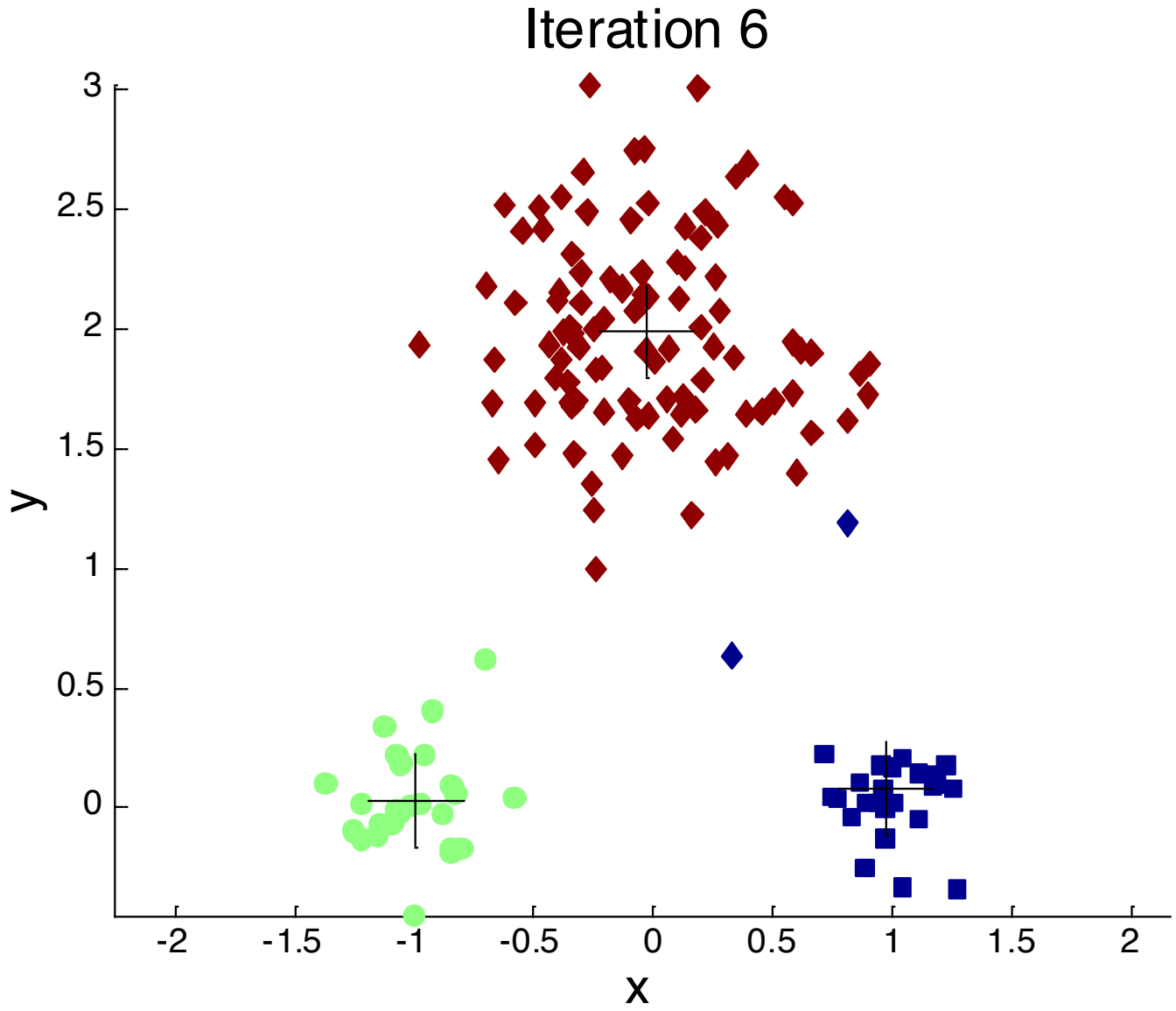
- K-means and its variants
- Hierarchical clustering
- Density-based clustering

K-Means Clustering Algorithm

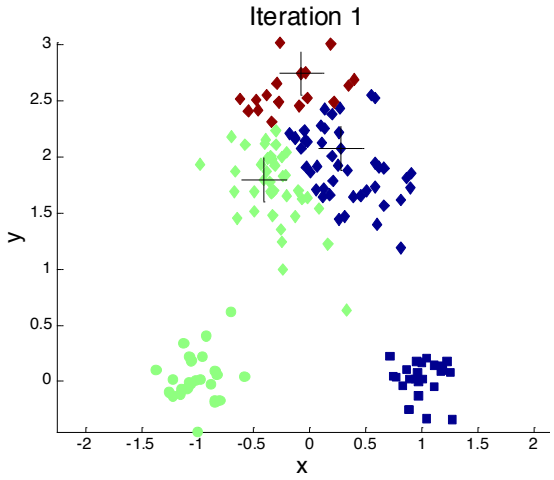
- Partitional clustering approach
- Number of clusters, K, must be specified
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- The basic algorithm is very simple

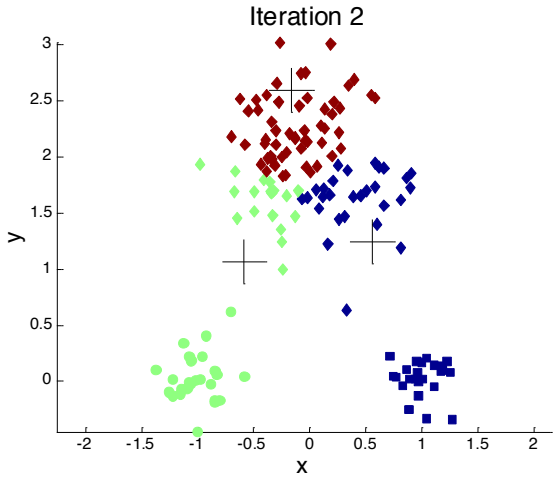
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

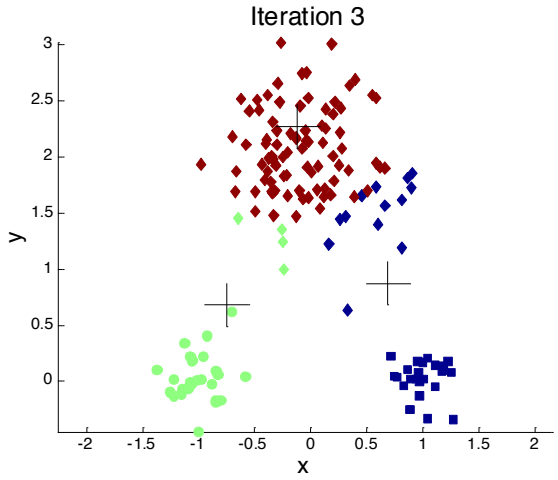
Example of K-Means Clustering

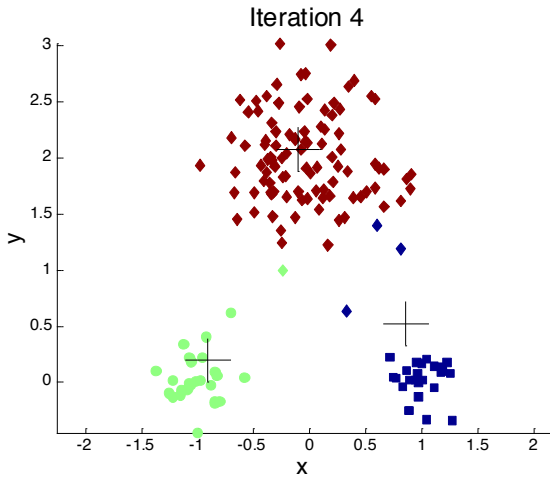


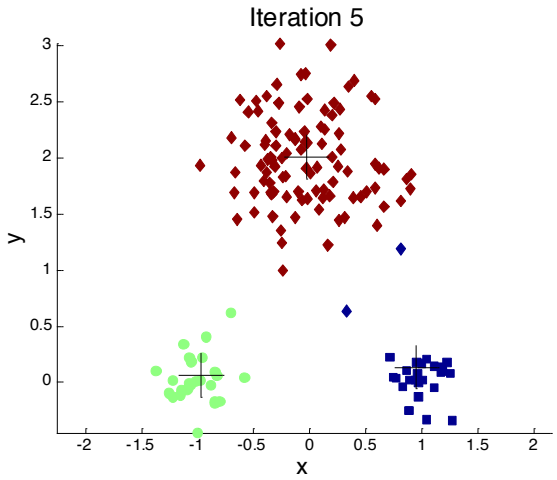
Example of K-Means Clustering

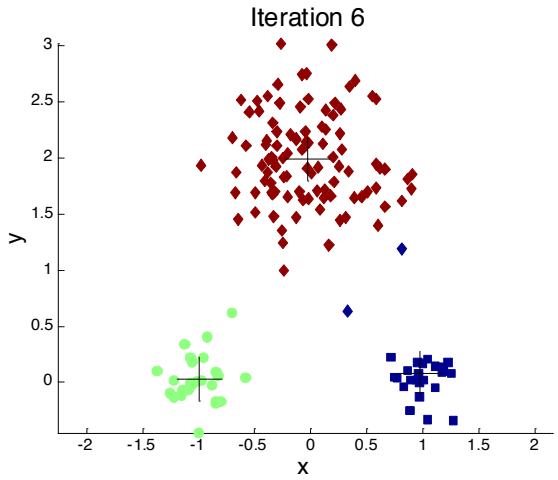












TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

K-Means Clustering - Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- 'Closeness' is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- 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 = number of attributes

Evaluating K-Means Clustering

- Most common measure is Sum of Squared Error (SSE)
- For each point, the error is the distance to the nearest cluster
- To get SSE, we square these errors and sum them.

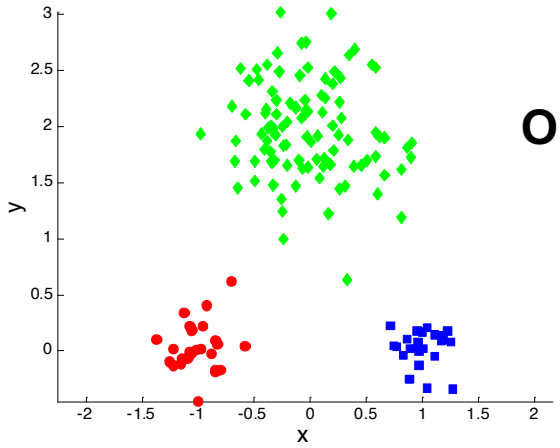
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

- x is a data point in cluster C_i and m_i is the representative point for cluster C_i
  - m_i corresponds to the center (mean) of the cluster
- Given two sets of clusters, we prefer the one with the smallest error
- One easy way to reduce SSE is to increase K, the number of clusters
  - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

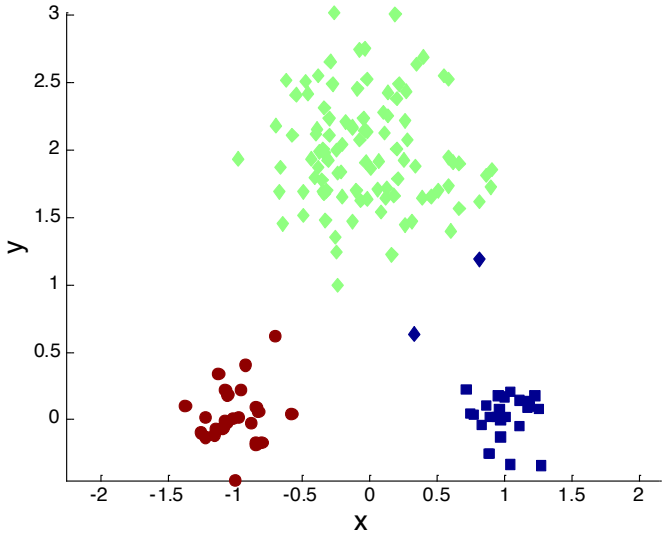
K-Means Clustering - Challenges

- The factors that contribute to the “goodness” (quality) of clustering:
  - Number of clusters
  - Initial centroids
  - Empty clusters

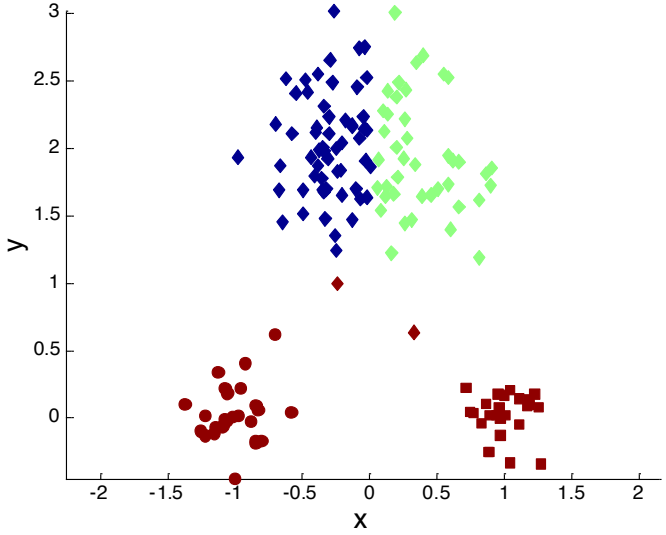
Two different K-Means Clusterings



Original Points



Optimal Clustering



Sub-optimal Clustering

TRANSP-OR

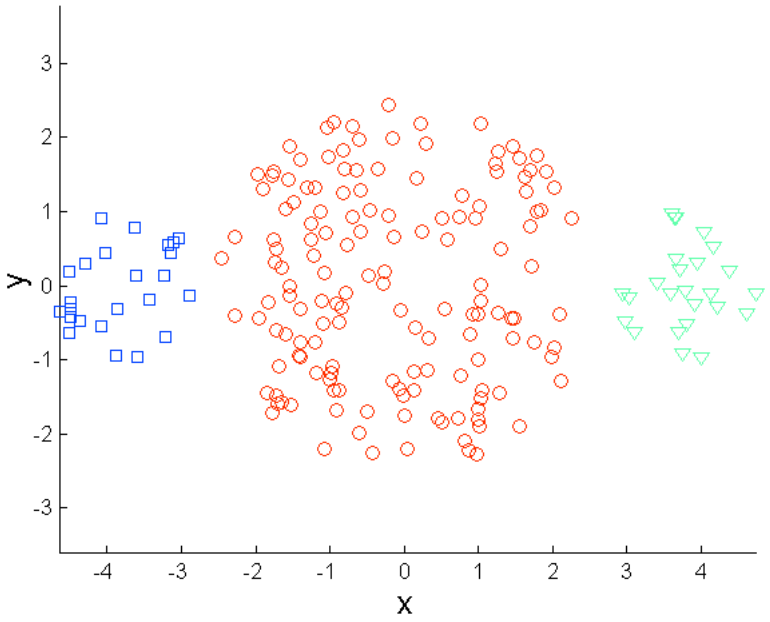
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Limitations of K-Means Clustering

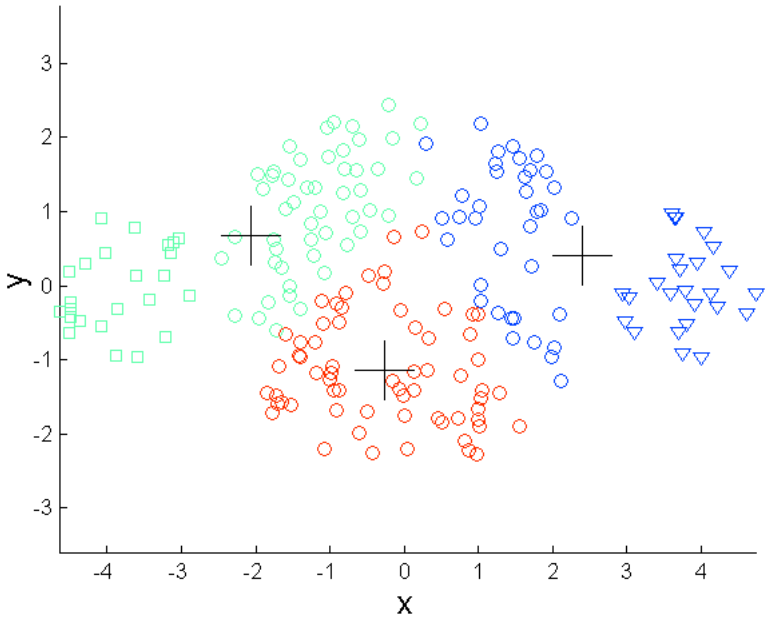
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

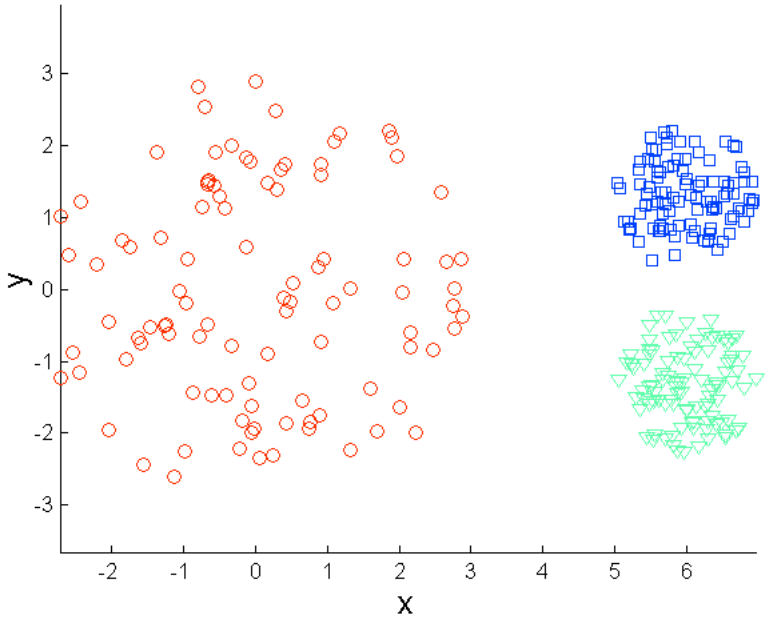


Original Points

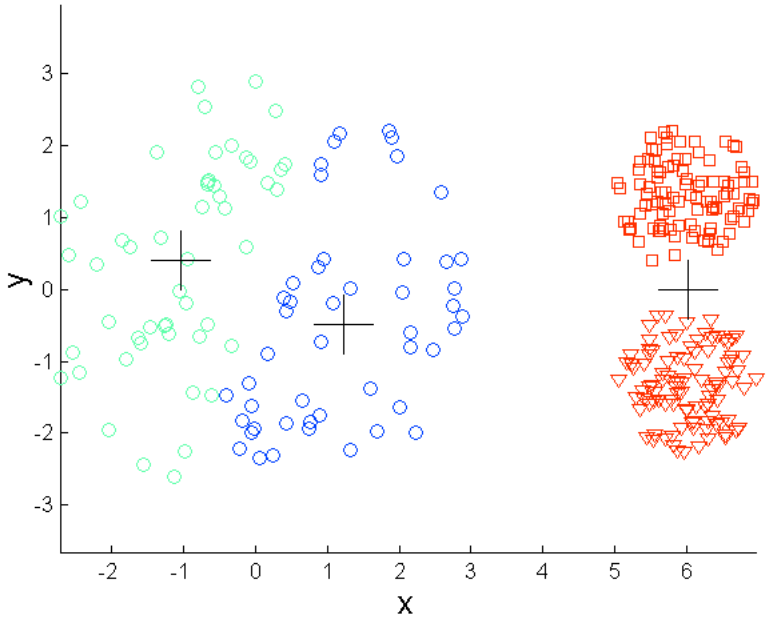


K-means (3 Clusters)

