

Focus

Training, Validation and Test

Thursday, 11 May 2017

Matteo Borrotti

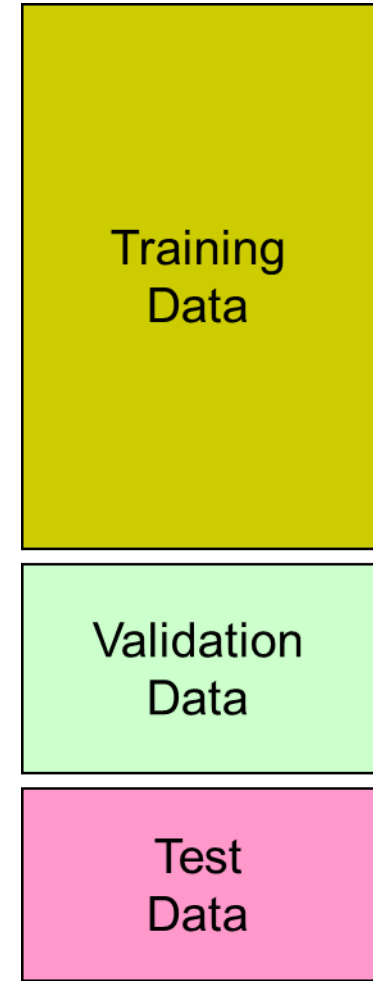
matteo.Borrotti@en-cre.it

Training, validation and test

Training set: a set of examples used for learning, where the target value is known.

Validation set: a set of examples used to tune the architecture of a classifier (or compare different classifiers) and estimate the error.

Test set: used only to assess the performances of a classifier. It is **never used** during the training process so that the error on the test set provides an unbiased estimate of the generalization error.



Training, validation and test

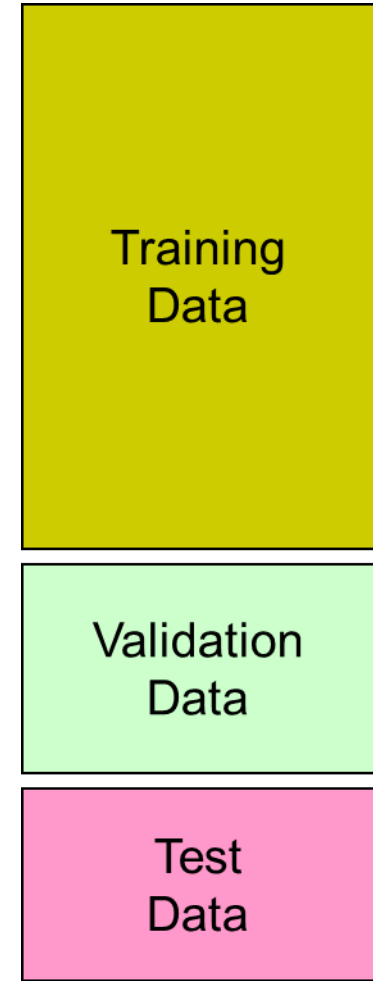
Method 1

Method 1 has one parameter to tune. The parameter has two possible levels.

Method 2

No parameter to set.

What is the best method?

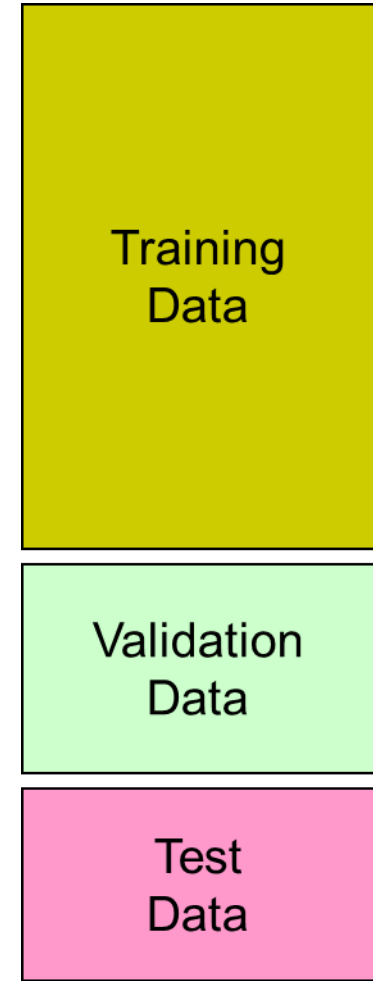


Training, validation and test

Train:

- Method 1 with parameter = level 1
- Method 1 with parameter = level 2
- Method 2

on the **training set**



Training, validation and test

Compare the performance of:

- Method 1 with parameter = level 1
- Method 1 with parameter = level 2
- Method 2

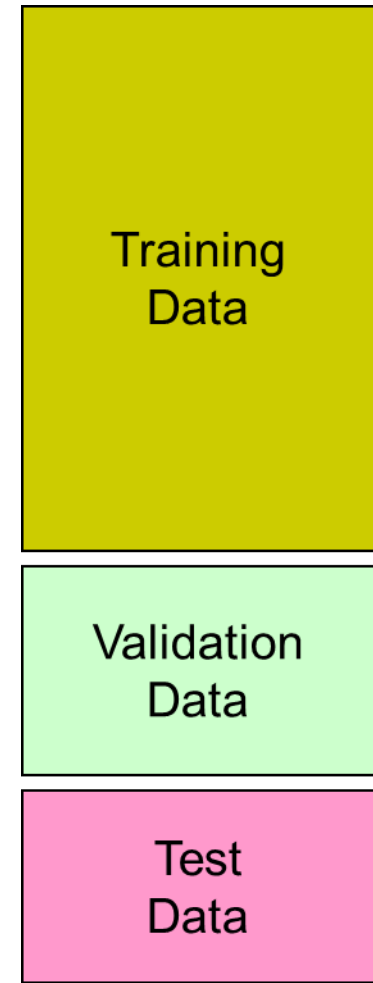
on the **validation set**.

Performance can be evaluated with different metrics such as:

- Accuracy
- Log Loss
- ROC curve
- ...

Select the best algorithm

i.e. Method 1 with parameter = level 2



Training, validation and test

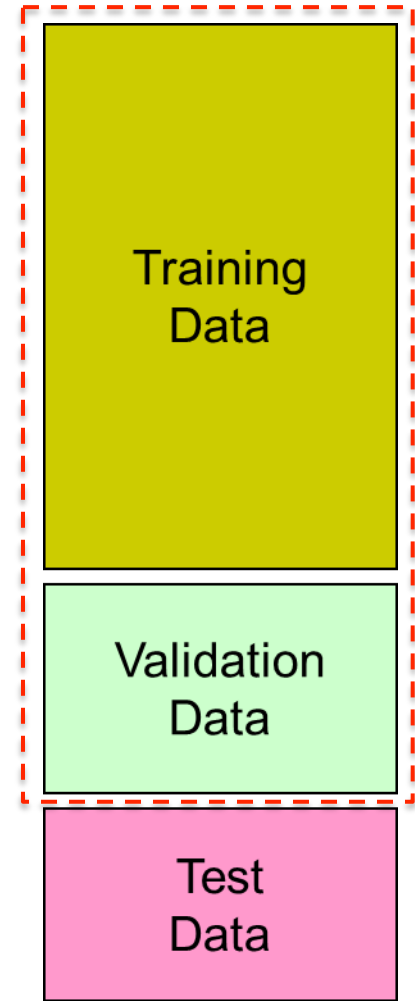
Evaluate the performance of:

- Method 1 with parameter = level 2
(selected algorithm)

on the **test set**.

Process:

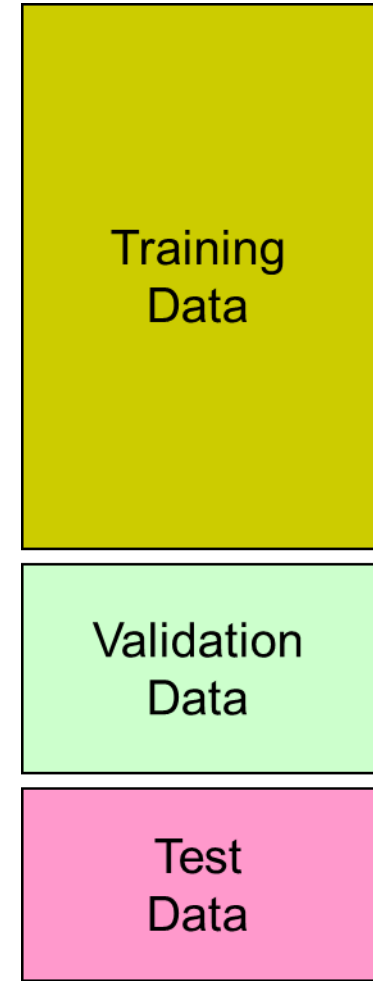
- Re-train the best classifier on Training Data + Validation Data
- Classify new instances
- Evaluate metrics