Limitations of K-means: Differing Density

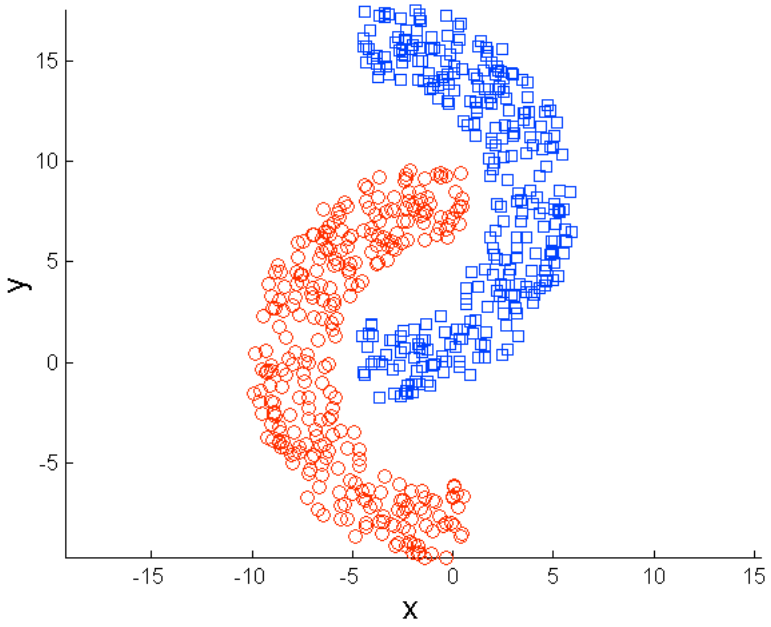


Original Points

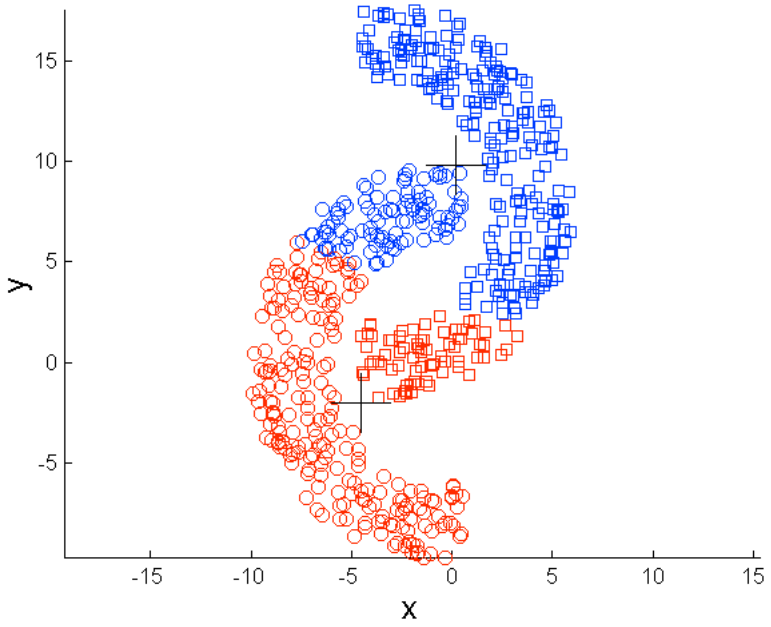


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

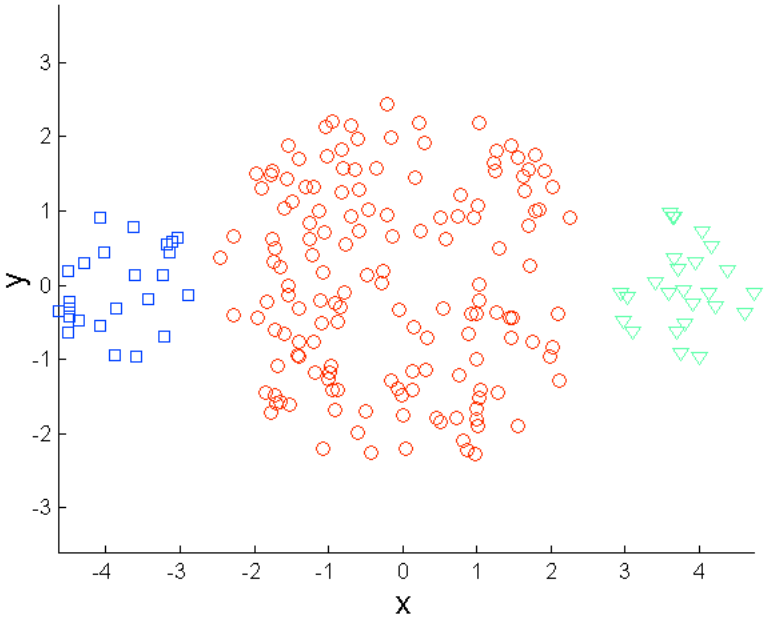


Original Points

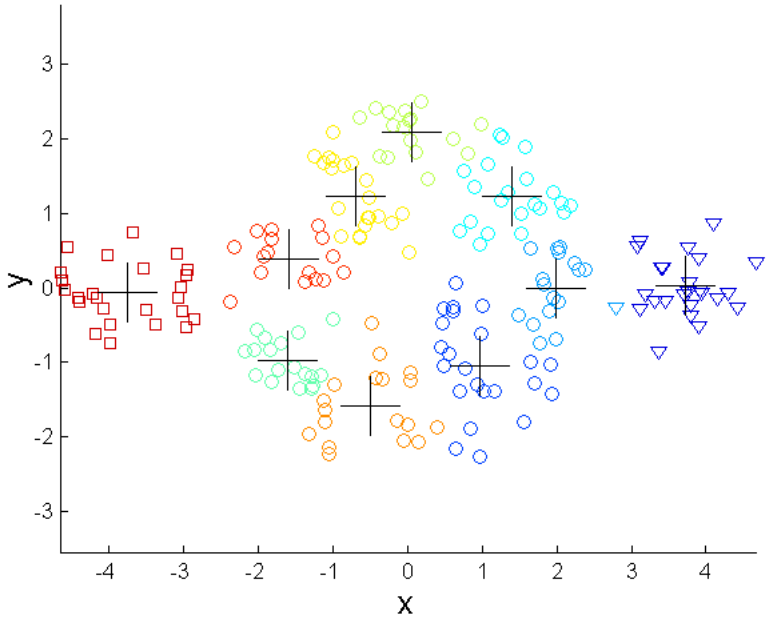


K-means (2 Clusters)

Overcoming K-means Limitations



Original Points

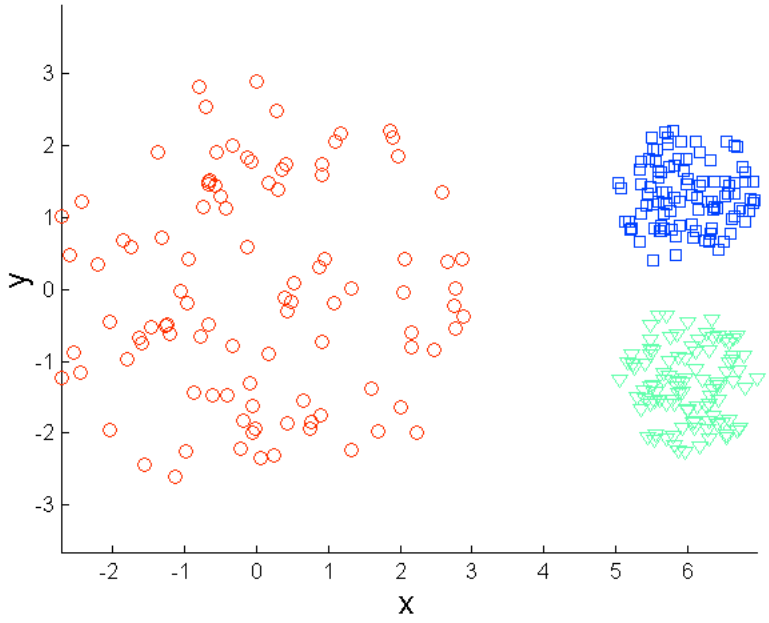


K-means Clusters

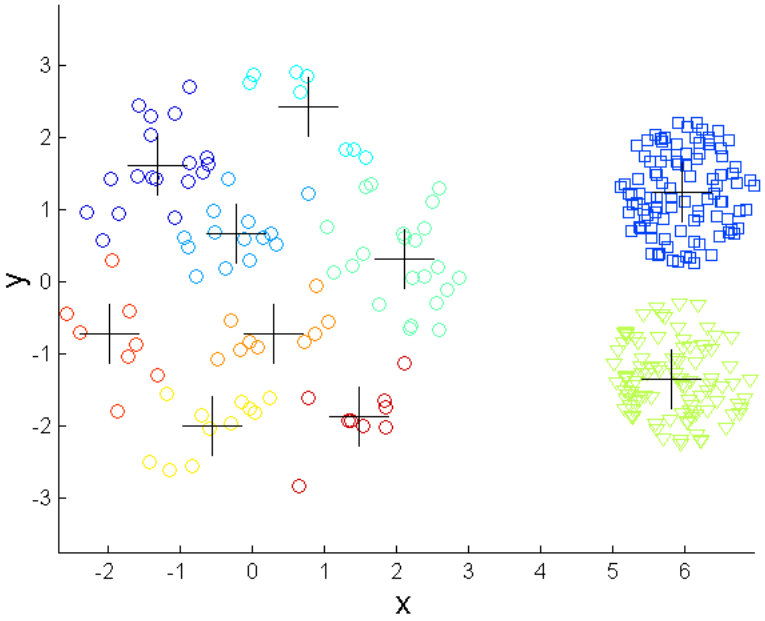
One solution is to use many clusters.

Find parts of clusters, but need to put together.

Overcoming K-means Limitations

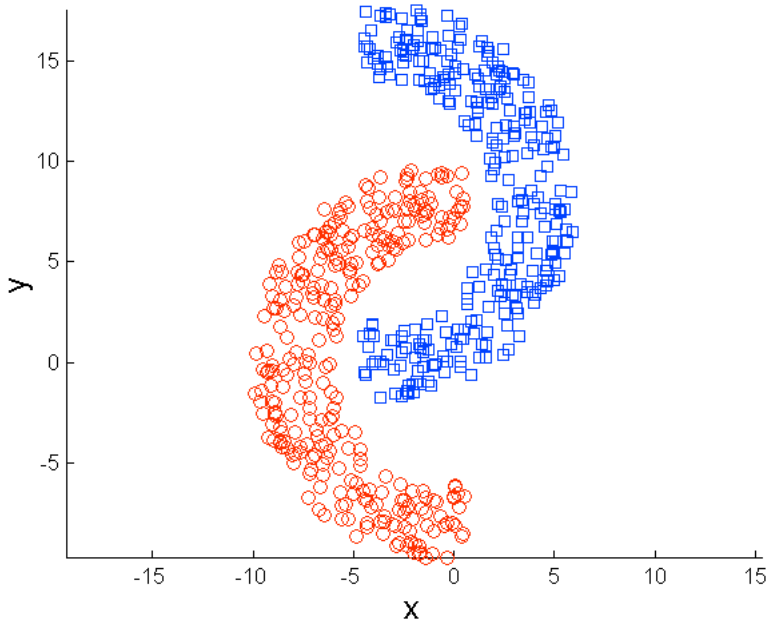


Original Points

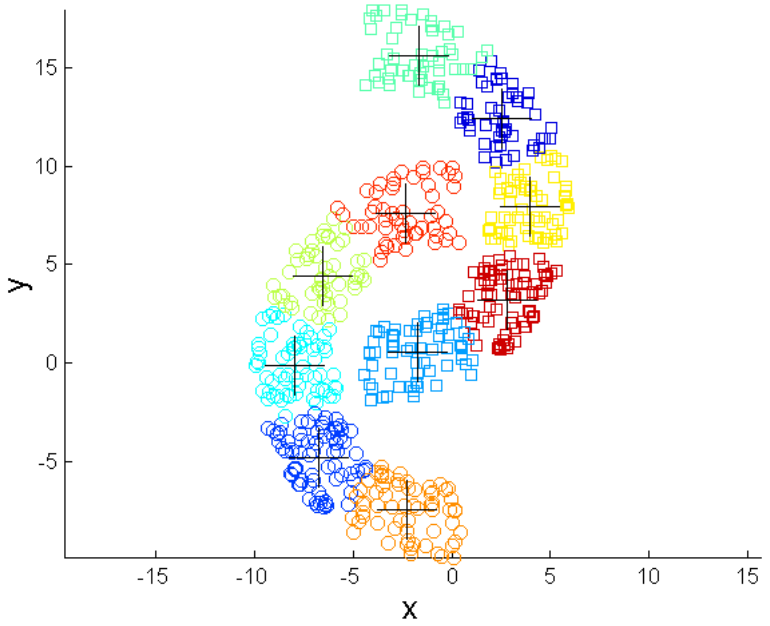


K-means Clusters

Overcoming K-means Limitations

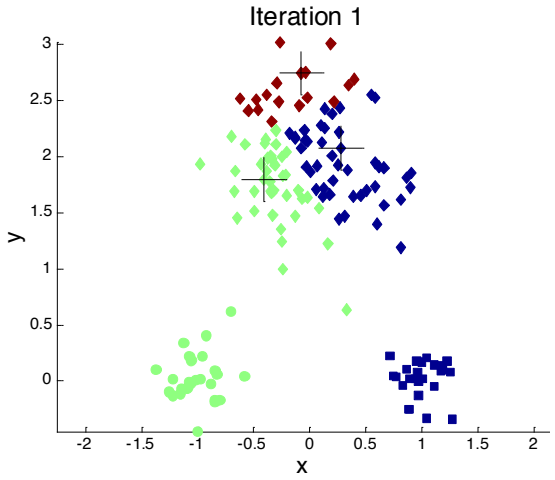


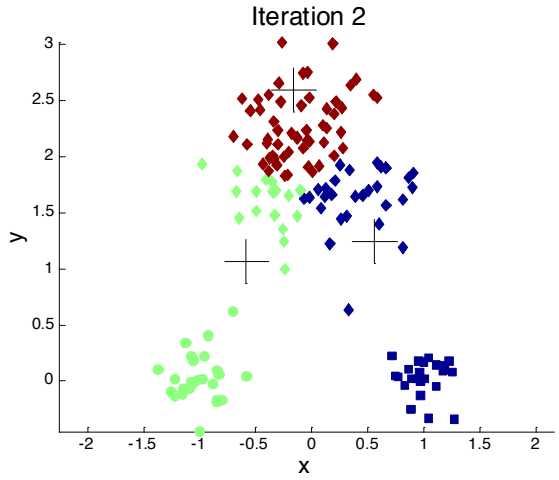
Original Points

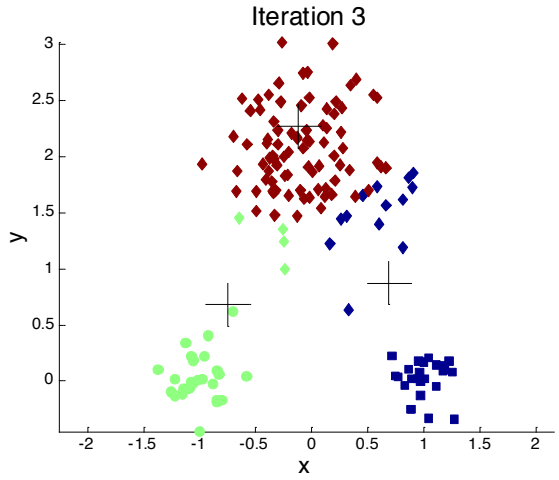


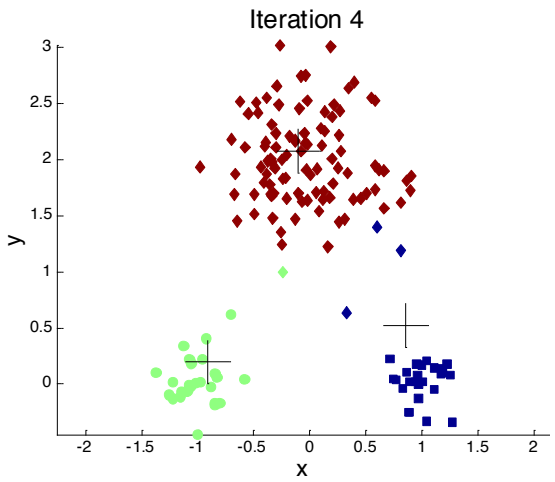
K-means Clusters

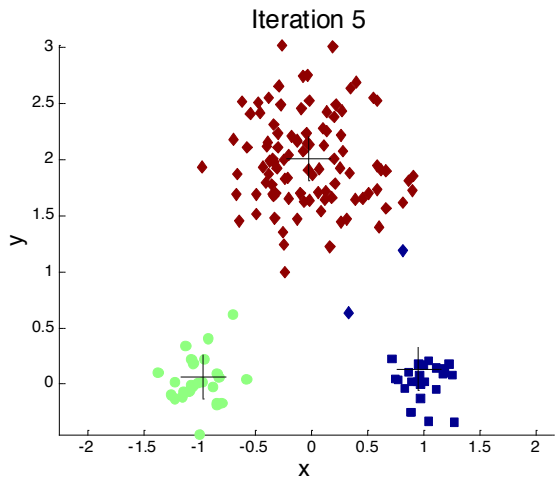
Importance of Choosing Initial Centroids

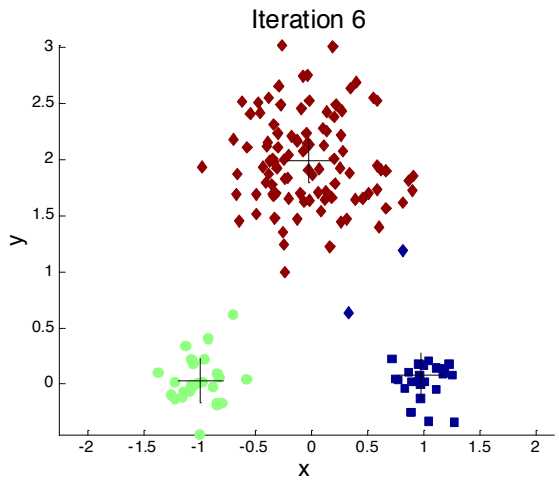








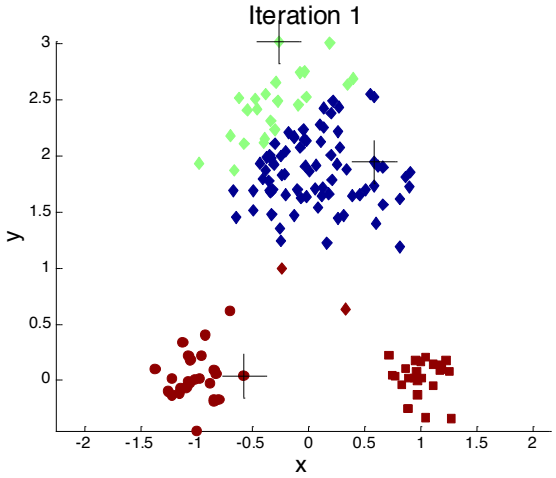


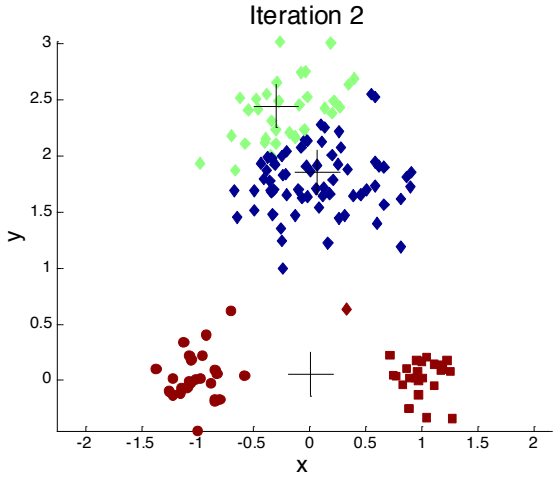


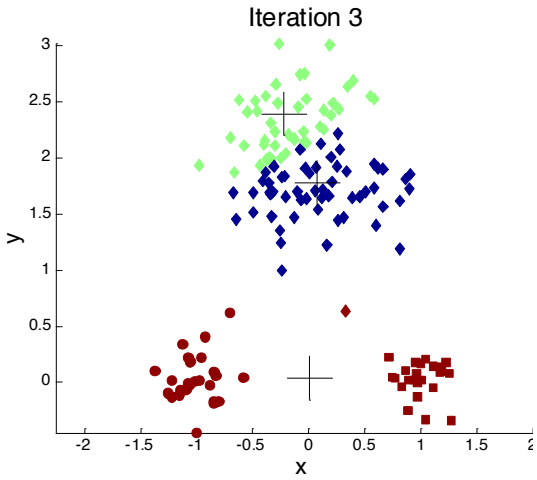
TRANSP-OR

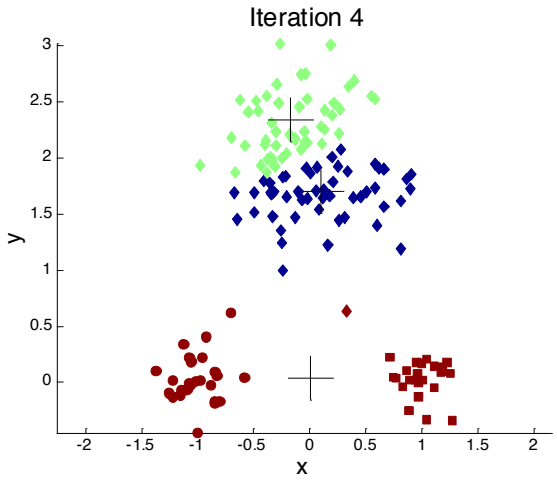
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

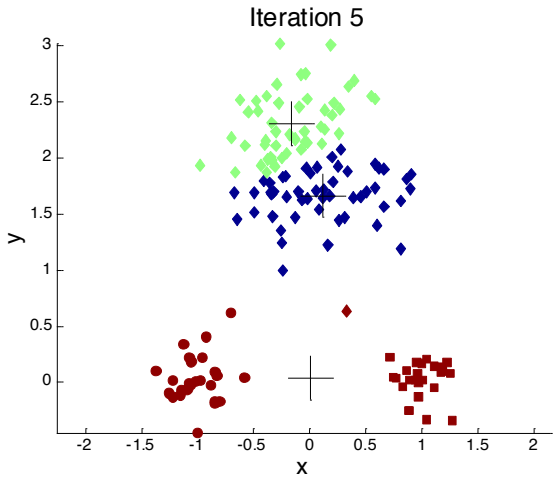
Importance of Choosing Initial Centroids











TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Solutions for Selecting Initial Points

- Multiple runs
  - Helps, but probability is not on your side
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated
  - Postprocessing – cluster merging
- K-means++
- Bisecting K-means avoids selection of initial clusters
  - In fact, using hierarchical clustering to determine initial centroids

K-Means++

- This approach can be slower than random initialization, but very consistently produces better results in terms of SSE
- To select a set of initial centroids, *C*, perform the following

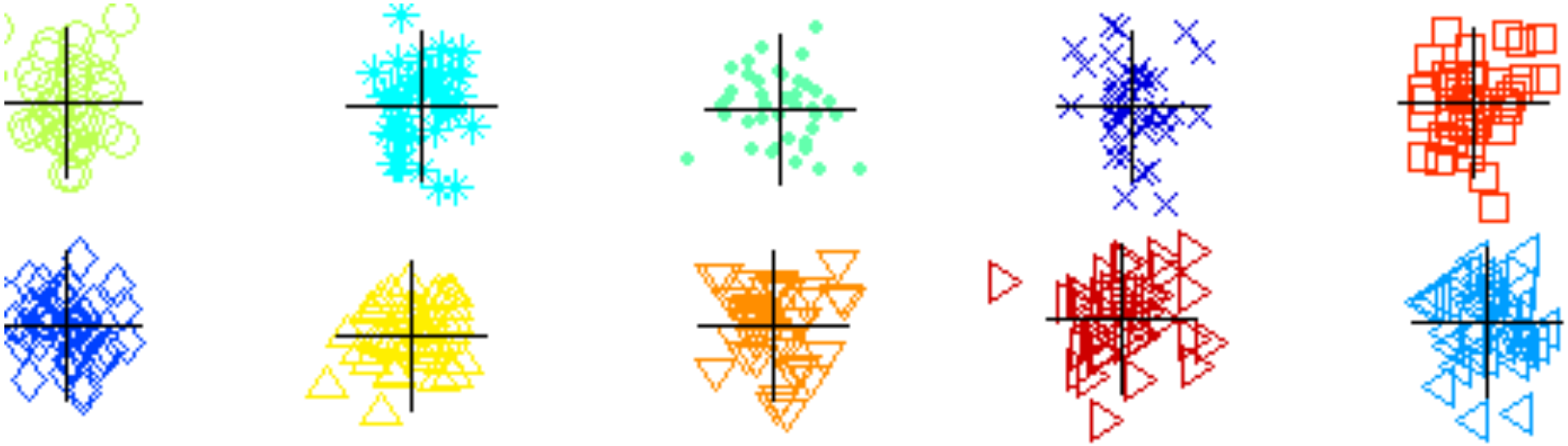
1. Select an initial point at random to be the first centroid
2. For k – 1 steps
3. For each of the N points, x_i, $1 \leq i \leq N$, find the minimum squared distance to the currently selected centroids, C_1, …, C_j, $1 \leq j < k$, i.e., $\min_j d^2(C_j, x_i)$
4. Randomly select a new centroid by choosing a point with probability proportional to $\frac{\min_j d^2(C_j, x_i)}{\sum_i \min_j d^2(C_j, x_i)}$
5. End For

Bisecting K-means

- Bisecting K-means algorithm
  - Variant of K-means that can produce a partitional or a hierarchical clustering

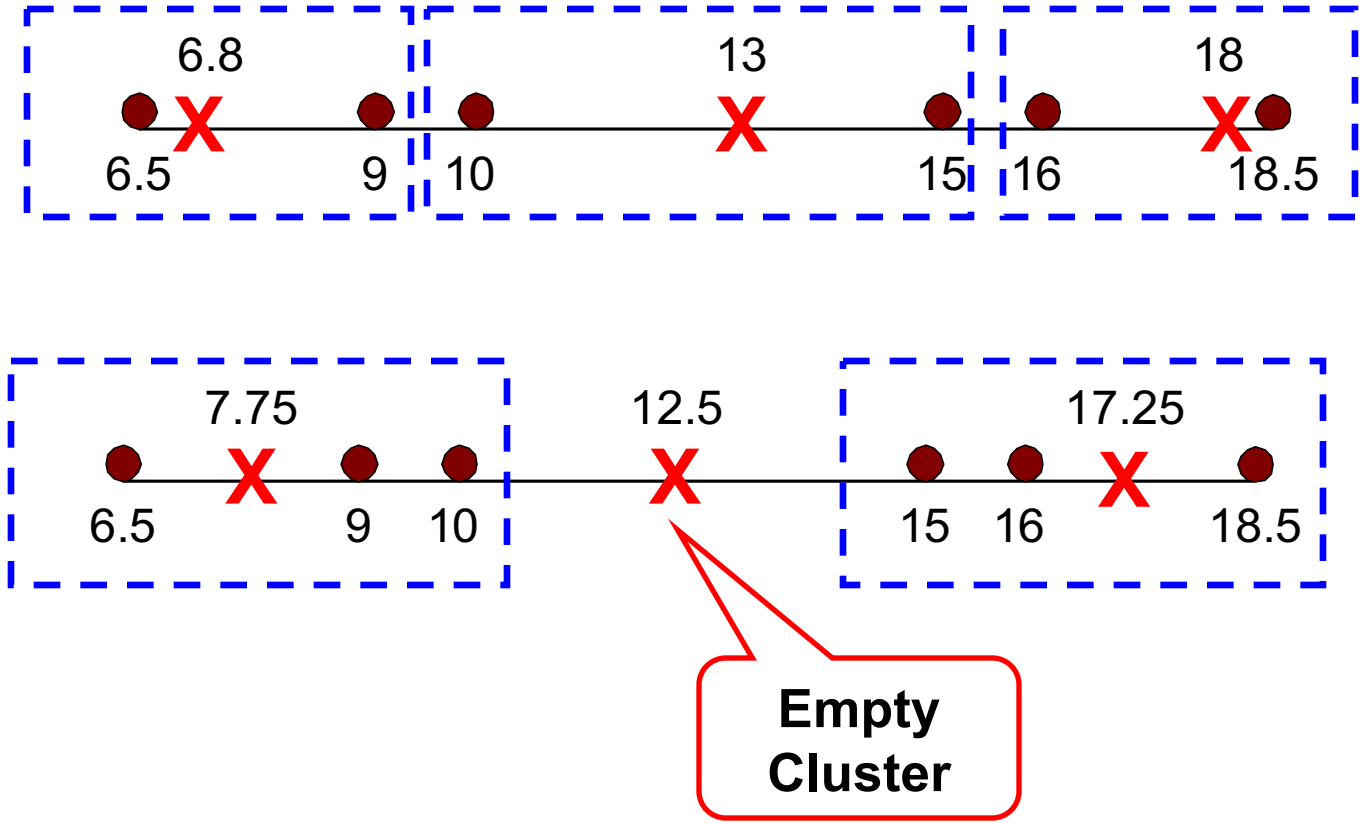
1: Initialize the list of clusters to contain the cluster containing all points.
2: **repeat**
3: Select a cluster from the list of clusters
4: **for** $i = 1$ to *number_of_iterations* **do**
5: Bisect the selected cluster using basic K-means
6: **end for**
7: Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: **until** Until the list of clusters contains K clusters

Bisecting K-means Example



Empty Clusters

- K-means can yield empty clusters



Empty Clusters Handling

- Basic K-means algorithm can yield empty clusters

- Several strategies
  - Choose the point that contributes most to SSE
  - Choose a point from the cluster with the highest SSE
  - If there are several empty clusters, the above can be repeated several times.

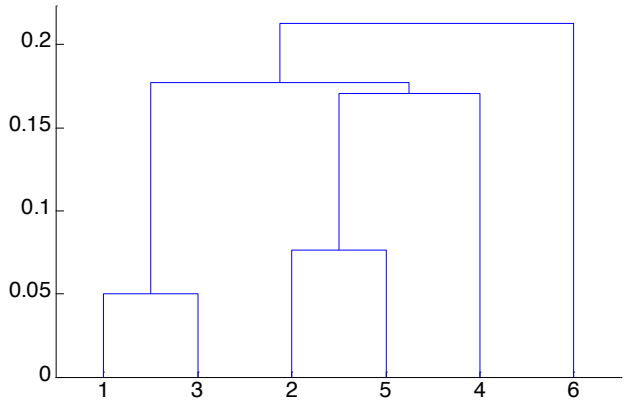
Pre-processing and Post-processing

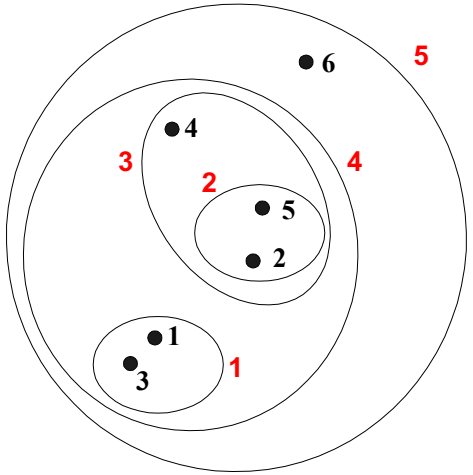
- Pre-processing
  - Normalize the data
  - Eliminate outliers

- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE

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 dendrogram at the proper level

- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom)

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) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains an individual point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

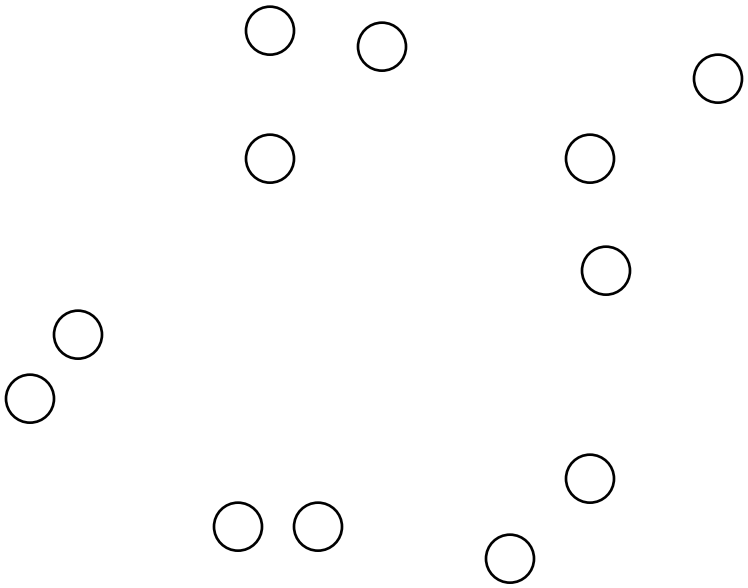
Agglomerative Clustering Algorithm

- Most 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

Starting Situation

- Start with clusters of individual points and a proximity matrix

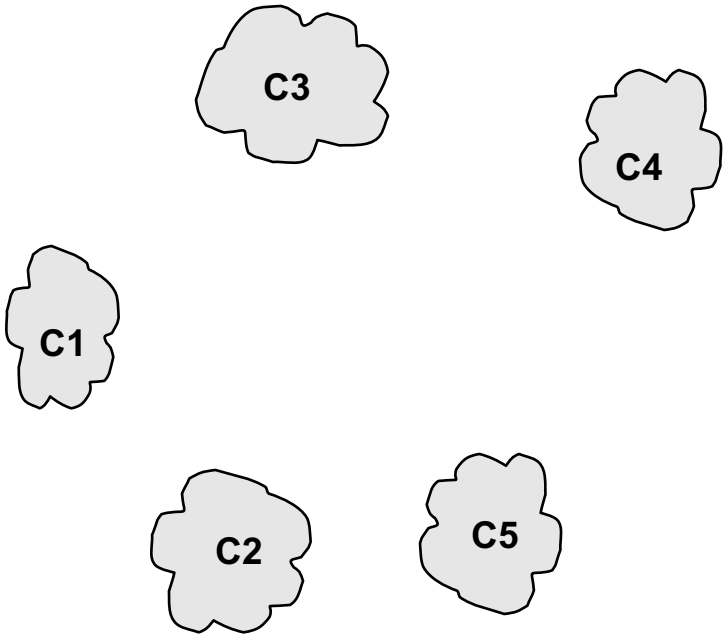


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

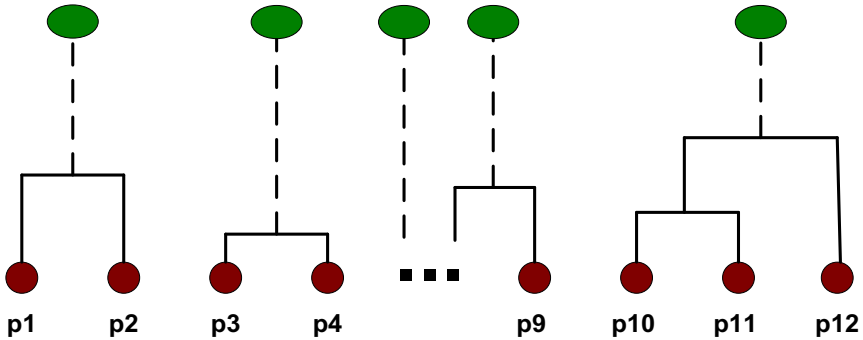
Intermediate Situation

- After some merging steps, we have some clusters



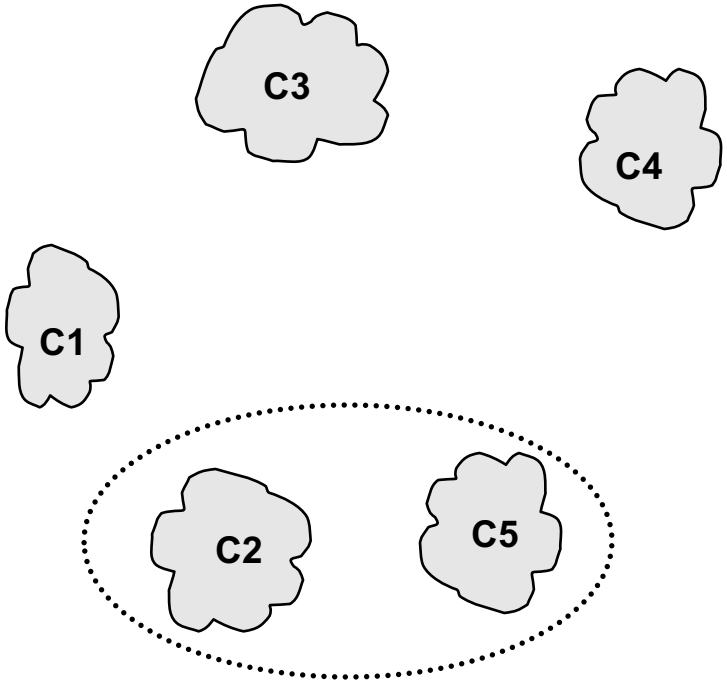
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



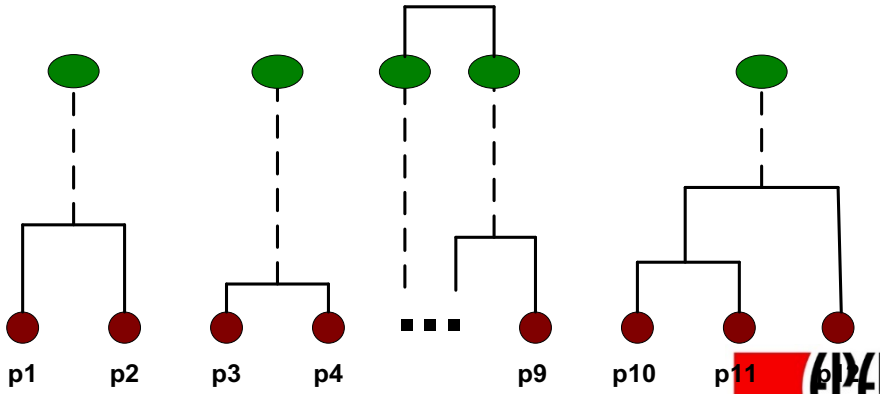
Intermediate Situation

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



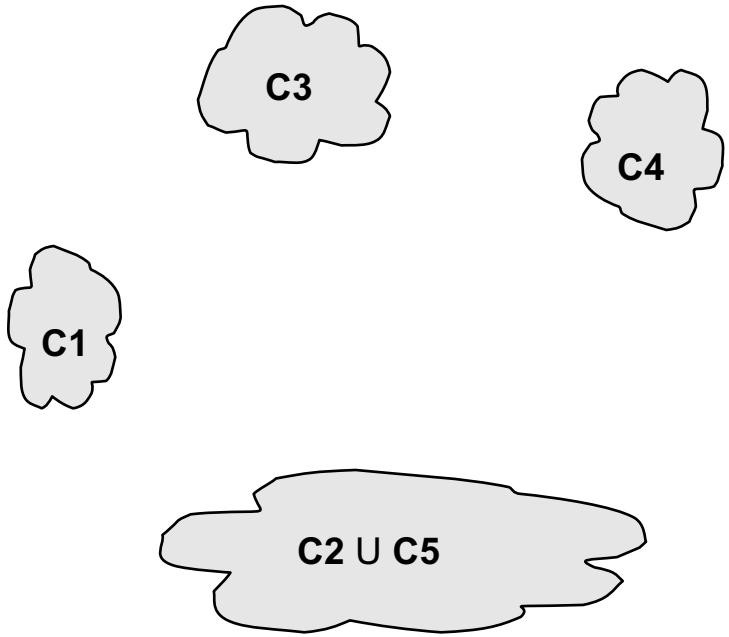
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