Training, validation and test

Possible sizes:

Training Set	Validation Set	Test Set
70%	15%	15%
50%	20%	30%
60%	20%	20%
70%	0%	30%



Pattern Recognition

Thursday, 11 May 2017

Matteo Borrotti

matteo.Borrotti@en-cre.it

Supervised and Unsupervised Approaches

Supervised approaches

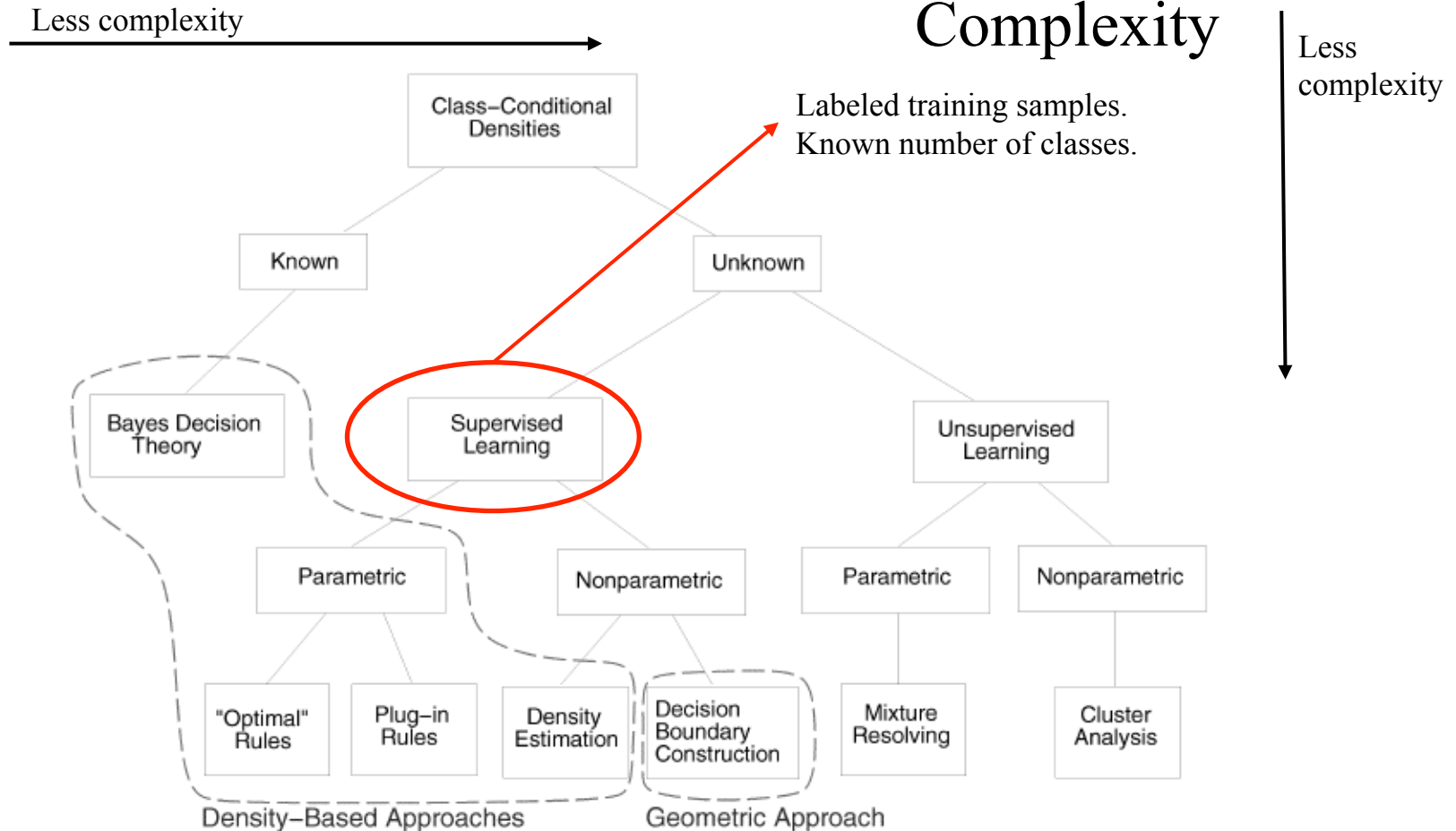
- Training data includes both the input and the desired results.
- For some examples the correct results (targets) are known and are given in input to the model during the learning process.
- The construction of a proper training, validation and test set is crucial.
- These methods are usually fast and accurate.
- Have to be able to **generalize**: give the correct results when new data are given in input without knowing a priori the target.

Supervised and Unsupervised Approaches

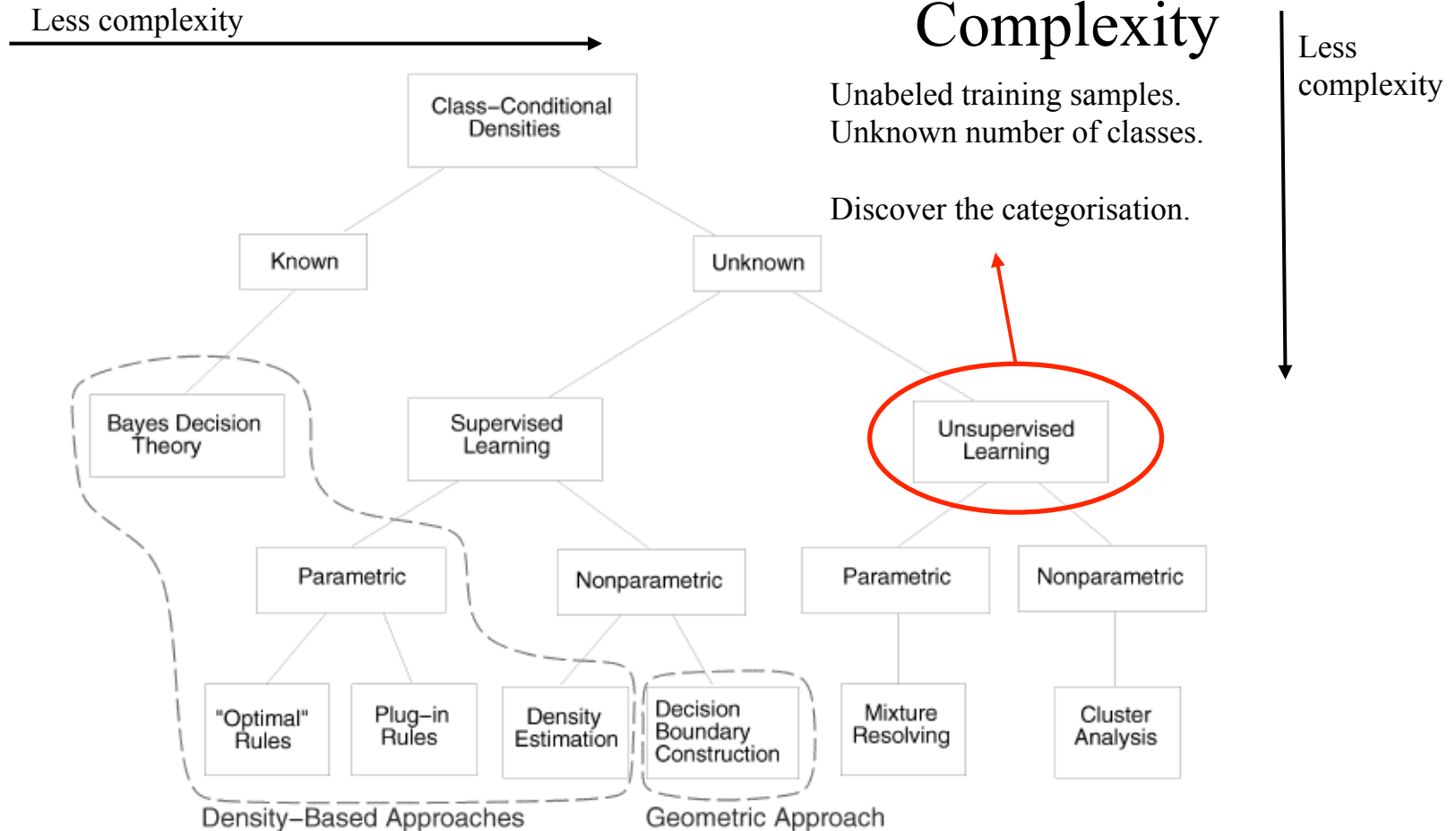
Unsupervised approaches

- The model is not provided with the correct results during the training.
- Can be used to cluster the input data in classes on the basis of their statistical properties only.
- Cluster significance and labeling.
- The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.

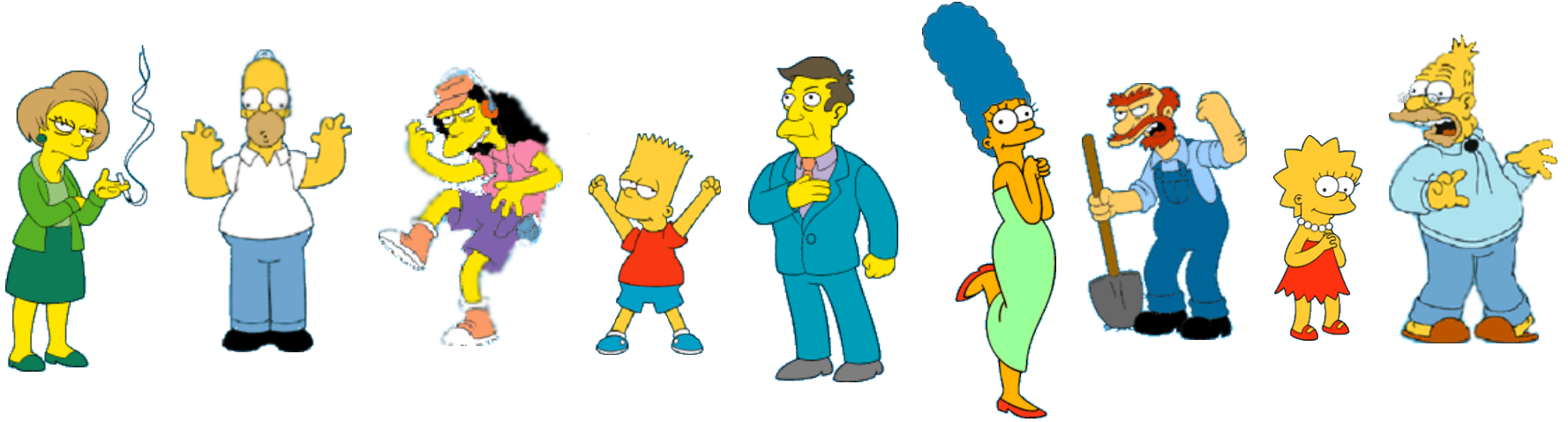
Supervised and Unsupervised Approaches



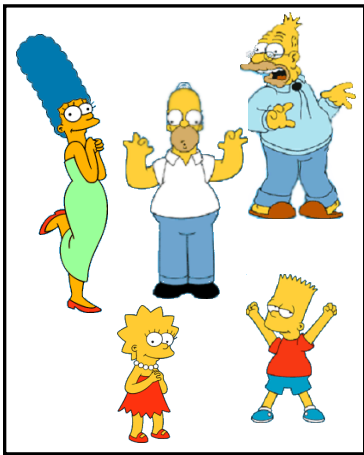
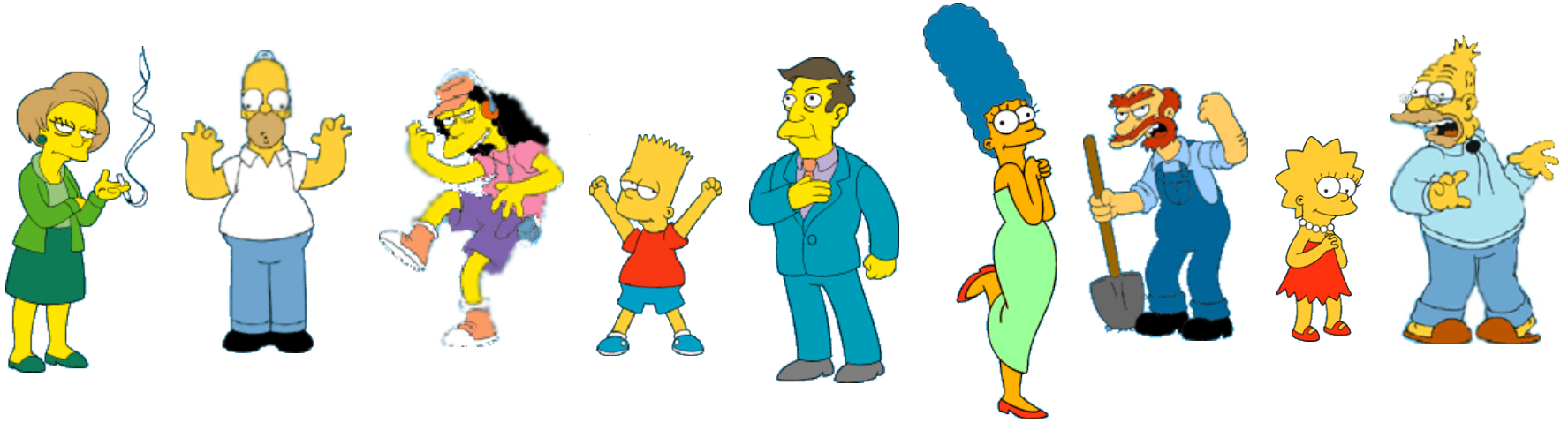
Supervised and Unsupervised Approaches



What is a natural grouping among these objects?



What is a natural grouping among these objects?

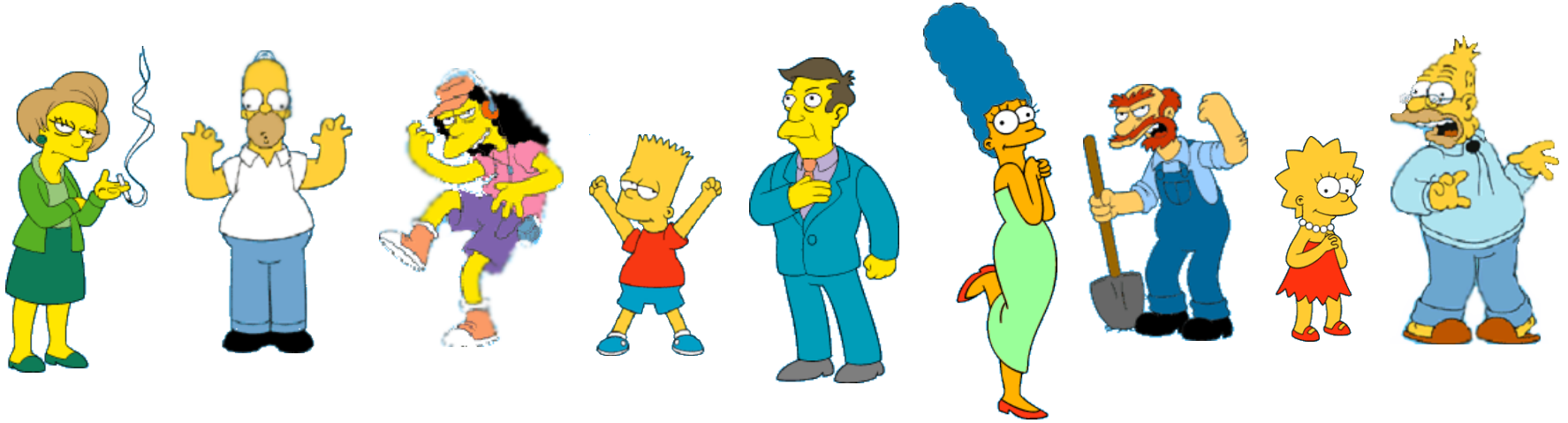


Simpson's Family

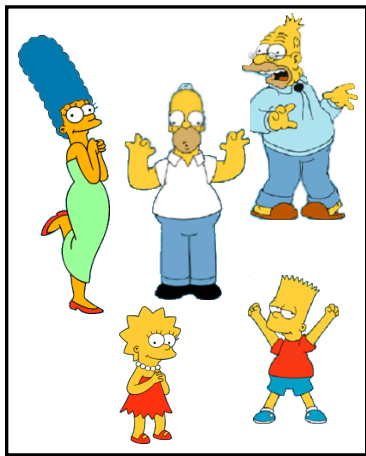


School Employees

What is a natural grouping among these objects?



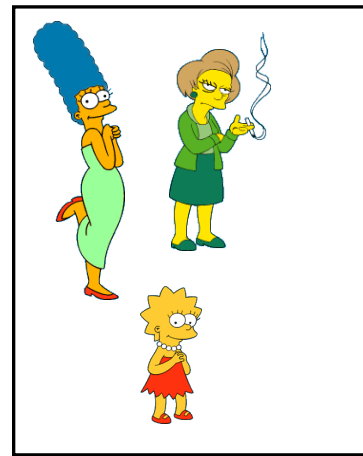
Clustering is subjective



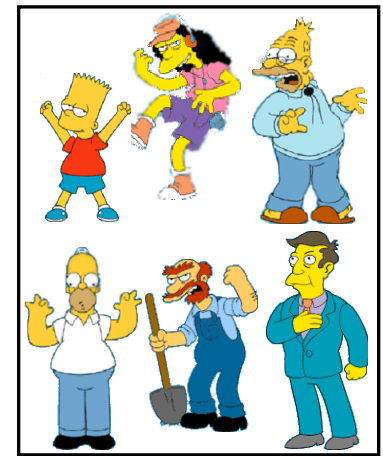
Simpson's Family



School Employees



Female



Male

Clustering

What is clustering?