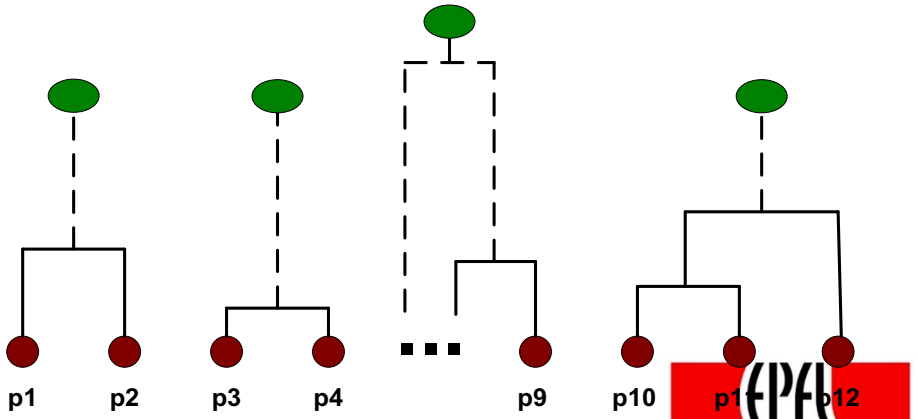
After Merging

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



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

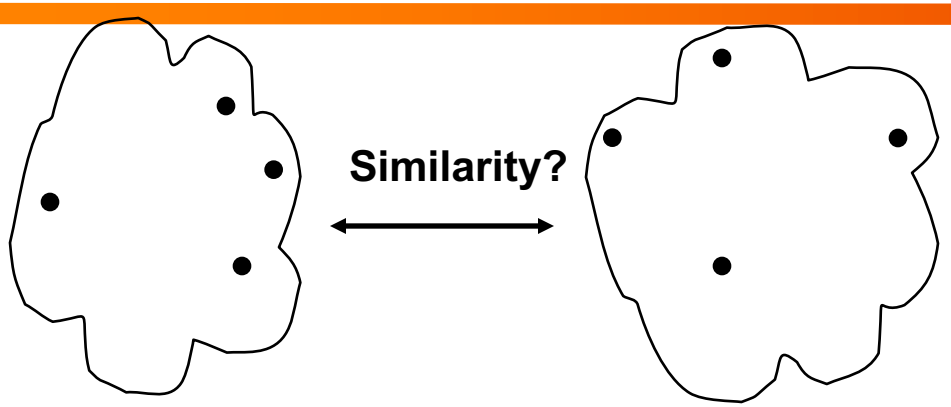
Proximity Matrix



TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

How to Define Inter-Cluster Distance



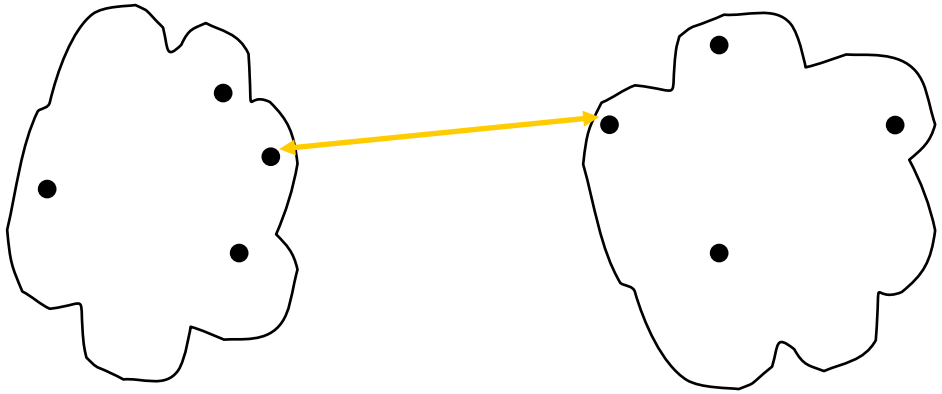
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

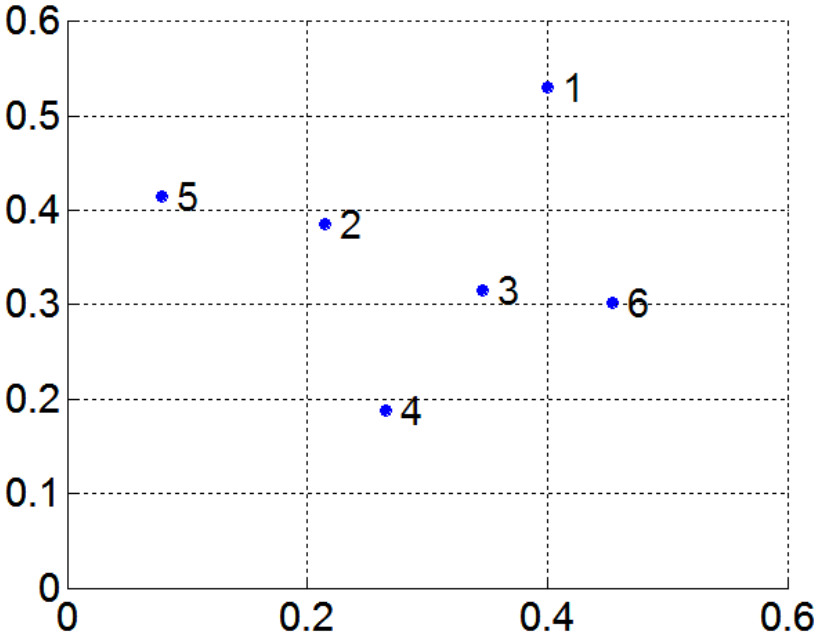
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

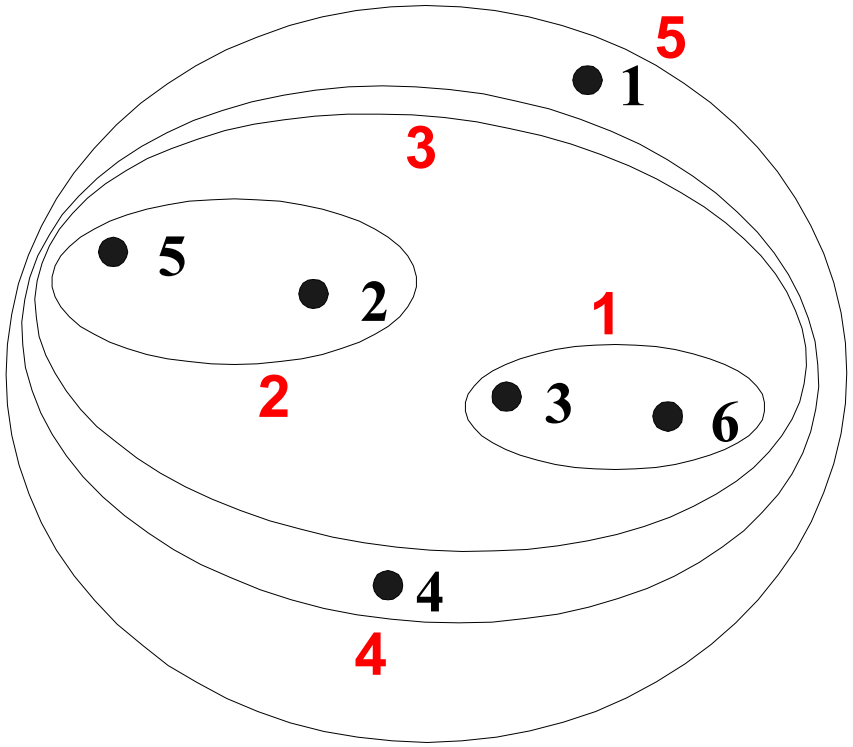
MIN or Single Link

- Proximity of two clusters is based on the two closest points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph



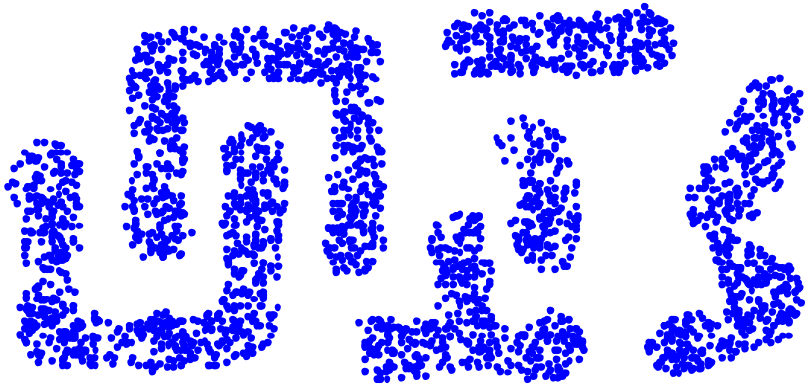
Hierarchical Clustering: MIN



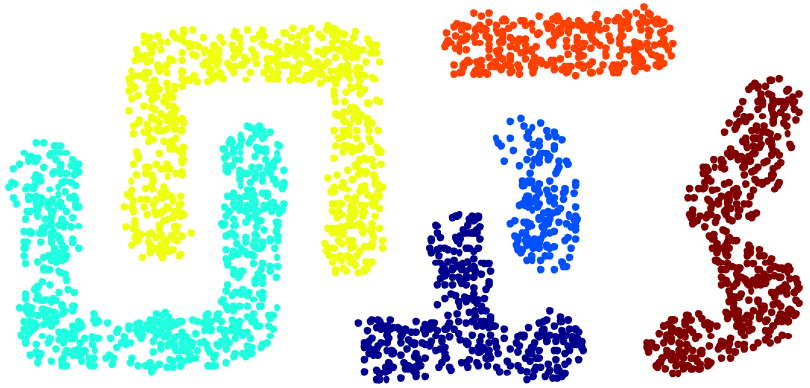


Nested Clusters

Strength of MIN



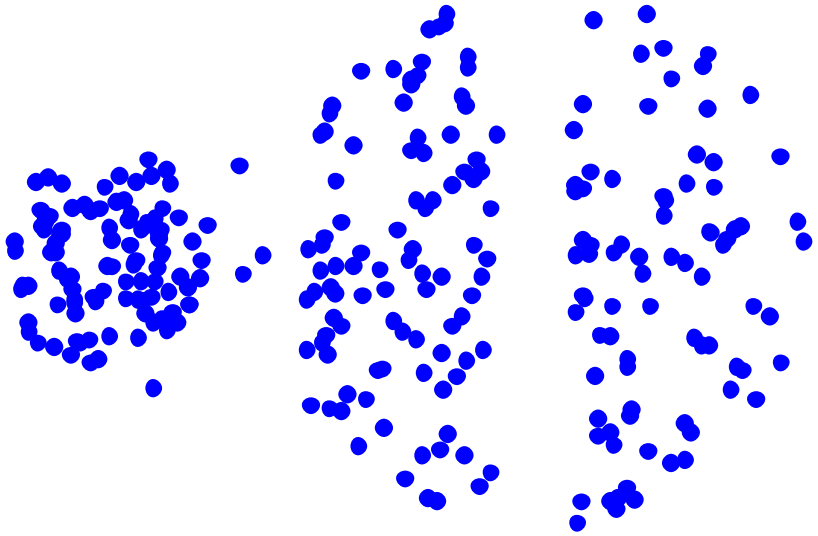
Original Points



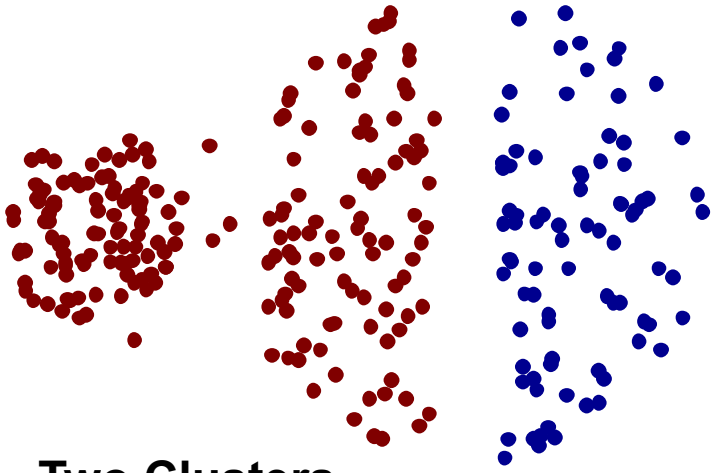
Six Clusters

- **Can handle non-elliptical shapes**

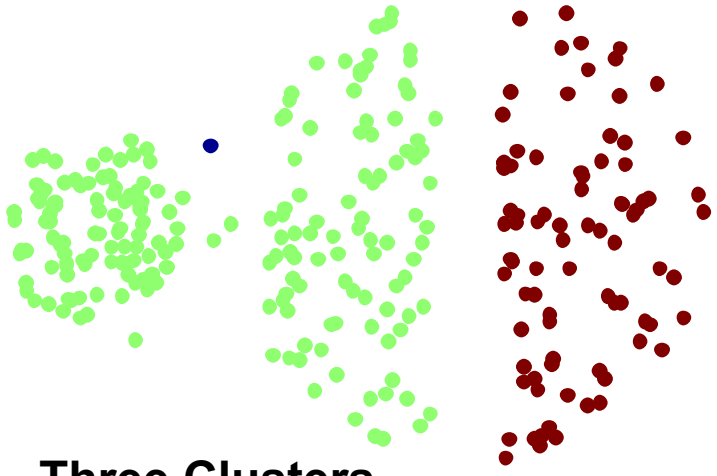
Limitations of MIN



Original Points



Two Clusters

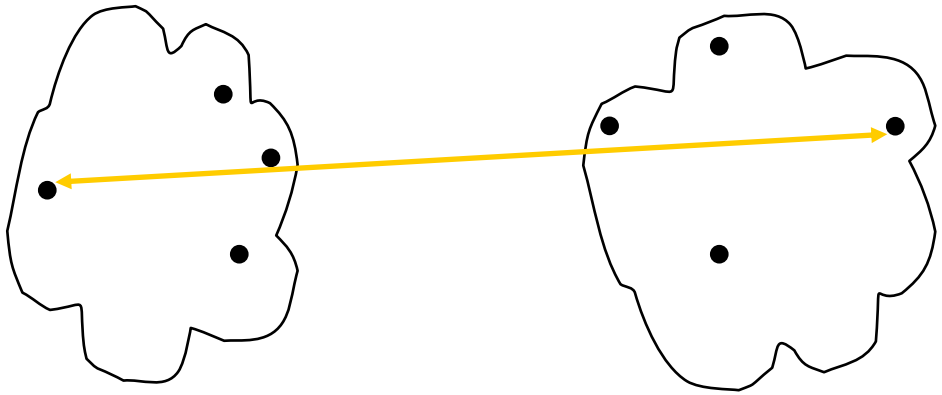


Three Clusters

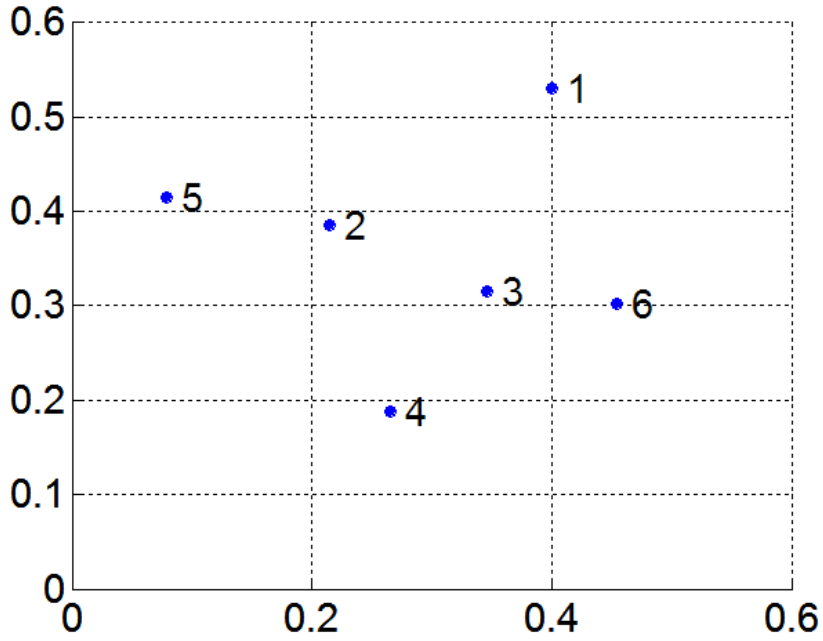
- **Sensitive to noise and outliers**

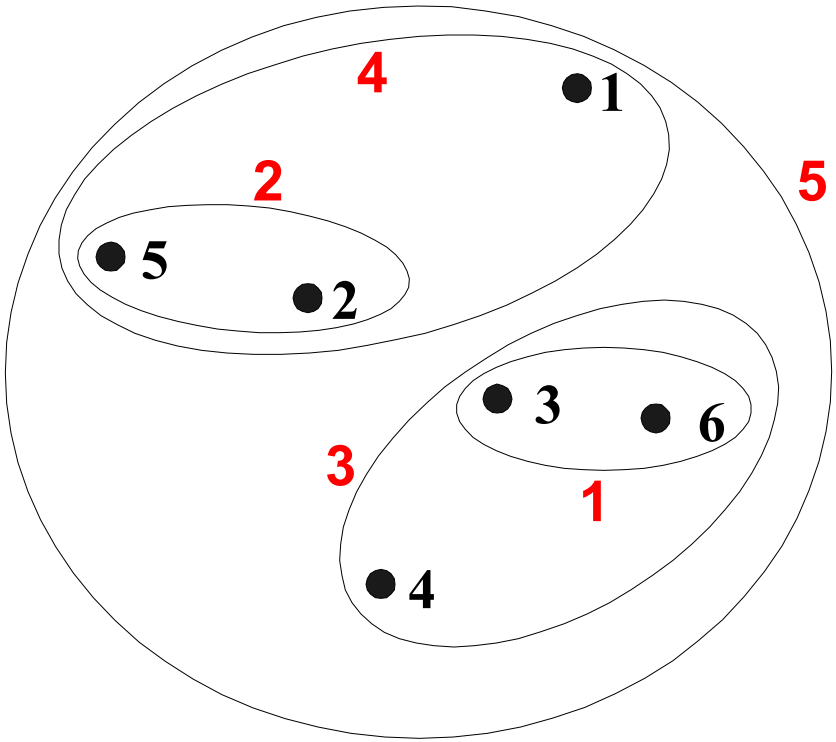
MAX or Complete Linkage

- Proximity of two clusters is based on the two most distant points in the different clusters
  - Determined by all pairs of points in the two clusters