- A way of grouping together data samples that are ***similar*** in some way - according to some criteria
- A form of ***unsupervised learning*** – you generally don't have examples demonstrating how the data *should* be grouped together
- So, it is a method of ***data exploration*** – a way of looking for patterns or structure in the data that are of interest

Clustering

What is clustering?

- Organizing data into classes such that there is
 - high intra-class similarity
 - low inter-class similarity
- Finding the class labels and the number of classes directly from the data (in contrast to classification).
- More informally, finding natural groupings among objects.

Clustering

Definition. *Clustering is a division of data into groups of similar objects. Each group (= a cluster) consists of objects that are similar between themselves and dissimilar to objects of other groups.*

What can be clustered?

- Images (astronomical data)
- Patterns (e.g. robot vision data)
- Shopping items
- Words
- Documents

Clustering

Definition. *Clustering is a division of data into groups of similar objects. Each group (= a cluster) consists of objects that are similar between themselves and dissimilar to objects of other groups.*

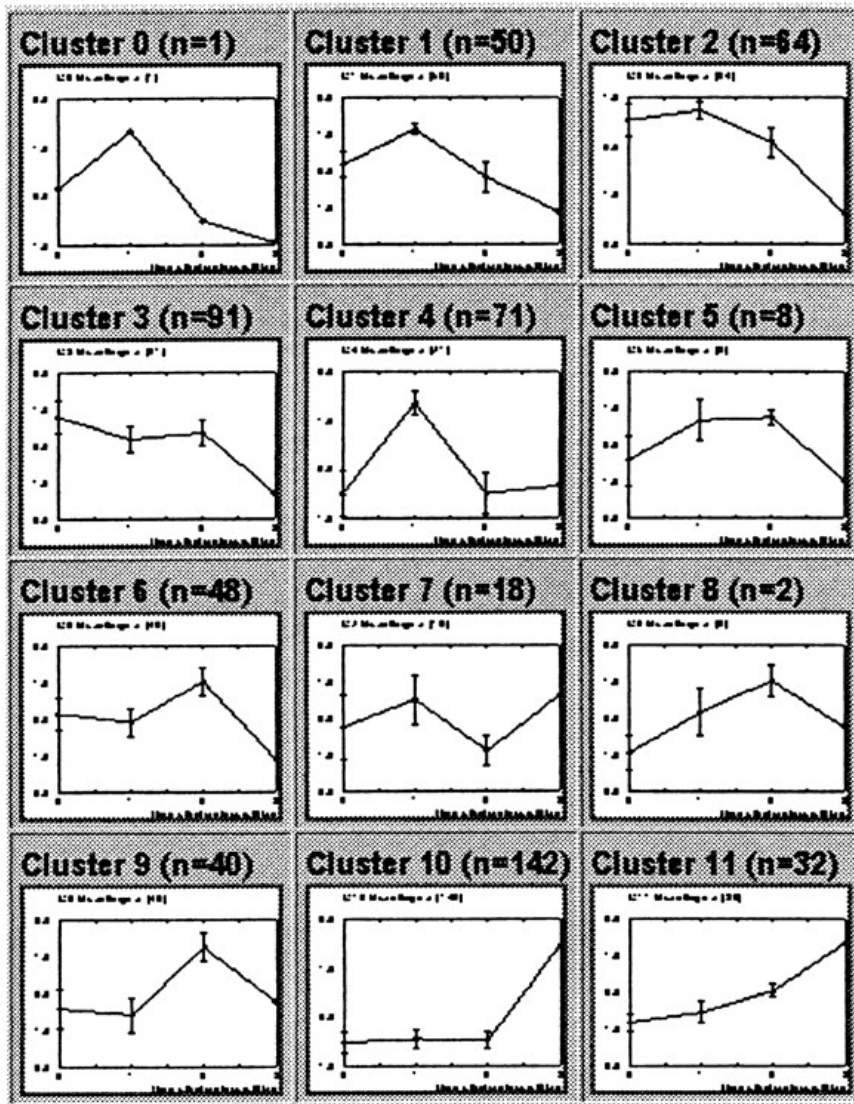
Applications:

- Data Mining (DNA-Analysis, Marketing studies, Insurance studies, ...)
- Text Mining (Text type clustering)
- Information Retrieval (Document clustering)
- Statistical Computational Linguistics (cluster-based n-gram models)
- Corpus-Based Computational Lexicography

Example: clustering genes

- P. Tamayo *et al.*, Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation, PNAS 96: 2907-12, 1999.
 - Treatment of HL-60 cells (myeloid leukemia cell line) with PMA leads to differentiation into macrophages
 - Measured expression of genes at 0, 0.5, 4 and 24 hours after PMA treatment

Example: clustering genes



- Self Organized Maps technique
- Cluster averages
- Clusters contain a number of known related genes involved in macrophage differentiation
- e.g., late induction cytokines, cell-cycle genes (down-regulated since PMA induces terminal differentiation), etc.

Clustering

Requirements that should be satisfied by clustering algorithms:

- Scalability
- Dealing with different types of attributes
- Discovering clusters of arbitrary shape
- Ability to deal with noise and outliers
- Insensitivity to the order of attributes
- Interpretability and usability

Problems encountered with clustering algorithms:

- Dealing with a large number of dimensions and a large number of objects can be prohibitive due to time complexity
- The effectiveness of an algorithm depends on the definition of similarity (distance)
- The outcomes of an algorithm can be interpreted in different ways

Clustering

Requirements that should be satisfied by clustering algorithms:

- Scalability
- Dealing with different types of attributes
- Discovering clusters of arbitrary shape
- Ability to deal with noise and outliers
- High dimensionality
- Insensitivity to the order of attributes
- Interpretability and usability

Problems encountered with clustering algorithms:

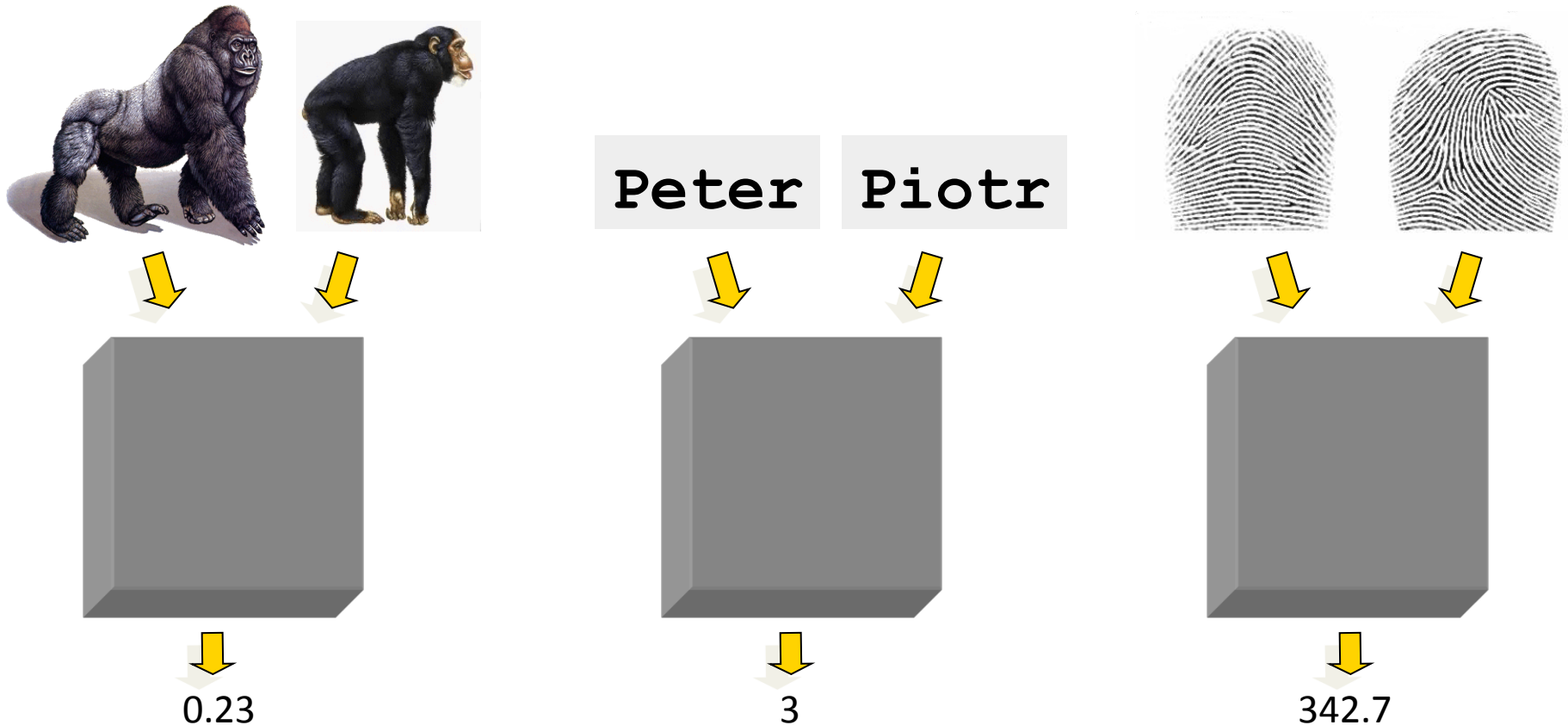
- Dealing with a large number of dimensions and a large number of objects can be prohibitive due to time complexity
- The effectiveness of an algorithm depends on the definition of **similarity** (distance)
- The outcomes of an algorithm can be interpreted in different ways

How do we define similarity?

- **Goal:** to group together “similar” data – but what does this mean?
- No single answer – it depends on what we want to find or emphasize in the data; this is one reason why clustering is an “art”
- The similarity measure is often more important than the clustering algorithm used – do not overlook this choice!

Distance Measures

Definition: Let O_1 and O_2 be two objects from the universe of possible objects. The distance (similarity or dissimilarity) between O_1 and O_2 is a real number denoted by $D(O_1, O_2)$



Properties of distance measures

For all points x, y, z a distance measure $D(\cdot, \cdot)$ must satisfy the following properties:

- Non negativity: $D(x, y) \geq 0$
- Reflexivity: $D(x, y) = 0$ if and only if $x = y$
- Symmetry: $D(x, y) = D(y, x)$
- Triangle inequality: $D(x, y) + D(y, z) \geq D(x, z)$

If the second property is not satisfied $D(\cdot, \cdot)$ is called pseudometrics.

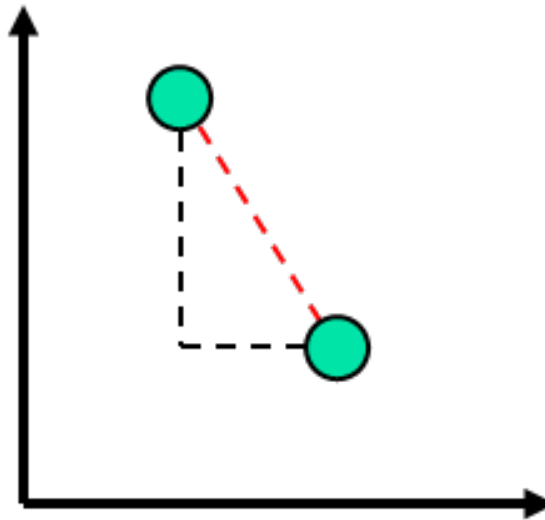
(Dis)similarity measures

- Instead of talking about similarity measures, we often equivalently refer to dissimilarity measures
- Jagota defines a dissimilarity measure as a function $f(\mathbf{x}, \mathbf{y})$ such that $f(\mathbf{x}, \mathbf{y}) > f(\mathbf{w}, \mathbf{z})$ if and only if \mathbf{x} is less similar to \mathbf{y} than \mathbf{w} is to \mathbf{z}
- This is always a *pair-wise* measure

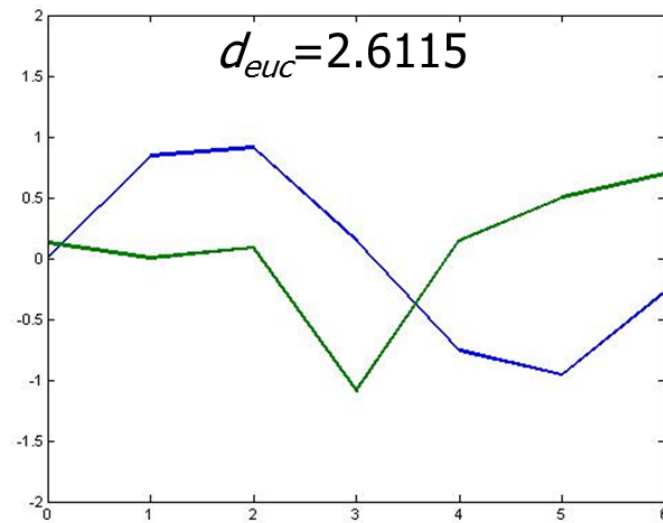
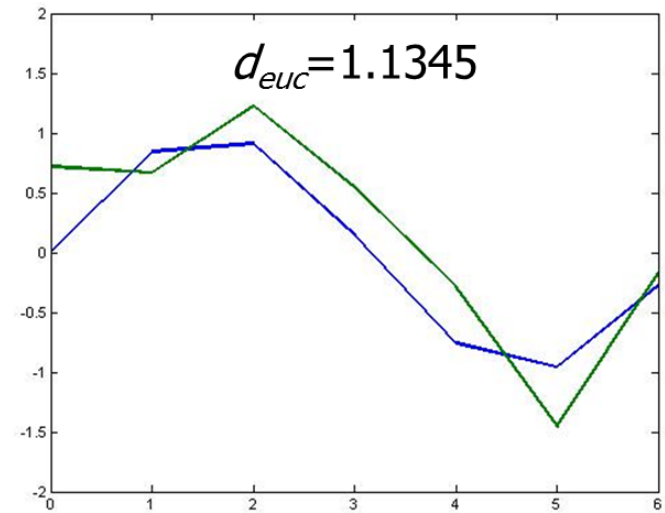
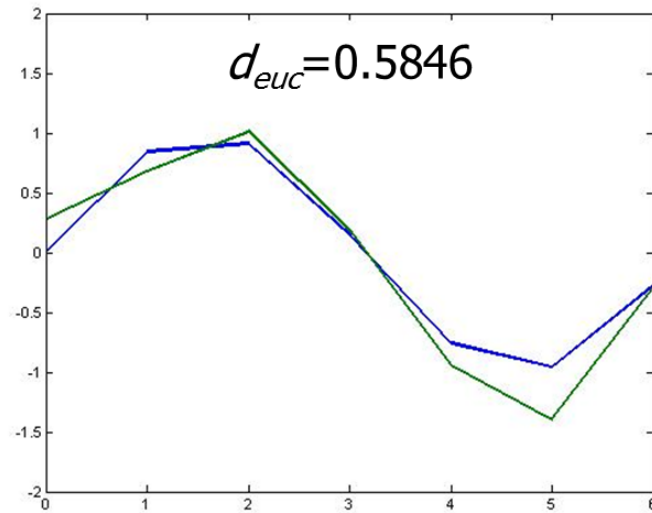
Euclidean distance

$$d_{euc}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- Here n is the number of dimensions in the data vector.



Euclidean distance



Person Linear Correlation

- Pearson linear correlation (PLC) is a measure that is invariant to scaling and shifting (vertically) of the expression values

$$\rho(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

$$\bar{x} = \frac{1}{n} \sum_i^n x_i$$

$$\bar{y} = \frac{1}{n} \sum_i^n y_i$$

- PLC measures the degree of a *linear* relationship between two features vectors!

Person Linear Correlation

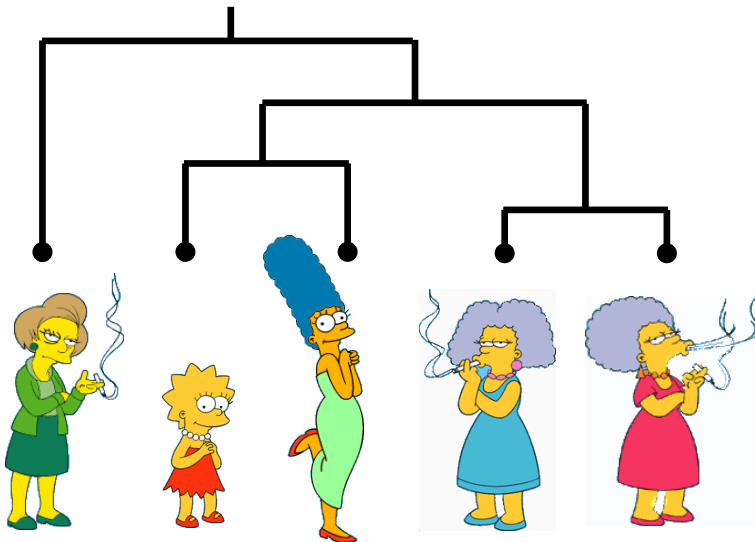
- Always between -1 and $+1$ (perfectly anti-correlated and perfectly correlated)
- This is a similarity measure, but we can easily make it into a dissimilarity measure:

$$d_p = \frac{1 - \rho(\mathbf{x}, \mathbf{y})}{2}$$

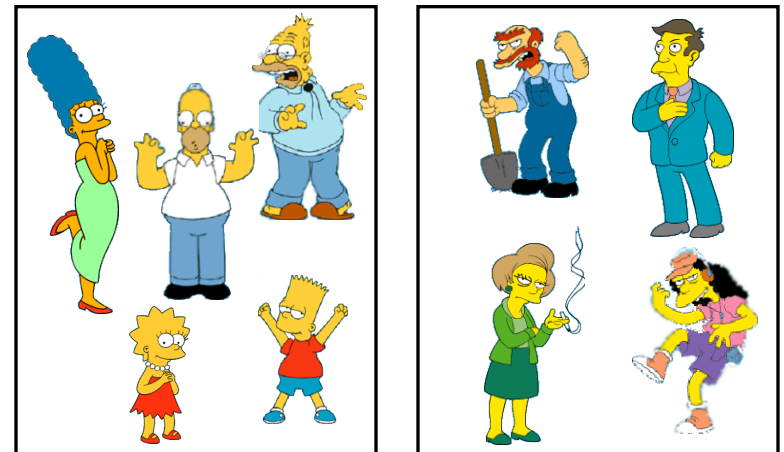
Types of Clustering

- **Hierarchical algorithms:** Create a hierarchical decomposition of the set of objects using some criterion
- **Partitional algorithms:** Construct various partitions and then evaluate them by some criterion