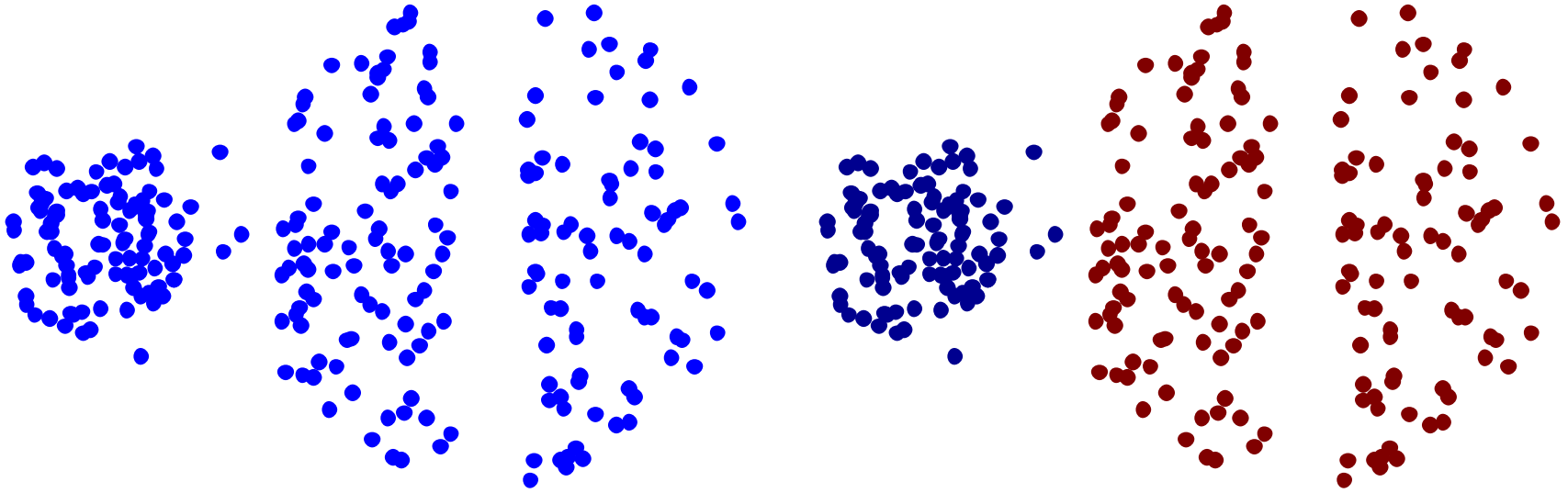
Hierarchical Clustering: MAX





Nested Clusters

Strength of MAX

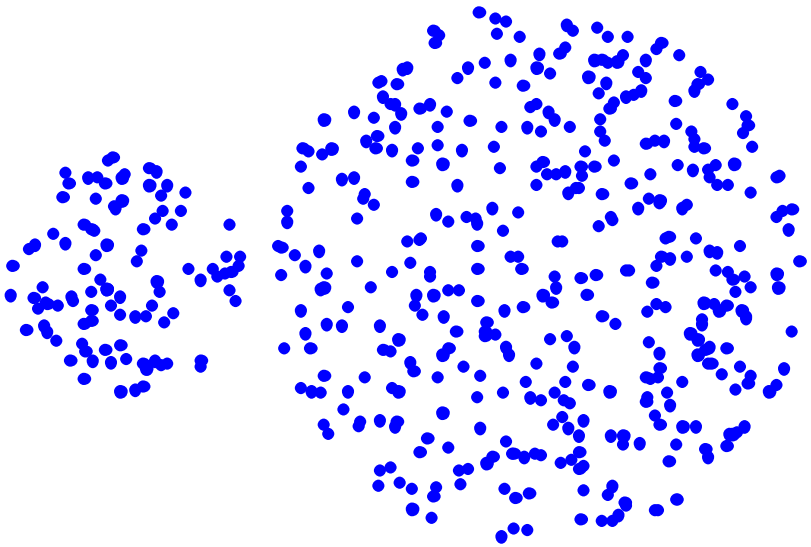


Original Points

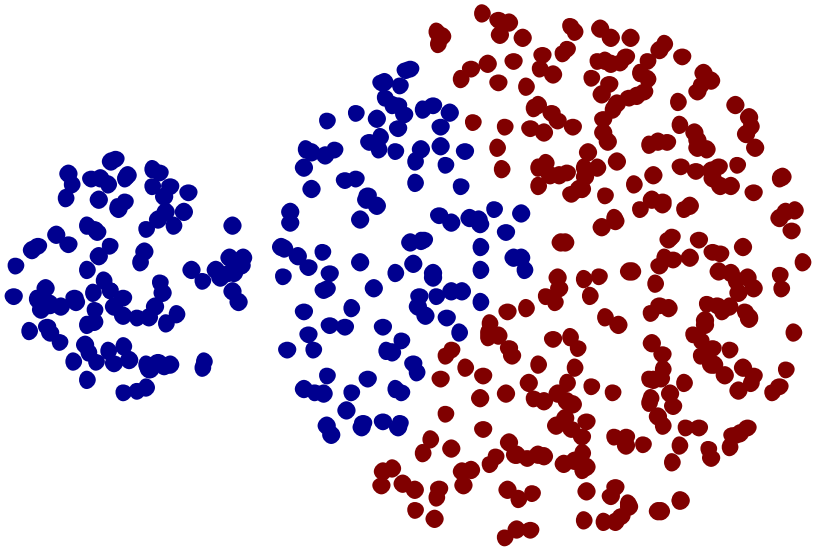
Two Clusters

- **Less susceptible to noise and outliers**

Limitations of MAX



Original Points



Two Clusters

- **Tends to break large clusters (globular clusters mostly)**

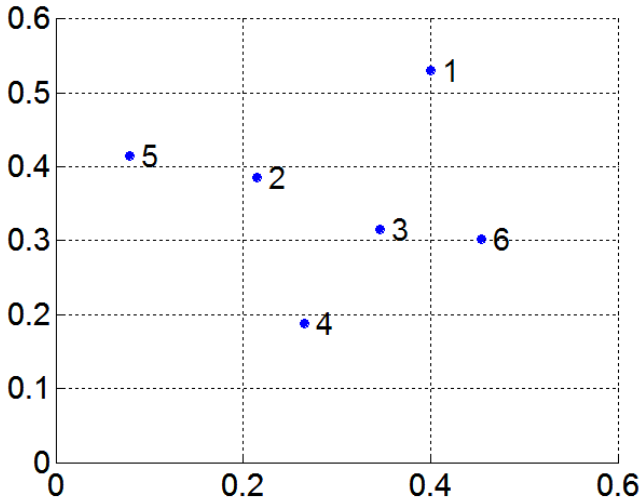
Group Average

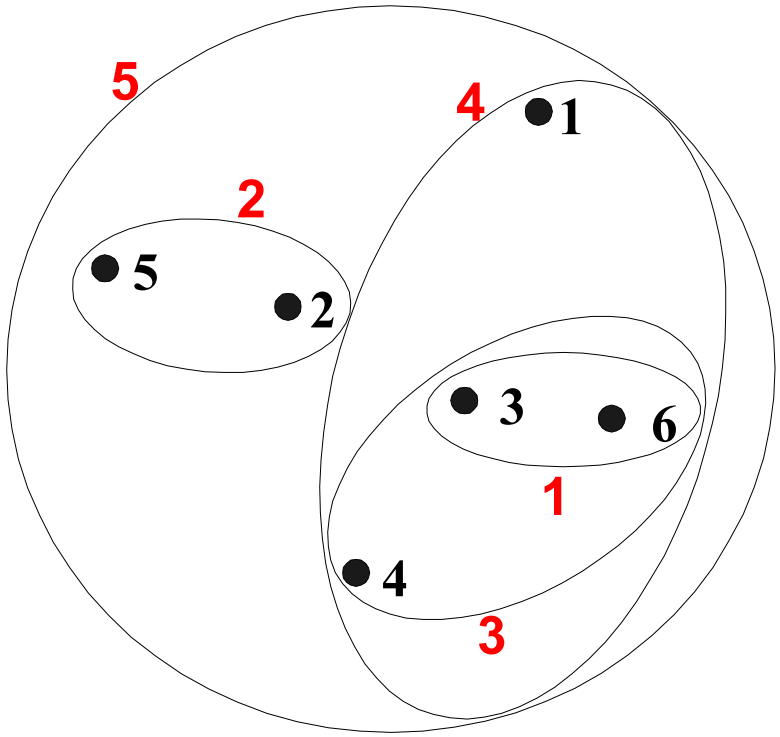
- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

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

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

Hierarchical Clustering: Group Average





Nested Clusters

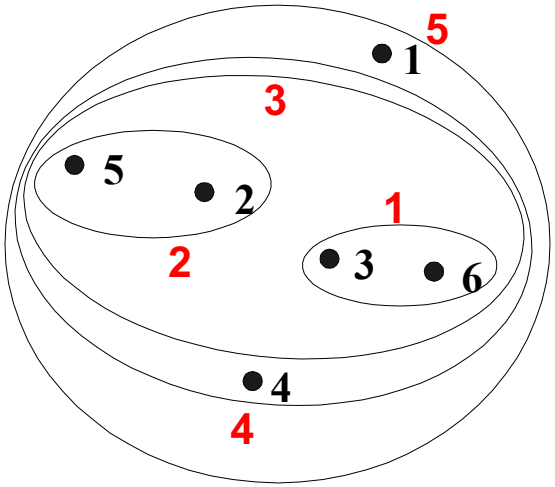
Group Average – Pros and Cons

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

Cluster Similarity: Ward's Method

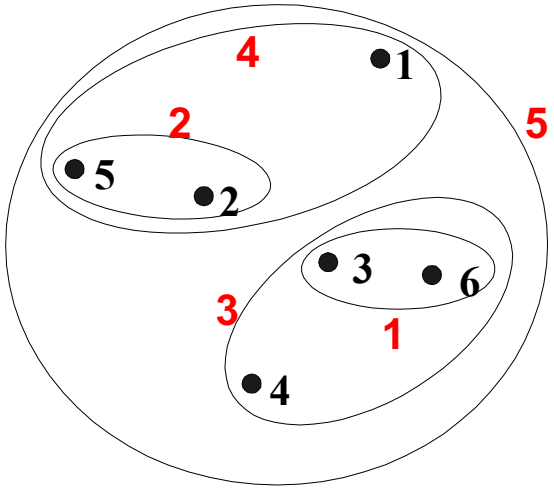
- Similarity of two clusters is based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
  - Can be used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

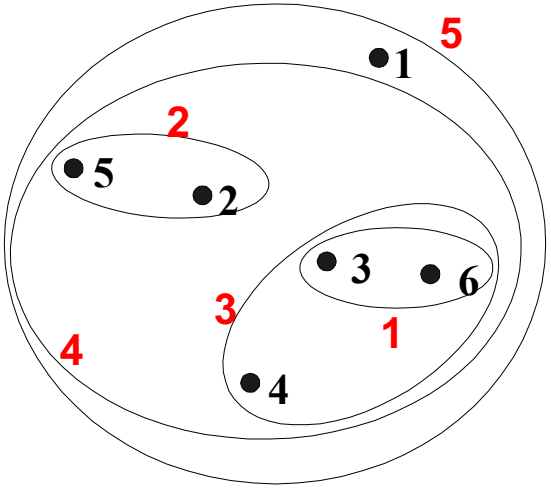
Hierarchical Clustering: Comparison



MIN

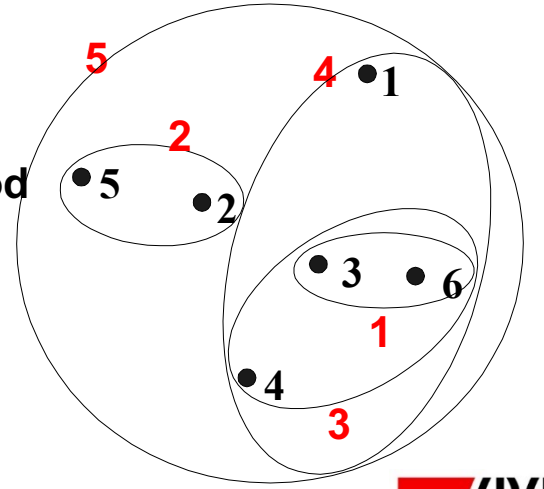
MAX





Group Average

Ward's Method

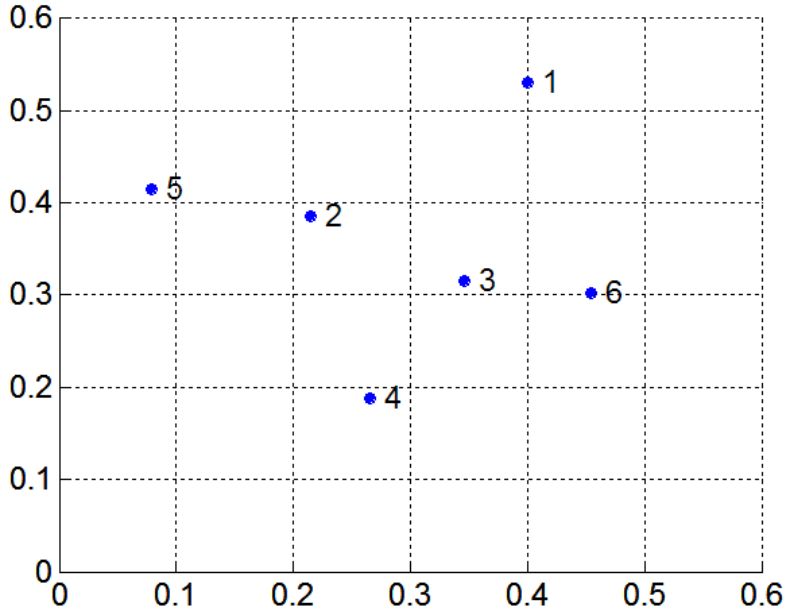


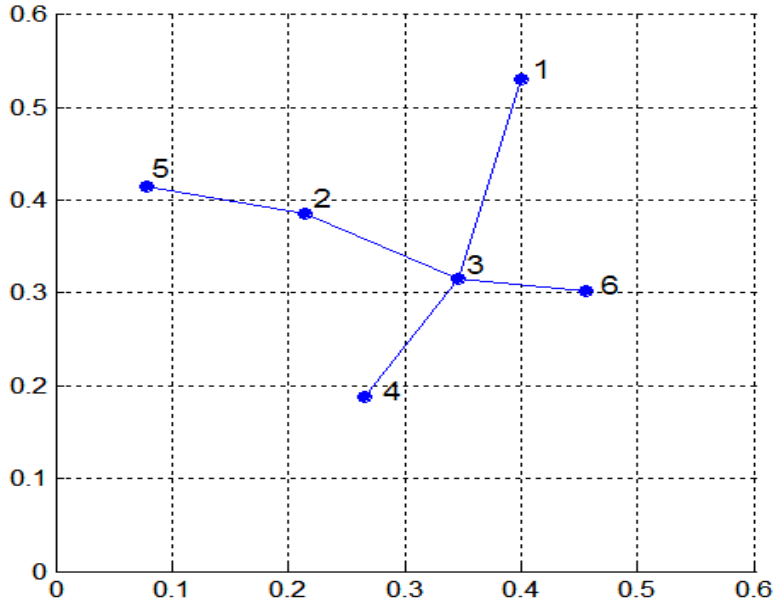
TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

MST: Divisive Hierarchical Clustering

- Build MST (Minimum Spanning Tree)
  - Start with a tree that consists of any point
  - In successive steps, look for the closest pair of points (p, q) such that one point (p) is in the current tree but the other (q) is not
  - Add q to the tree and put an edge between p and q





MST: Divisive Hierarchical Clustering

- Use MST for constructing hierarchy of clusters

Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

1: Compute a minimum spanning tree for the proximity graph.
2: **repeat**
3: Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
4: **until** Only singleton clusters remain

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 proximity matrix, of size N^2, must be updated and searched

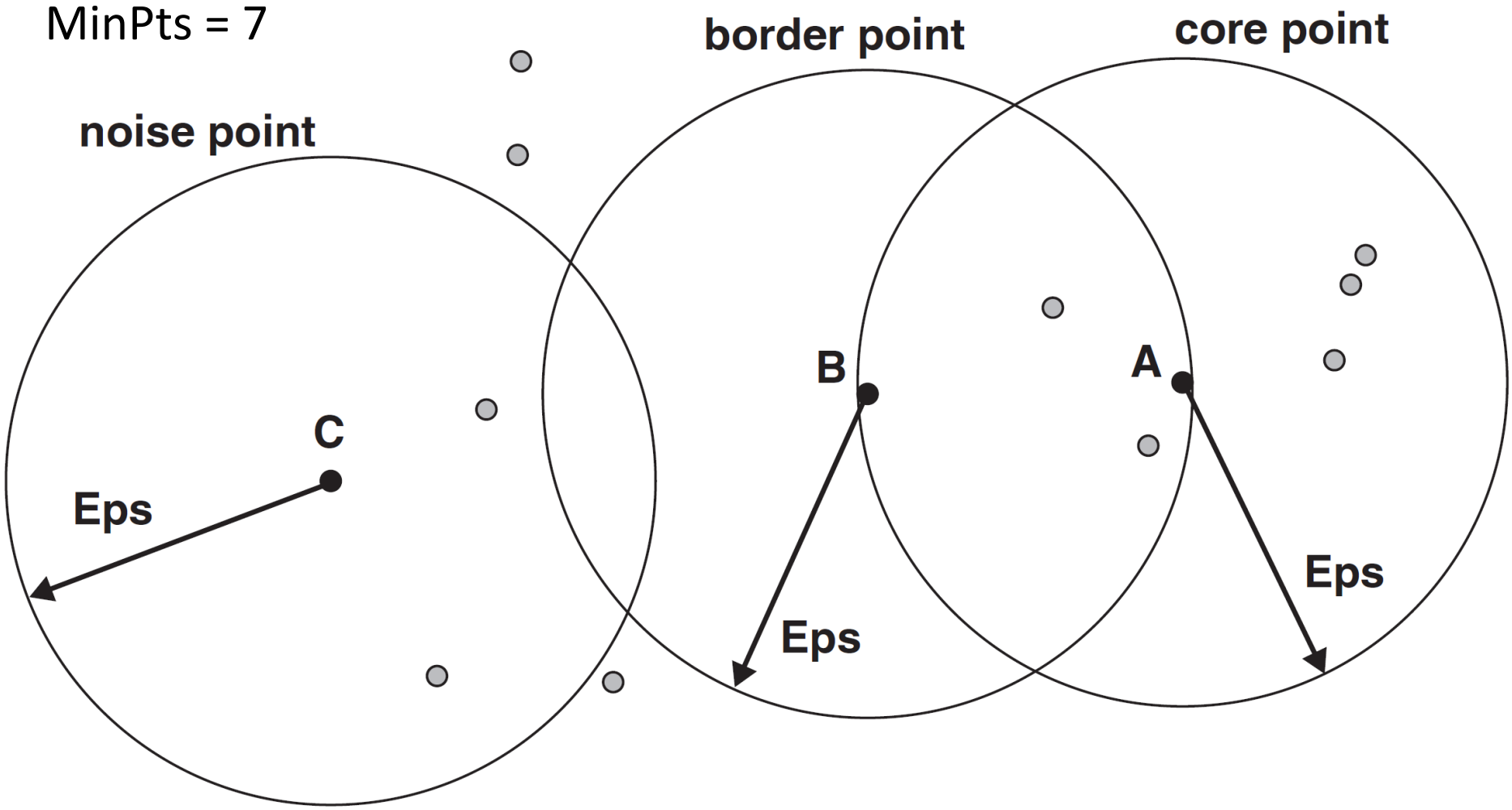
Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone
- No global objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling clusters of different sizes and non-globular shapes
  - Breaking large clusters

DBSCAN

- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius (Eps)
  - 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 nor a border point

DBSCAN: Core, Border, and Noise Points



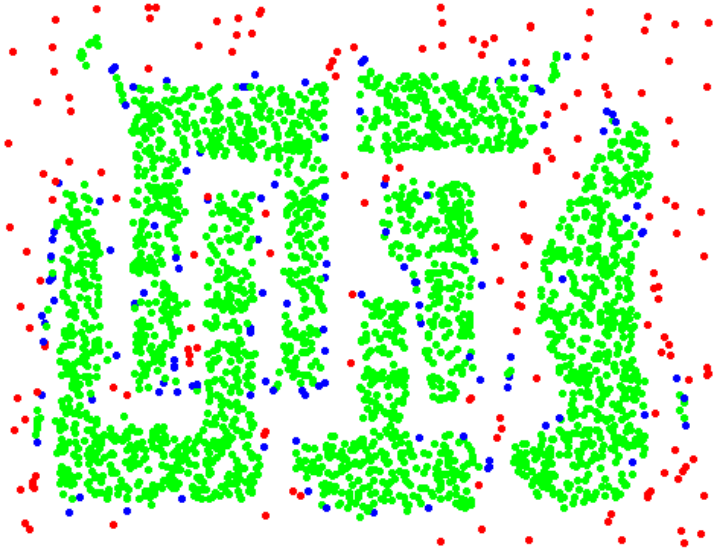
DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

current_cluster_label $\leftarrow 1$
for all core points **do**
  if the core point has no cluster label **then**
    current_cluster_label $\leftarrow$ *current_cluster_label* $+ 1$
    Label the current core point with cluster label *current_cluster_label*
  end if
  for all points in the *Eps*-neighborhood, except i^{th} the point itself **do**
    if the point does not have a cluster label **then**
      Label the point with cluster label *current_cluster_label*
    end if
  end for
end for