Hierarchical

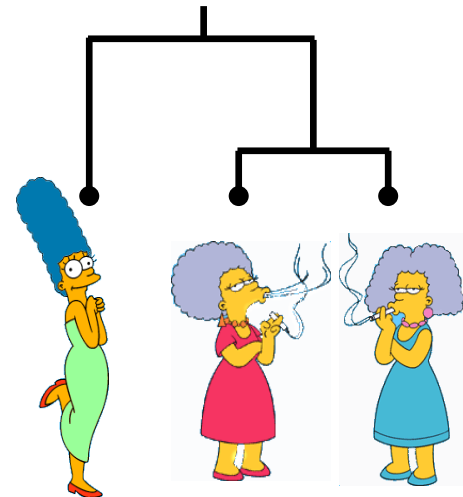
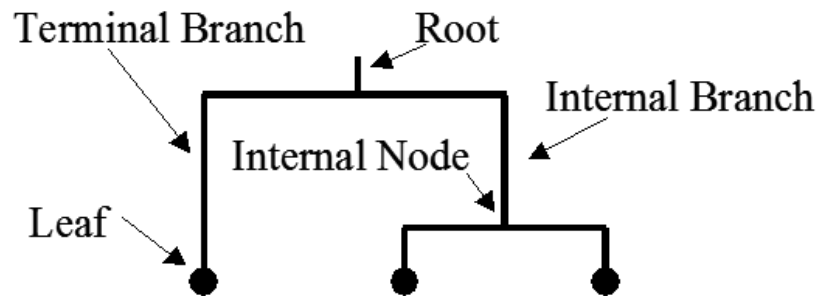


Partitional



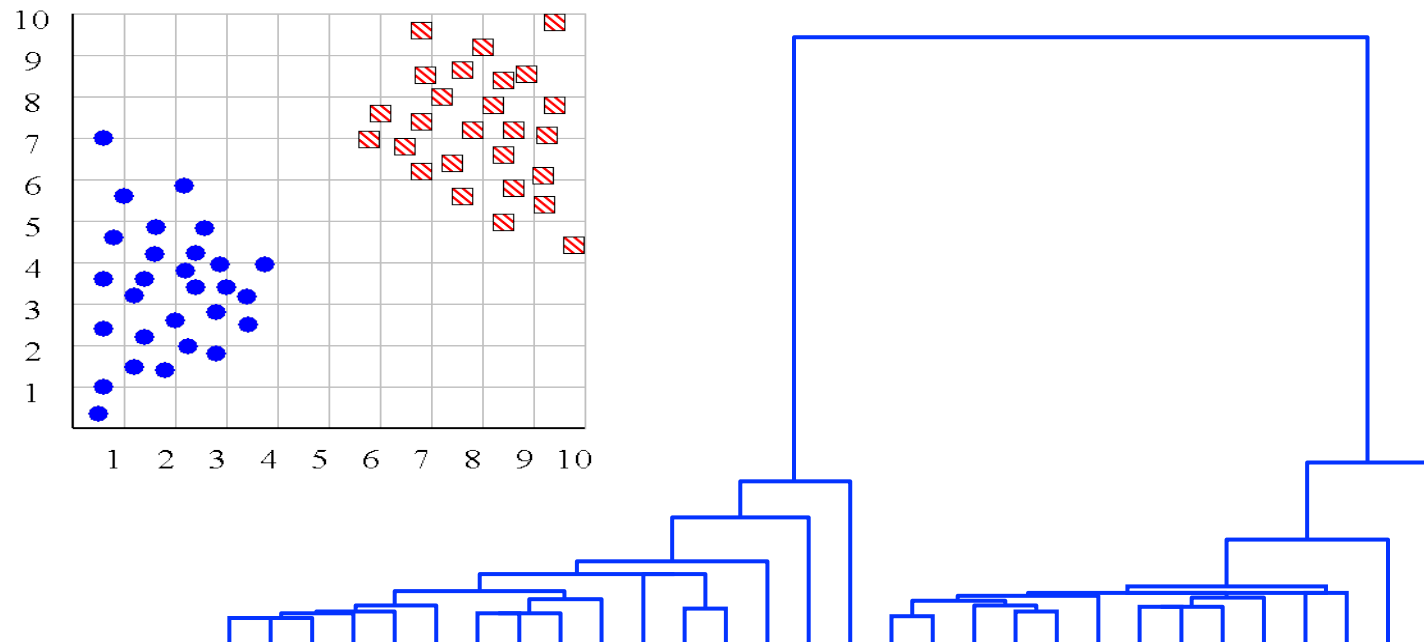
Dendrogram

- In order to better appreciate and evaluate hierarchical approaches, we will now introduce the *dendrogram*.
- The similarity between two objects in a dendrogram is represented as the height of the lowest internal node they share.

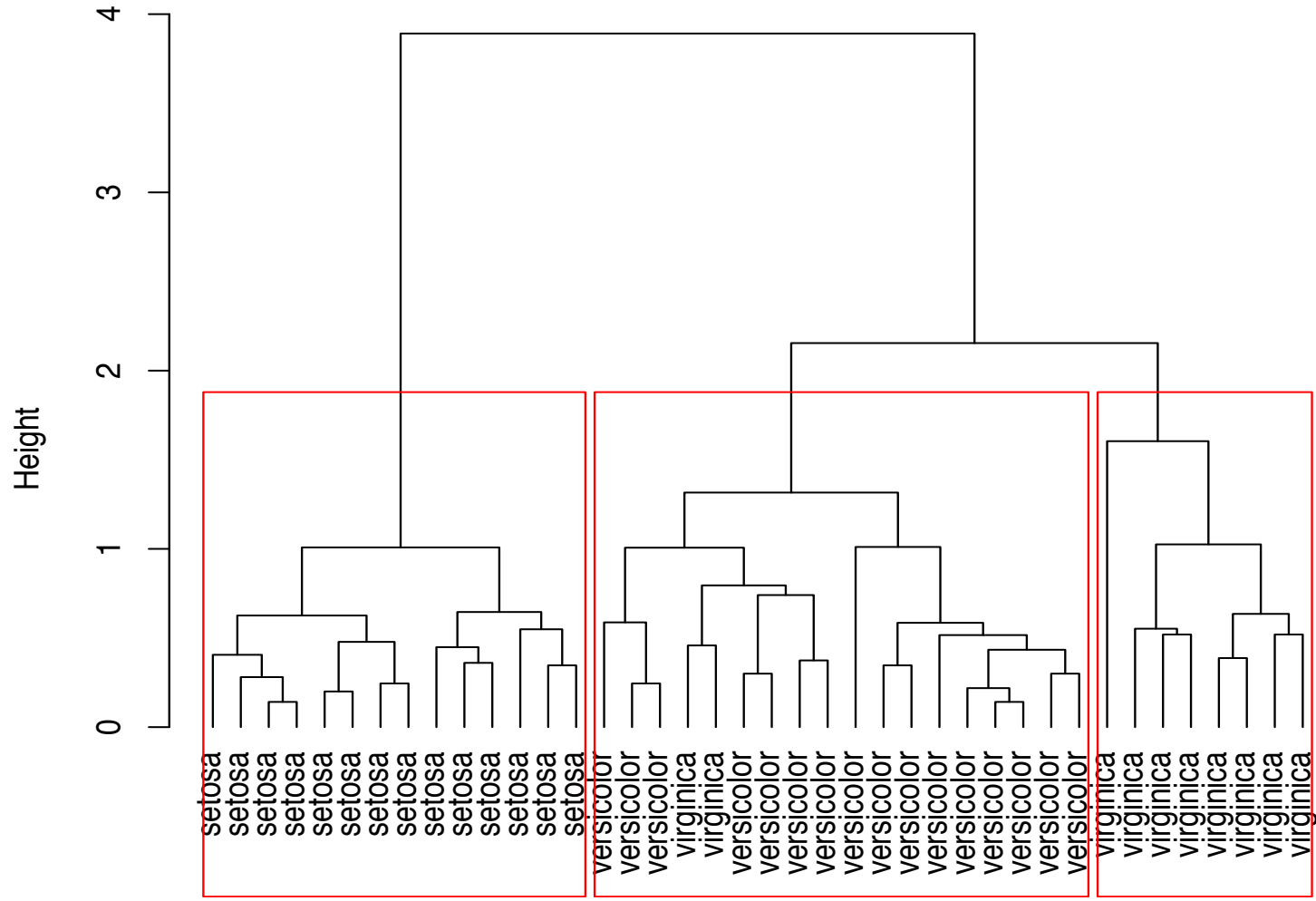


Dendrogram

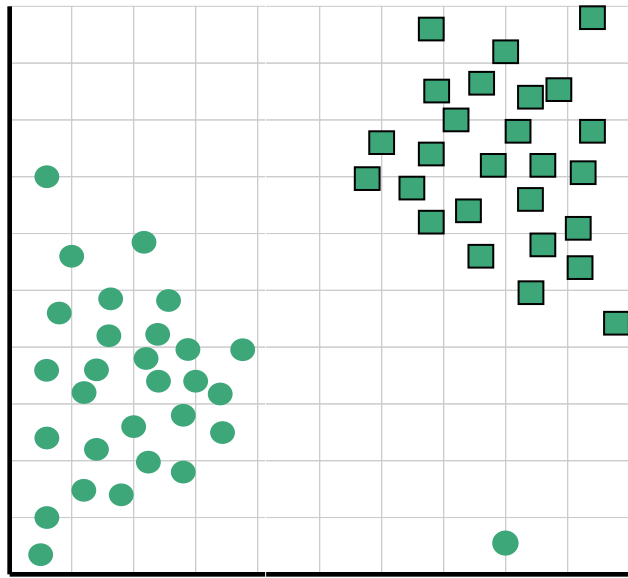
- We can look at the dendrogram to determine the “correct” number of clusters. In this case, the two highly separated subtrees are representative of two clusters.



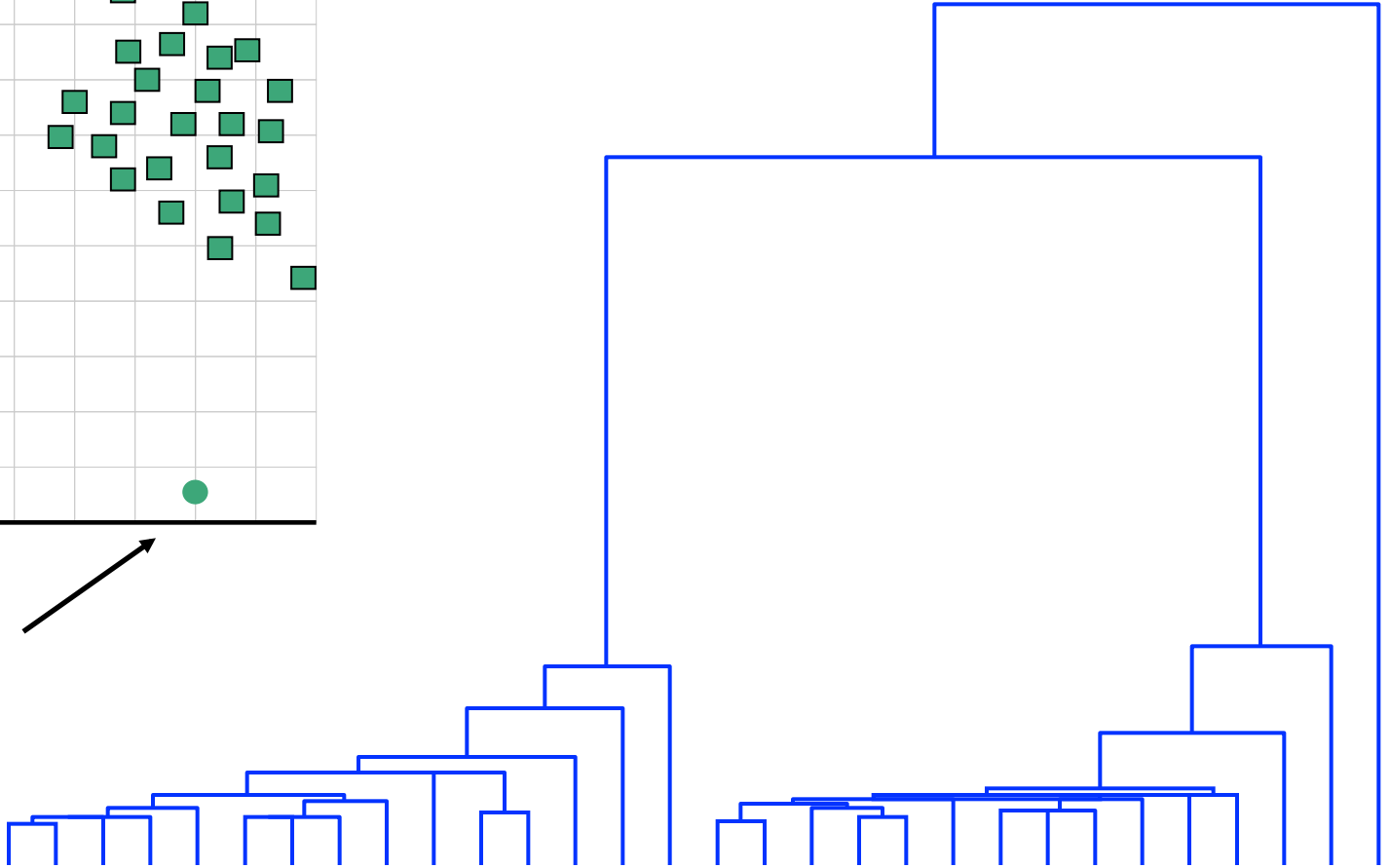
Dendrogram



Dendrogram



Outlier



Hierarchical Clustering

- Hierarchical Clustering builds a cluster hierarchy (a tree of clusters)

Bottom-Up (agglomerative):

- Starting with each item in its own cluster, find the best pair to merge into a new cluster.
- Repeat until all clusters are fused together.

Top-Down (divisive):

- Starting with all the data in a single cluster, consider every possible way to divide the cluster into two.
- Choose the best division and recursively operate on both sides.

Hierarchical Clustering


The number of dendrograms with n leafs


$$(2n - 3)! / [(2^{n-2}) (n - 2)!]$$

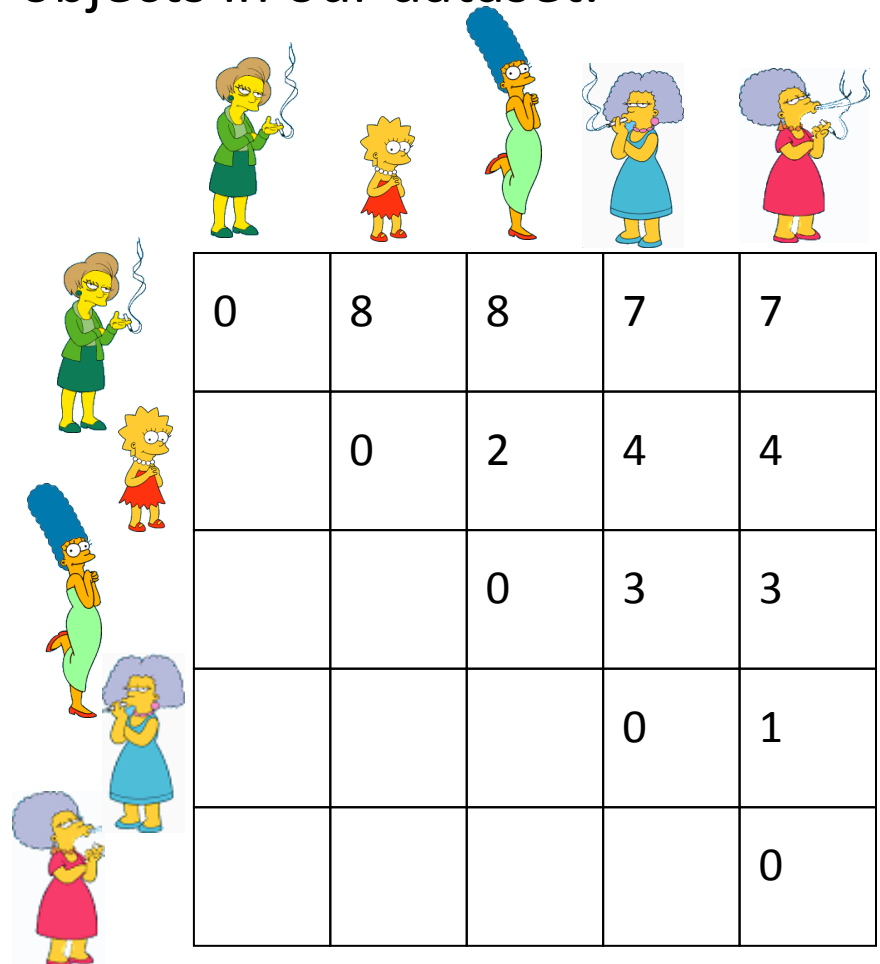
Number of Leafs	Number of Possible Dendrograms
2	1
3	3
4	15
5	105
...	...
10	34.459.425