DBSCAN: Core, Border and Noise Points

Original Points

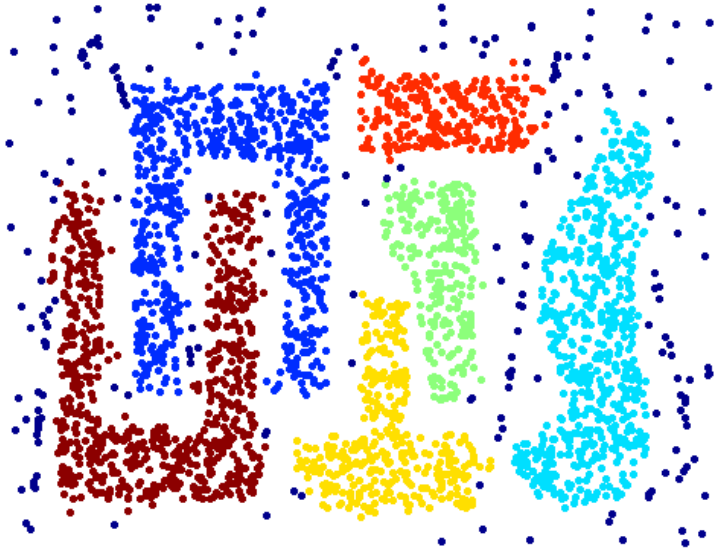


Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well

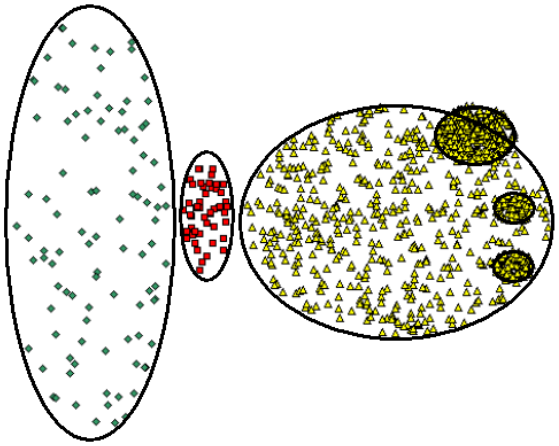
Original Points



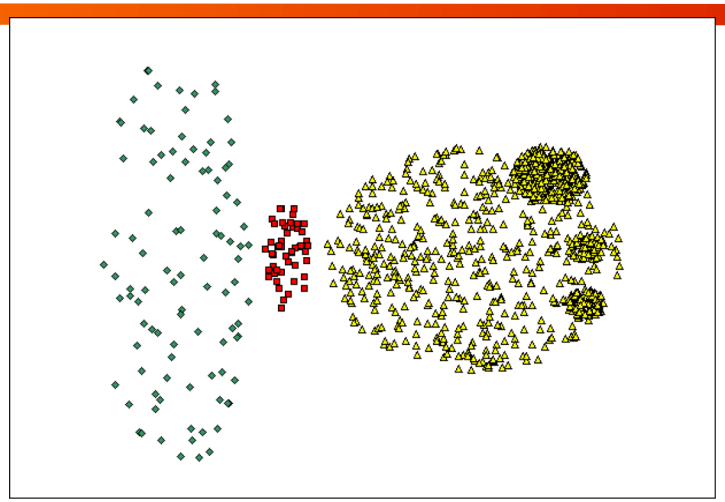
Clusters

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

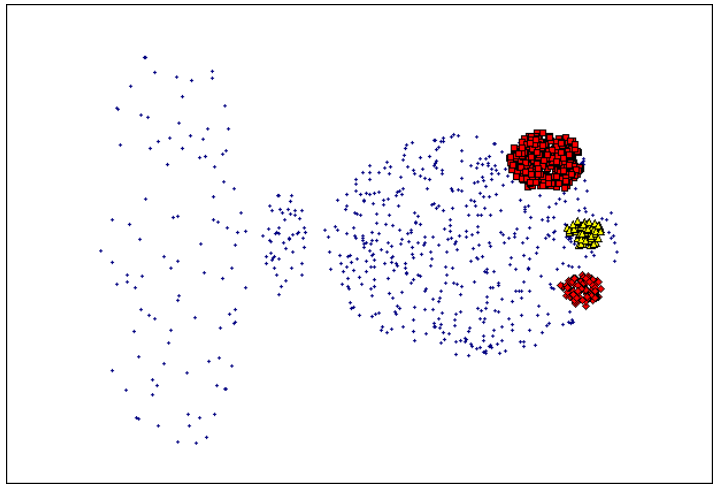
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).

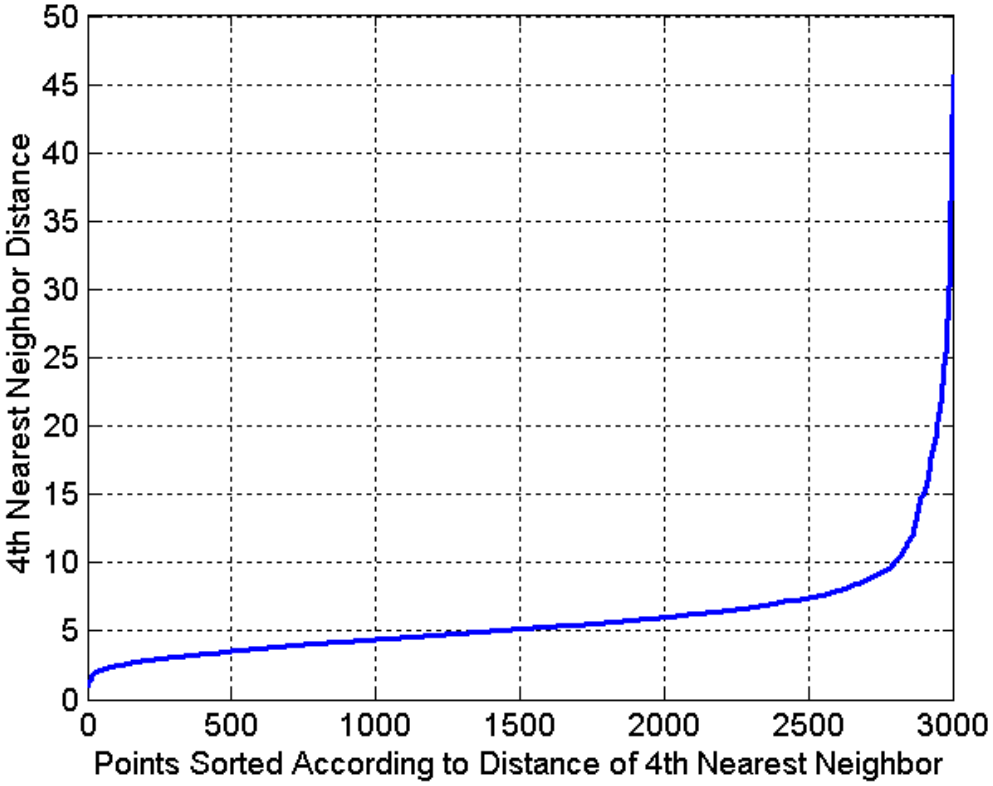


(MinPts=4, Eps=9.92)

- **Varying densities**

DBSCAN: Determining EPS and MinPts

- Idea is that 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
- So, plot sorted distance of every point to its k^{th} nearest neighbor

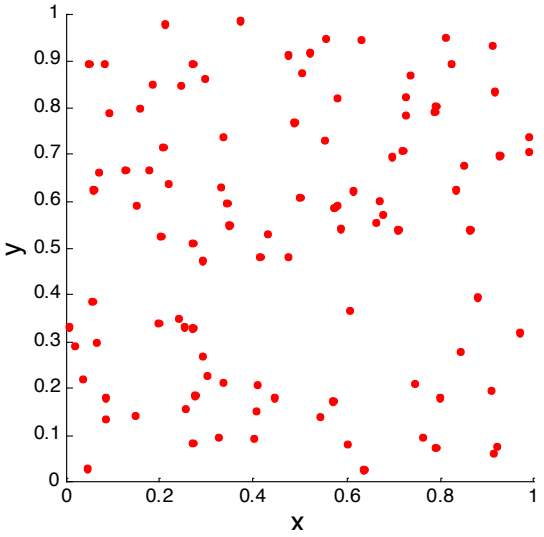


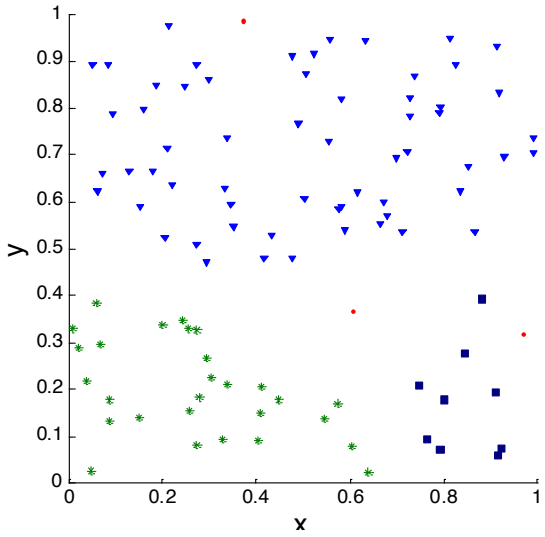
DBSCAN: Determining EPS and MinPts

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall
- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?
- But “clusters are in the eye of the beholder”!
- Then why do we want to evaluate them?
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Clusters found in Random Data

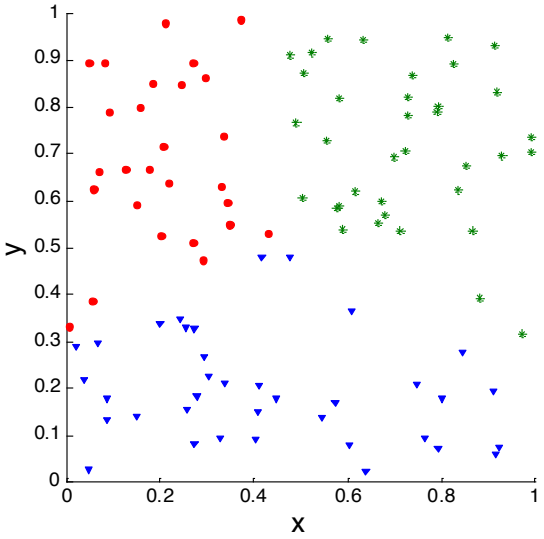
Random Points

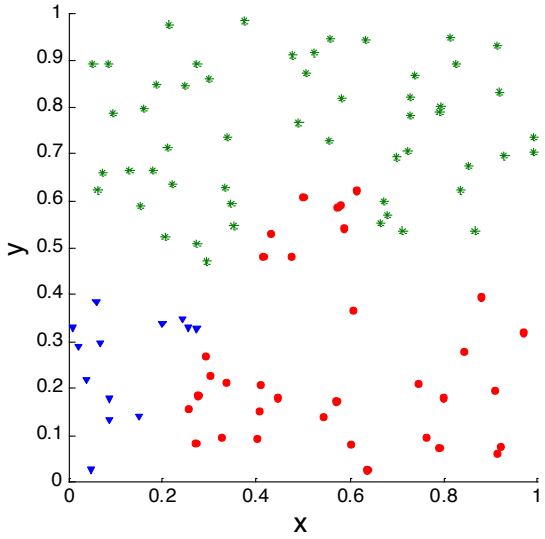




DBSCAN

K-means





Complete Link

Different Aspects of Cluster Validation

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Comparing the results of two different sets of cluster analyses to determine which is better.
4. Determining the ‘correct’ number of clusters.

- For 2 and 3, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Measures of Cluster Validity

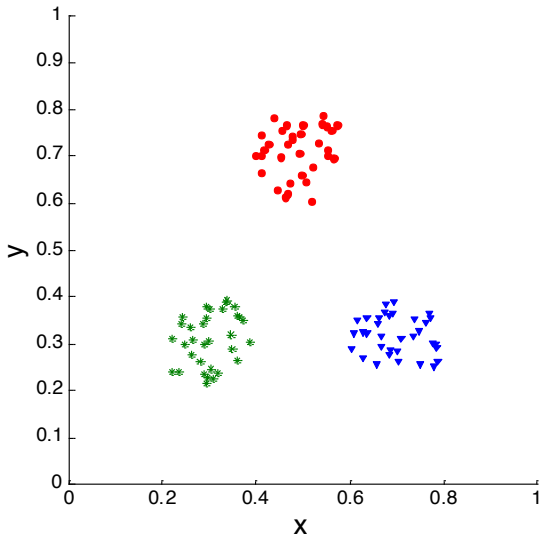
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - **External Index**: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - **Internal Index**: Used to measure the goodness of a clustering structure without respect to external information.
    - Sum of Squared Error (SSE)
  - **Relative Index**: Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as **criteria** instead of **indices**
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

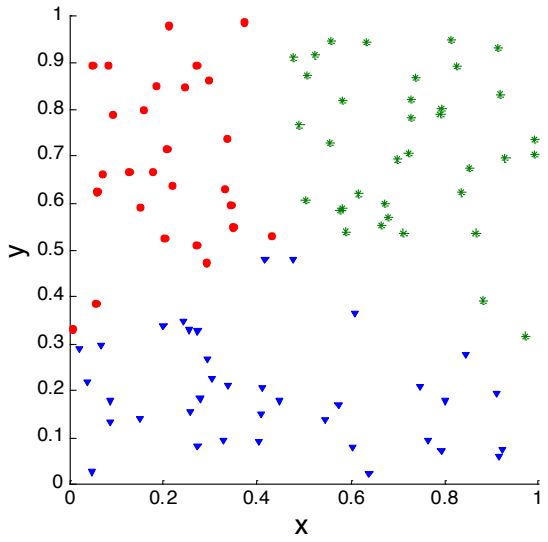
Measuring Cluster Validity Via Correlation

- Two matrices
  - Proximity Matrix
  - Ideal Similarity Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
  - Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density or contiguity based clusters.

Measuring Cluster Validity Via Correlation

- Correlation of ideal similarity and proximity matrices for the K-means clusterings of the following two data sets.





Corr = 0.9235

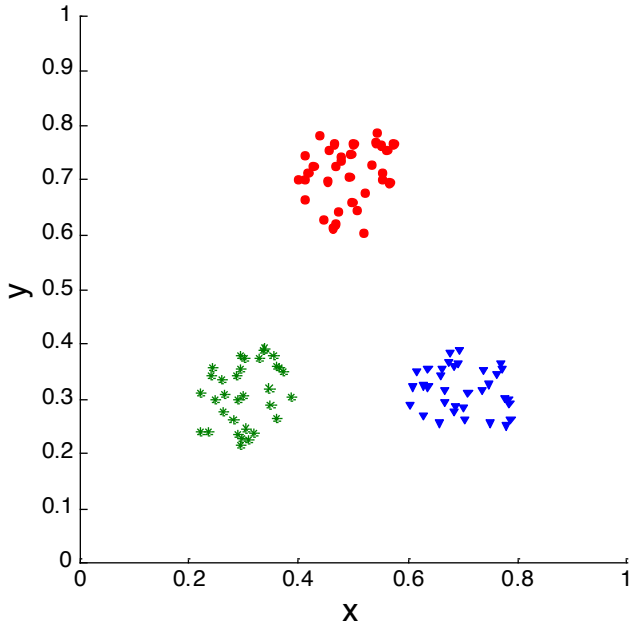
Corr = 0.5810

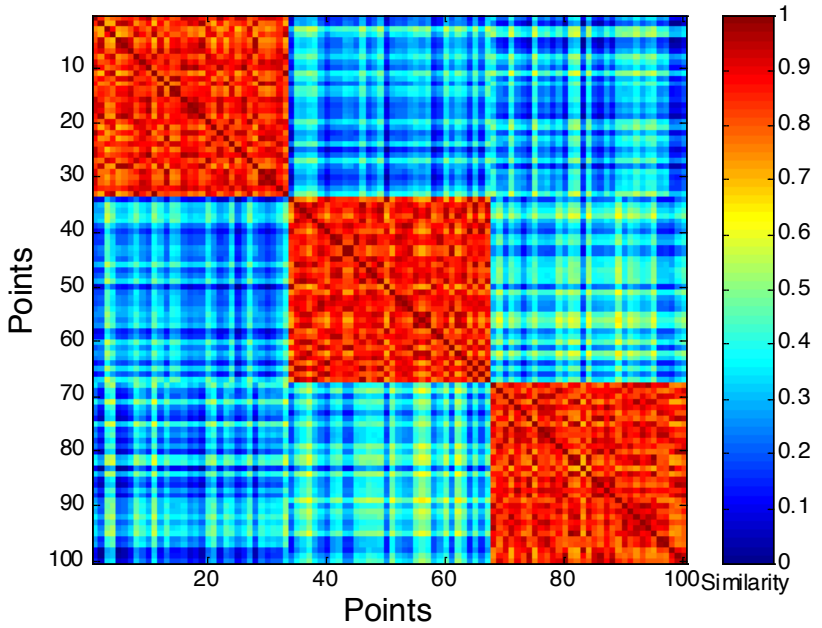
TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Using Similarity Matrix for Cluster Validation

- Order the similarity matrix with respect to cluster labels and inspect visually.



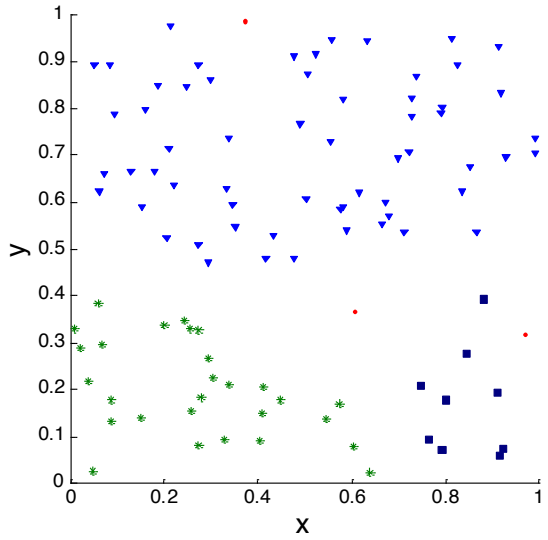


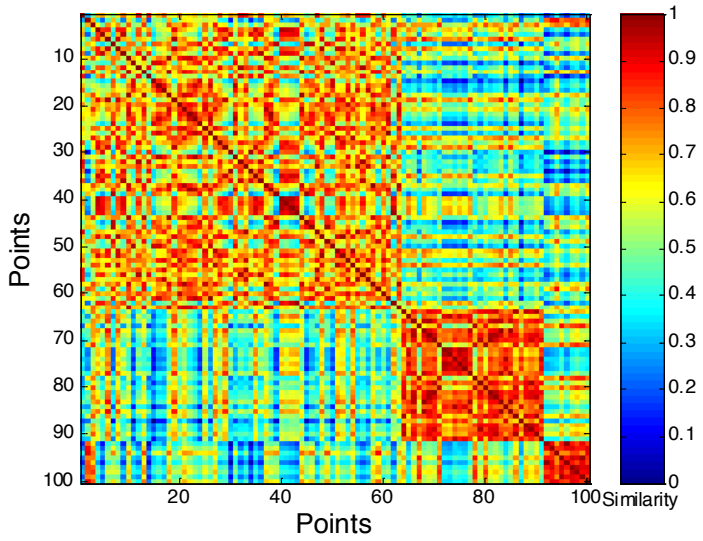
TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

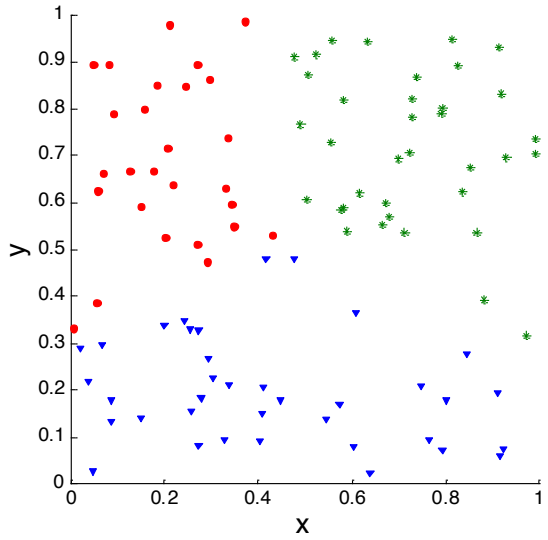


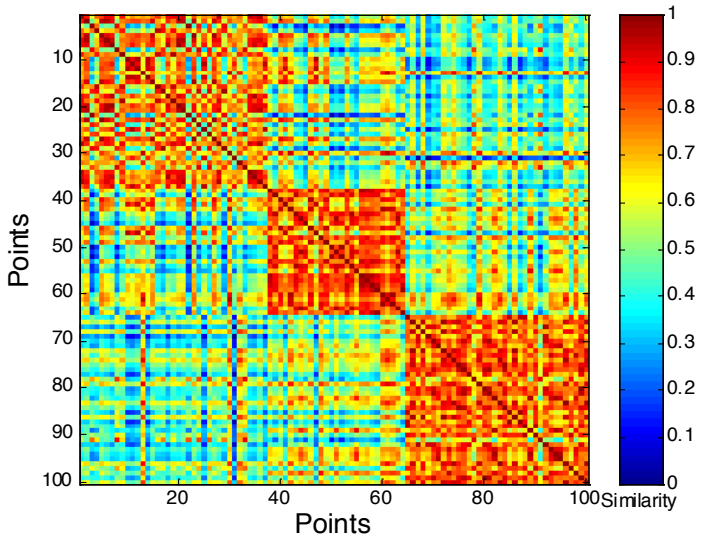


DBSCAN

Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

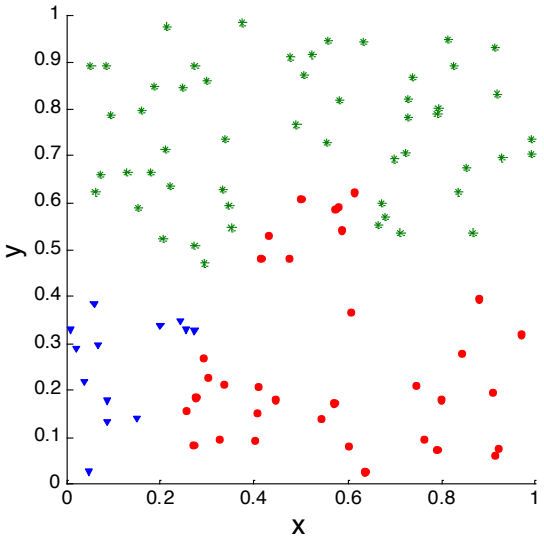


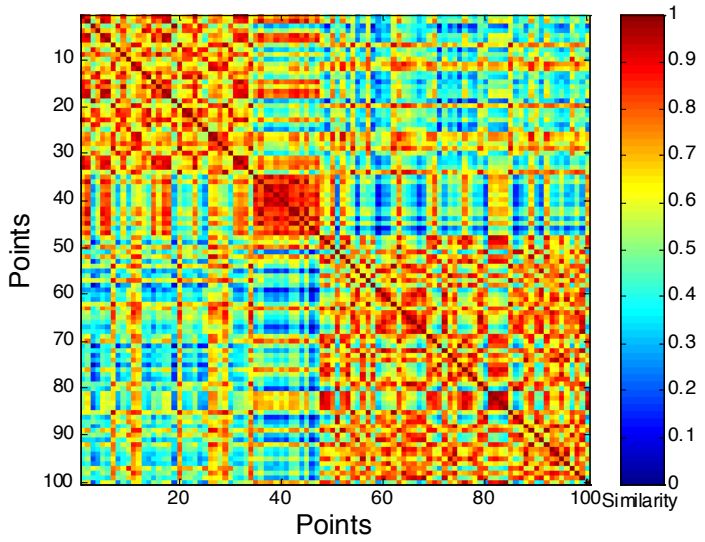


K-means

Using Similarity Matrix for Cluster Validation

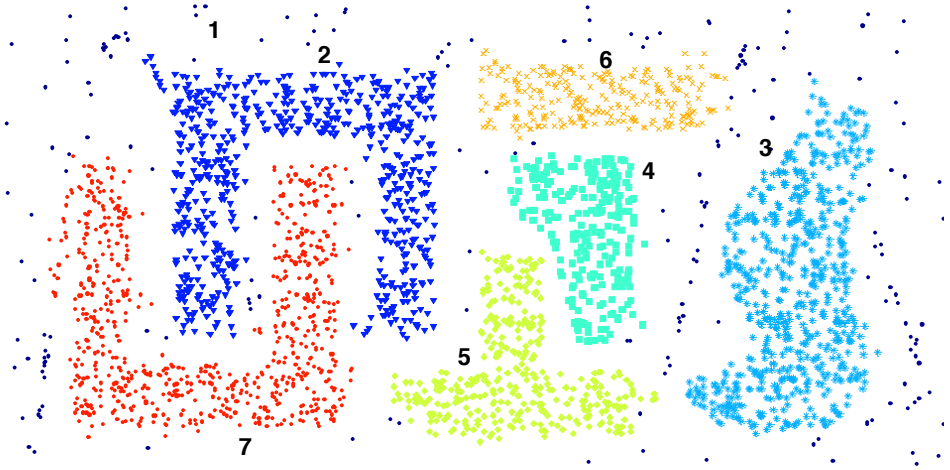
- Clusters in random data are not so crisp

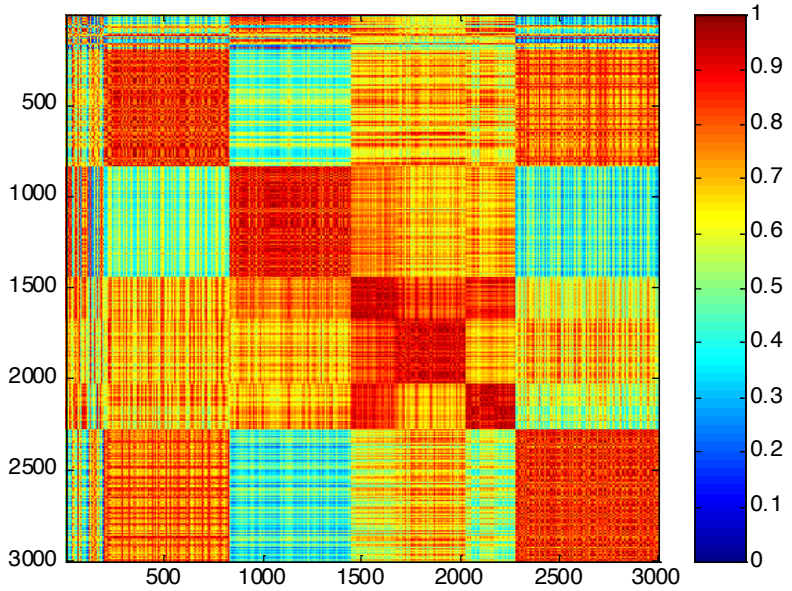




Complete Link

Using Similarity Matrix for Cluster Validation

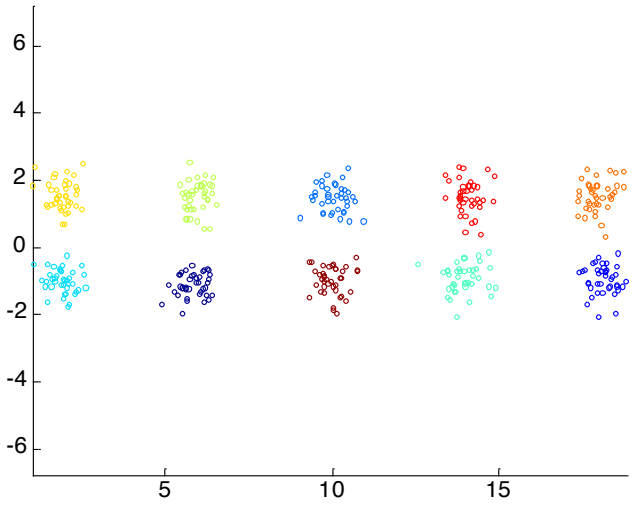


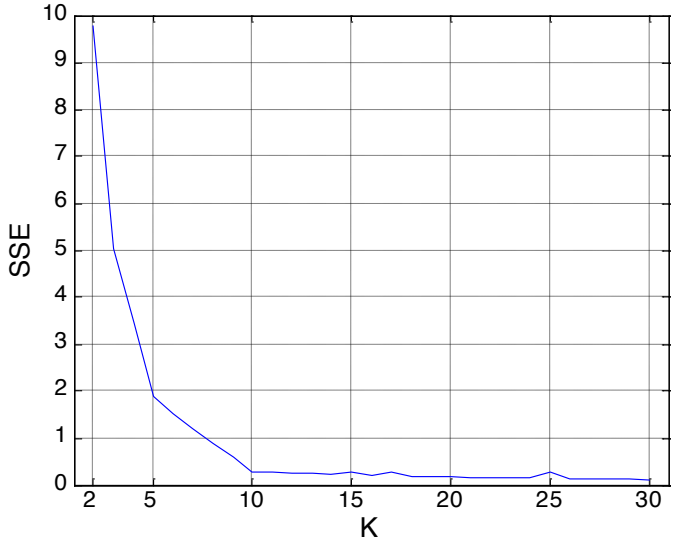


DBSCAN

Internal Measures: SSE

- Clusters in more complicated figures aren't well separated
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information
  - SSE
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters



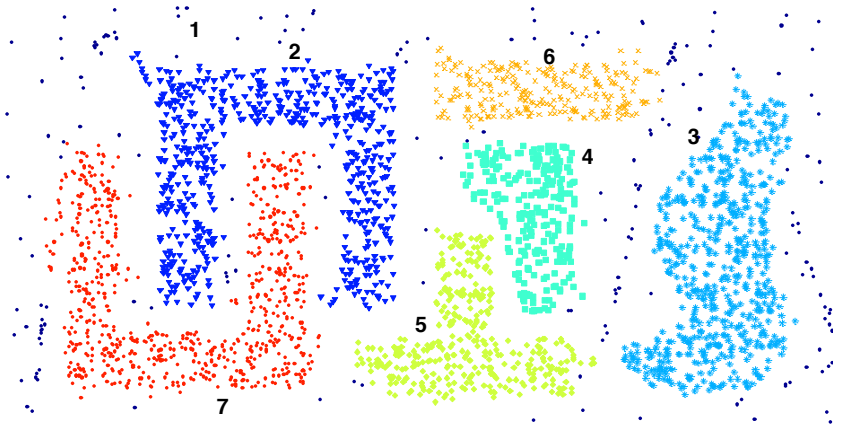


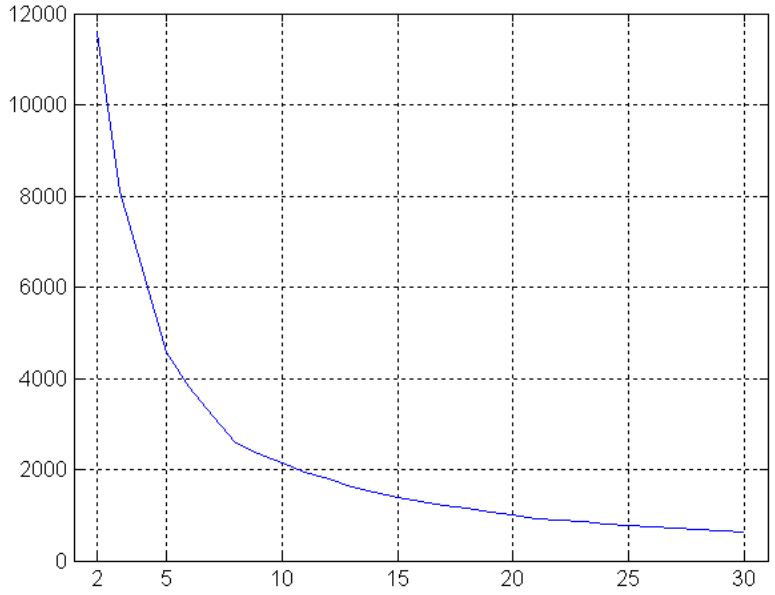
TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Internal Measures: SSE

- SSE curve for a more complicated data set





SSE of clusters found using K-means

Internal Measures: Cohesion and Separation

- Cluster Cohesion: Measures how closely related are objects in a cluster
  - Example: SSE
- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters
- Example: Squared Error
  - Cohesion is measured by the within cluster sum of squares (SSE)

$$SSE = WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

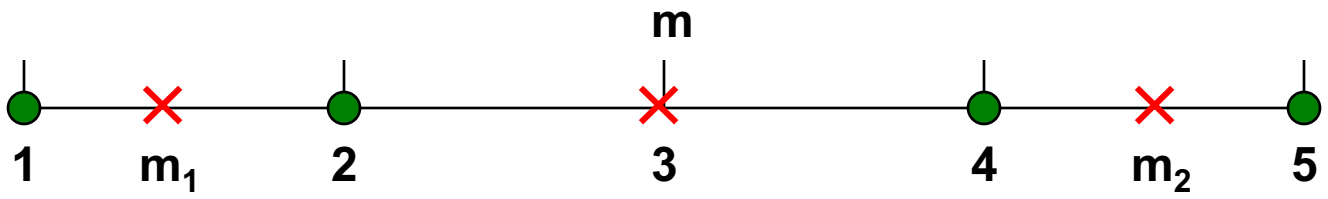
  - Separation is measured by the between cluster sum of squares

$$BSS = \sum_i |C_i| (m - m_i)^2$$

    - Where $|C_i|$ is the size of cluster i

Internal Measures: Cohesion and Separation

- Example: SSE
  - BSS + WSS = constant



K=1 cluster:

$$SSE = WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$BSS = 4 \times (3-3)^2 = 0$$

$$Total = 10 + 0 = 10$$

K=2 clusters:

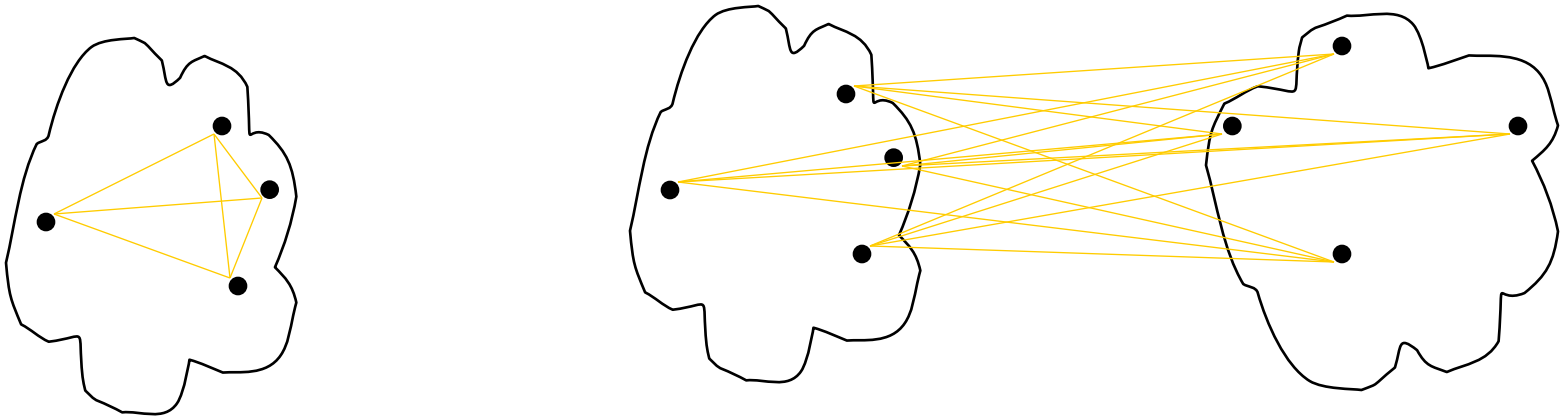
$$SSE = WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

$$Total = 1 + 9 = 10$$

Internal Measures: Cohesion and Separation

- A proximity graph based approach can also be used for cohesion and separation.
  - Cluster cohesion is the sum of the weight of all links within a cluster.
  - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion

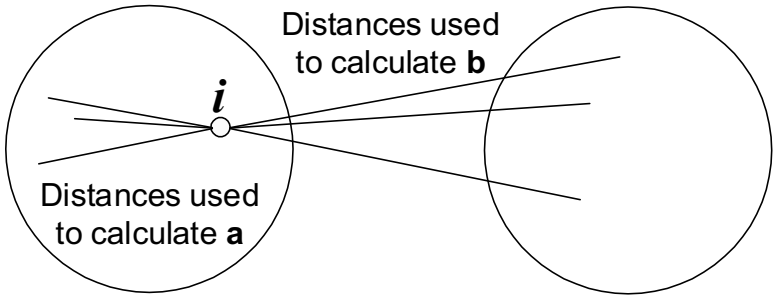
separation

Internal Measures: Silhouette Coefficient

- Silhouette coefficient combines ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point, i
  - Calculate a = average distance of i to the points in its cluster
  - Calculate b = min (average distance of i to points in another cluster)
  - The silhouette coefficient for a point is then given by

    s = (b – a) / max(a,b)

  - Typically between 0 and 1.
  - The closer to 1 the better.



- Can calculate the average silhouette coefficient for a cluster or a clustering

External Measures of Cluster Validity: Entropy and Purity

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

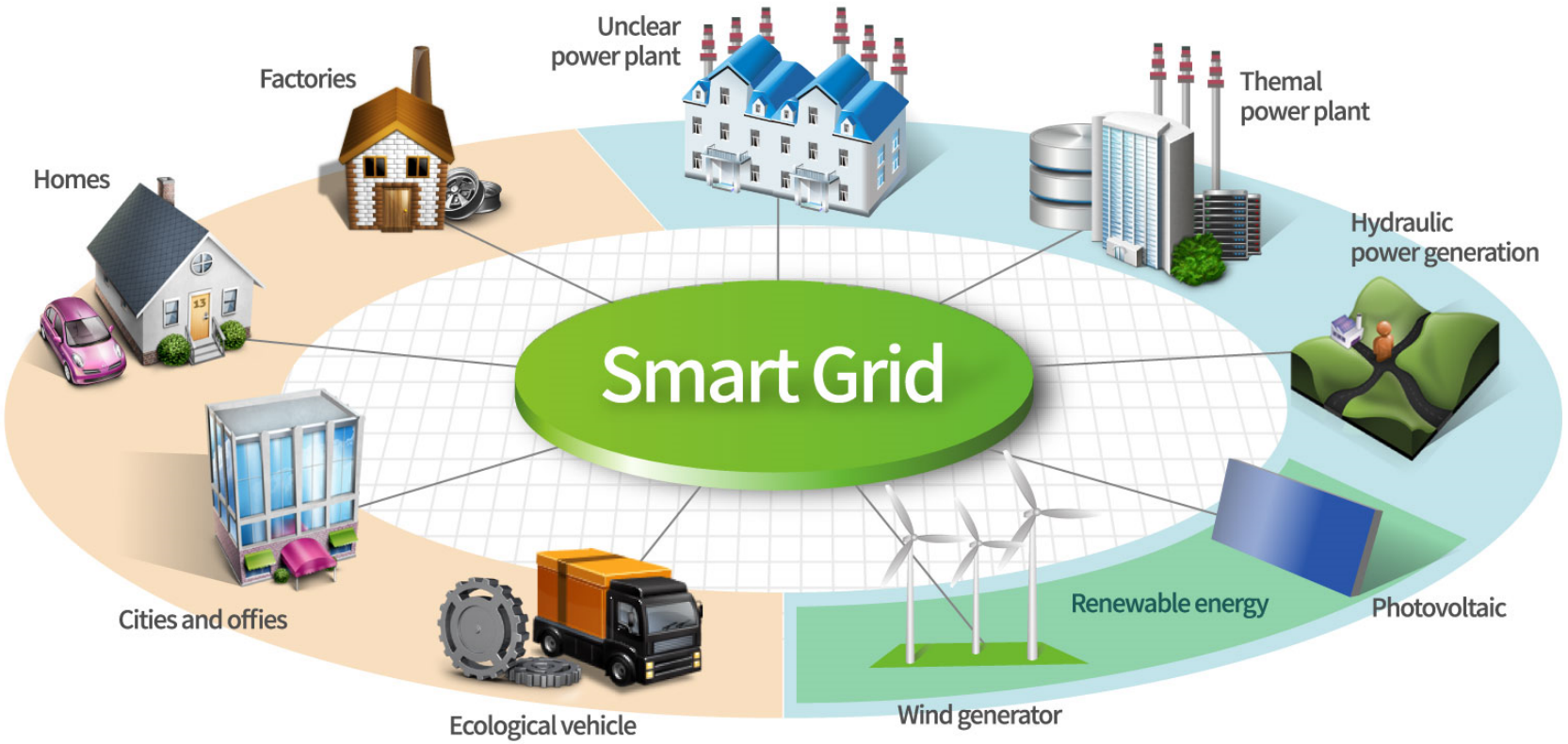
Case study

- Roy Ka Wei LEE, Tin Seong KAM: Time-Series Data Mining in Transportation: A Case Study on Singapore Public Train Commuter Travel Patterns, http://dx.doi.org/10.7763/ijet.2014.v6.737
  - Identification of commuter travel patterns
  - Clustering of passenger smart card readings
    - Time-series data
    - Hierarchical clustering
    - Pay attention to the used metric to compare two time-series – DTW
  - Better understanding of the passenger requirements allows transport engineers to design more appropriate public transport networks
  - Discovered correlations between the type of area (e.g. residential, industrial, commercial or retail) and commuter patterns
  - Prediction of passenger behavior after network exetnsions

Smart Grid Case study

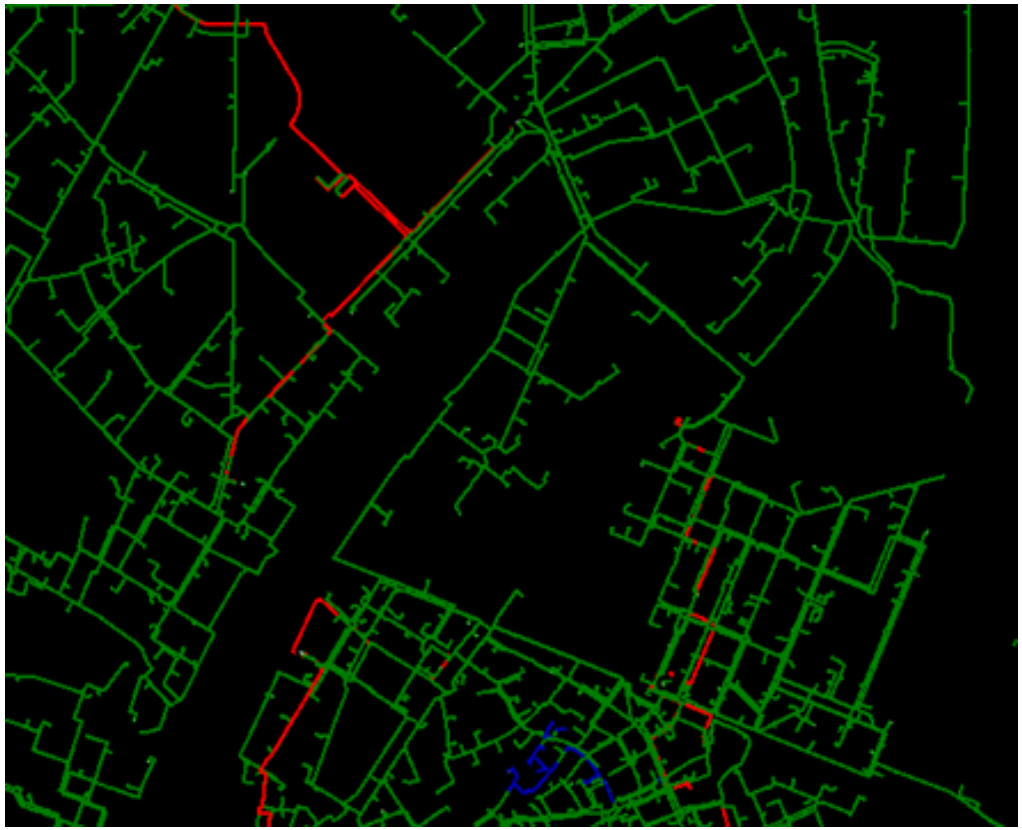
- Creation of power load type profiles
- Applicable to any measurement of flow, e.g. traffic flow

Smart Grid


Unclear
power plant
Factories
Themal
power plant
Homes
Hydraulic
power generation
Smart Grid
Cities and offies
Renewable energy
Photovoltaic
Ecological vehicle
Wind generator

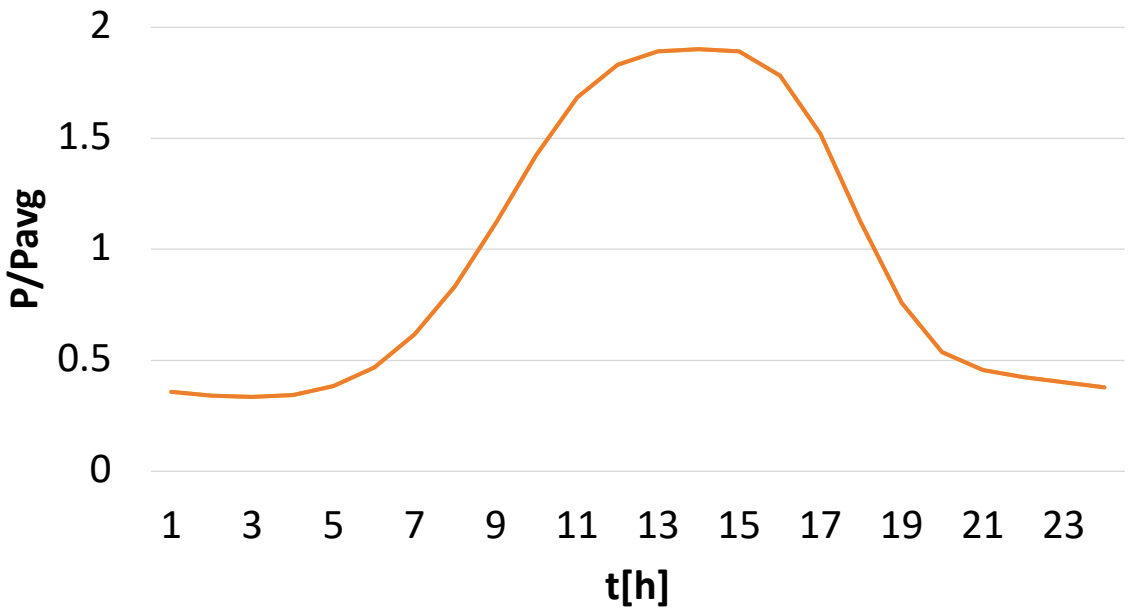
Power Distribution Management System

- Network monitoring
  - Load flow
- Network control
- Network planning



Load Model

- Model of an electrical consumer:
  - Annual average active power
  - Annual average reactive power
  - Load type
- Load type:
  - Set of normalized daily load profiles (DLP)
  - One per (season, day type)



TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

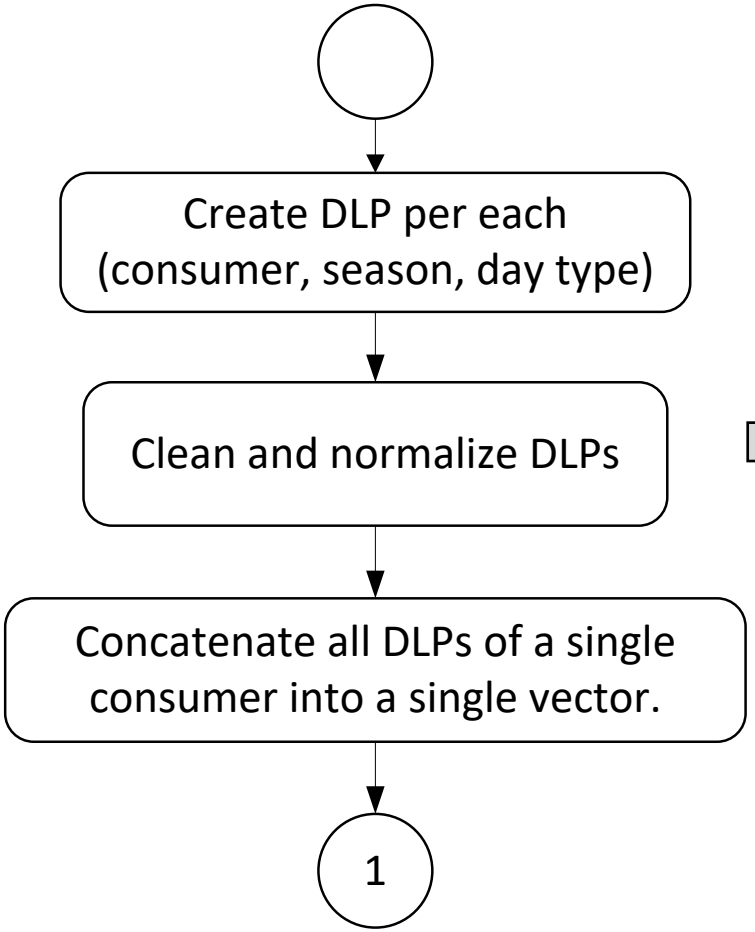
Load Type Creation Algorithm

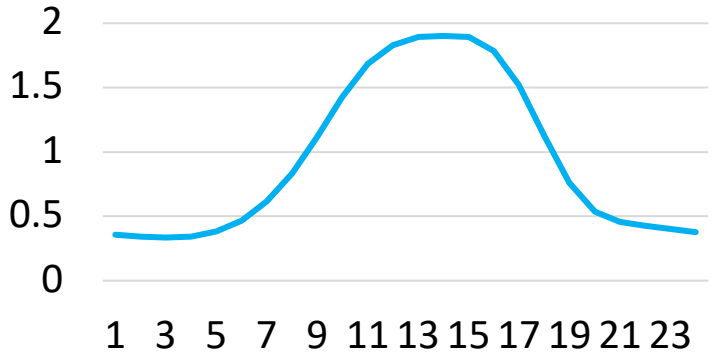
- Input
  - Measurements of P and Q for a year period – 15’ sampling rate
  - Seasons
  - Day types
  - Minimum and maximum number of load types
- Output
  - Set of load types (clusters)
- Implemented in pure C#
- 7 Million consumers

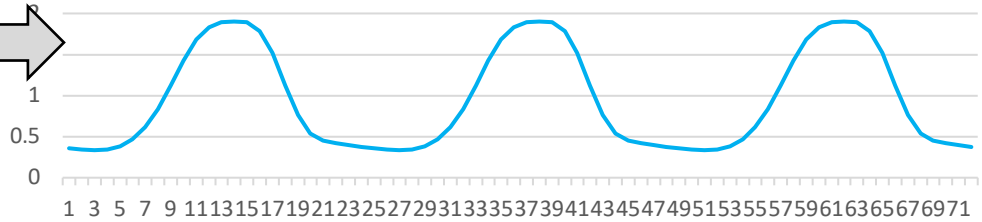
TRANSP-OR

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Load Type Creation Algorithm



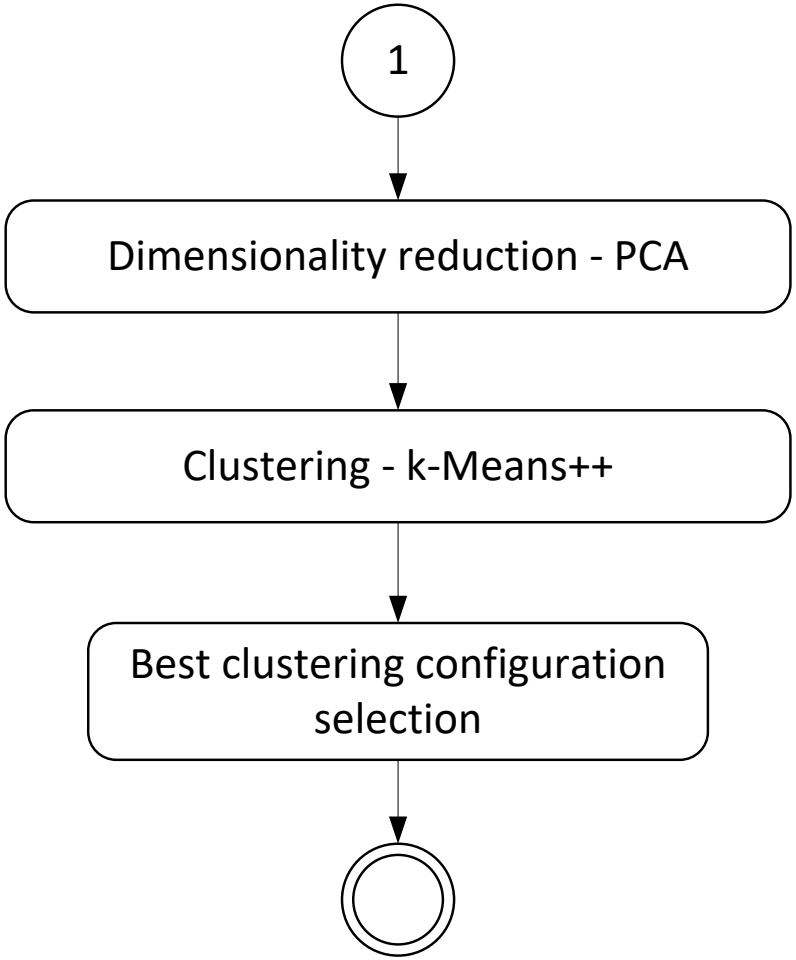




TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Load Type Creation Algorithm



Analyzed Metrics

- **Minkowski distance**

$$d_M(x, y) = (\sum_{k=1}^{n} |x_k - y_k|^r)^{\frac{1}{r}}$$

- **Manhattan distance:** $r = 1$
- **Euclidean distance:** $r = 2$

x, y - consumer's vectors

x_k, y_k - value at index k of consumer's vectors

n – number of dimensions (consumer's vector length)

r – metric parameter

Analyzed Metrics

- **Cosine distance**

$$d_S(x,y) = 1 - \cos(x,y) = 1 - \frac{xy}{\|x\|\|y\|}$$

- **Cross Correlation distance**

$$d_{CC}(x,y,\alpha) = 1 - \frac{\sum_{k=1}^{n}[(x_k - \bar{x})(y_{k-\alpha} - \bar{y})]}{\sqrt{\sum_{k=1}^{n}(x_k - \bar{x})^2}\sqrt{\sum_{k=1}^{n}(y_{k-\alpha} - \bar{y})^2}}$$

- α - delay
- $\bar{x}$ - mean value of x
- $\bar{y}$ - mean value of y

Analyzed Metrics

- **Spearman's rank correlation coefficient**
  - Ranks each value in the time series
  - Time series normalized to the unitless domain

$$d_{SR}(x,y) = 1 - \left(\frac{6\sum_{k=1}^{n}(rank(x_k) - rank(y_k))^2}{n(n^2-1)}\right)$$

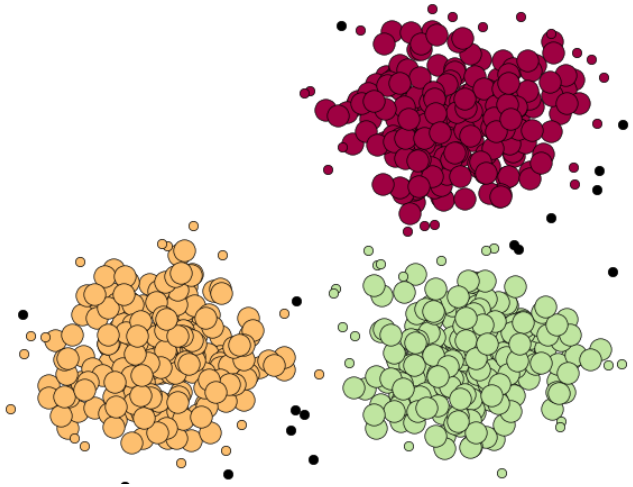
- **Curve Shape Distance**
  - Accounts for the curvature

$$d_{CS}(x,y) = d_E(x,y) + \sum_{k=1}^{n-1} |(x_{k+1} - x_k) - (y_{k+1} - y_k)|/\Delta t$$

Validity Indices

Validity assessment includes two measurement criteria:

- *Compactness:* The members of each cluster should be as close to each other as possible
  - A common measure: the variance
- *Separation:* The clusters should be widely separated
  - Distance between the closest member of the clusters
  - Distance between the most distant members
  - Distance between the centres of the clusters



Davies-Bouldin (DB) Validity Index

- DB index: the average of similarity between each cluster and its most similar one
- The lower DB index – the better cluster configuration

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

$$DB = \frac{1}{n_c} \sum_{i=1}^{n_c} R_i$$

$$d_{ij} = d\left(v_i, v_j\right), \; s_i = \frac{1}{\left\|c_i\right\|} \sum_{x \in c_i} d\left(x, v_i\right)$$

$$R_i = \max_{j=1 \ldots nc, i \neq j} \left(R_{ij}\right), \; i = 1 \ldots n_c$$

SD Validity Index

- SD validity index: the average scattering of clusters and inversed total separation of clusters
- The lower SD index – the better cluster configuration

$$Scatt = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{\left\| \sigma(v_i) \right\|}{\left\| \sigma(x) \right\|}$$

$$Dis = \frac{\max\limits_{i,j=1\ldots n_c} \left(\left\| v_j - v_i \right\| \right)}{\min\limits_{i,j=1\ldots n_c} \left(\left\| v_j - v_i \right\| \right)} \sum_{i=1}^{n_c} \left(\sum_{\substack{j=1, \\ i \neq j}}^{n_c} \left\| v_j - v_i \right\| \right)^{-1}$$

$$SD = \alpha \cdot Scatt + Dis$$

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Assessment Process

- Data set
  - European power network
  - 3 different locations: CR1, CR2, CR3
  - 2000 monitored consumers per location
- Input data
  - Metered active power
  - Metered reactive power
- Clustering with different metrics in each data set
  - From 2 to 20 clusters

Results with PCA

VALUES OF SD VALIDITY INDEX

	CR1	CR2	CR3
Euclidean (L2)	0.70	0.72	0.70
Cosine	1.00	1.02	0.80
Cross Correlation	1.00	1.18	0.96
Spearman	1.06	1.23	1.23
Curve Shape Dis.	0.84	0.70	0.63

VALUES OF DB VALIDITY INDEX

	CR1	CR2	CR3
Euclidean (L2)	1.52	1.59	1.25
Cosine	2.70	2.33	2.63
Cross Correlation	1.85	1.92	1.89
Spearman	4.35	4.76	4.76
Curve Shape Dis.	1.79	1.67	1.79

Results without PCA

VALUES OF SD VALIDITY INDEX

	CR1	CR2	CR3
Euclidean (L2)	0.79	0.72	0.69
Cosine	0.68	0.62	0.65
Cross Correlation	1.32	1.39	1.16
Spearman	1.25	1.25	1.19
Curve Shape Dis.	0.35	0.47	0.42

VALUES OF DB VALIDITY INDEX

	CR1	CR2	CR3
Euclidean (L2)	1.56	1.33	1.43
Cosine	1.59	1.61	1.59
Cross Correlation	2.22	2.17	2.17
Spearman	2.38	2.44	2.56
Curve Shape Dis.	1.28	1.47	1

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Conclusions

- Data transformed with PCA ⇒ Euclidean metric
- Original (untransformed) data ⇒ Curve Shape Distance
- Overall best performance: original data and Curve Shape Distance

Future research paths

- Test other clustering algorithms
- Development of a consumption-based metric
- A larger number of data sets from different countries

Main references

- P.-N. Tan, M. Steinbach, A. Karpatne, V. Kumar: Introduction to Data Mining, 2nd Edition, 2006, Pearson Education Inc.
- Obrenović N., Vidaković G., Luković I. (2017) The Choice of Metric for Clustering of Electrical Power Distribution Consumers. In: Haber P., Lampoltshammer T., Mayr M. (eds) Data Science – Analytics and Applications. Springer Vieweg, Wiesbaden