Hierarchical Clustering

We begin with a distance matrix which contains the distances between every pair of objects in our dataset.


$$D(\text{Mrs. Muntz}, \text{Lisa Simpson}) = 8$$


$$D(\text{Marge Simpson}, \text{Mrs. Krabappel}) = 1$$

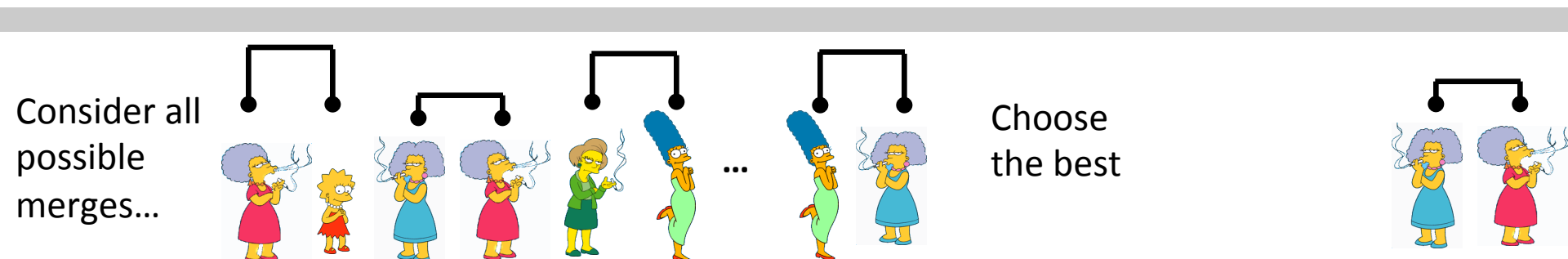


A diagram illustrating the distance matrix for the Simpson family members. The characters are arranged in a column to the left of the matrix, corresponding to the rows. From top to bottom: Mrs. Muntz, Lisa Simpson, Marge Simpson, Mrs. Krabappel, and Mrs. Gribble. The matrix is a 5x5 grid where the diagonal elements are 0, and the off-diagonal elements represent the distances between the characters. The distances are: Mrs. Muntz to Lisa (8), Marge (8), Mrs. Krabappel (7), and Mrs. Gribble (7); Lisa to Marge (2), Mrs. Krabappel (4), and Mrs. Gribble (4); Marge to Mrs. Krabappel (3) and Mrs. Gribble (3); Mrs. Krabappel to Mrs. Gribble (1); and Mrs. Gribble to herself (0).

0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

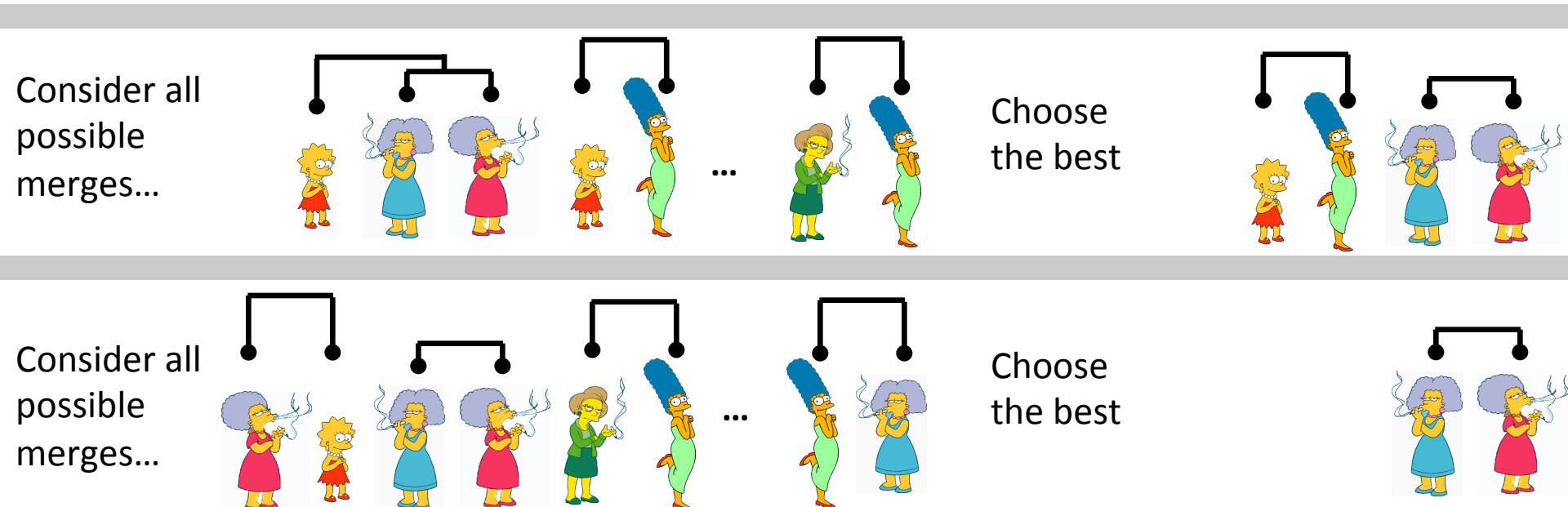
Bottom-Up (Agglomerative)

Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



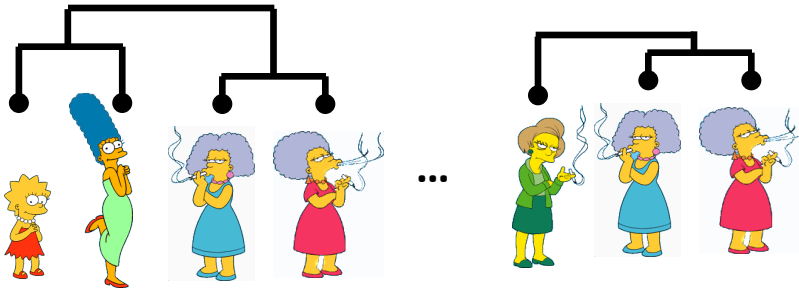
Bottom-Up (Agglomerative)

Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

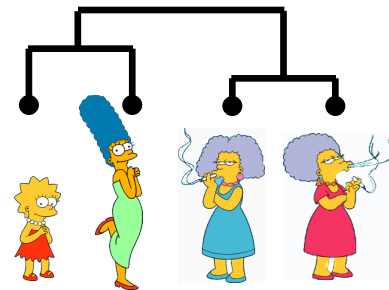


Bottom-Up (Agglomerative)

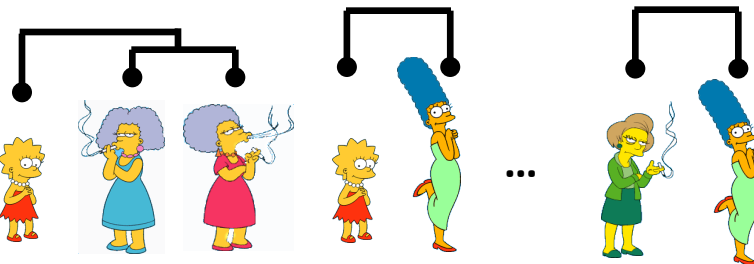
Consider all possible merges...



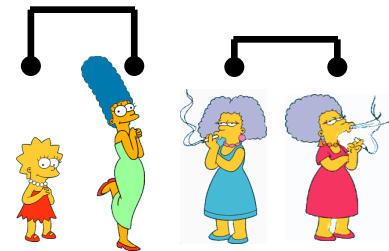
Choose the best



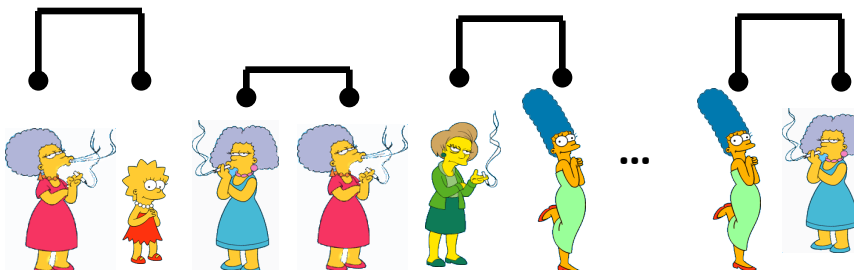
Consider all possible merges...



Choose the best



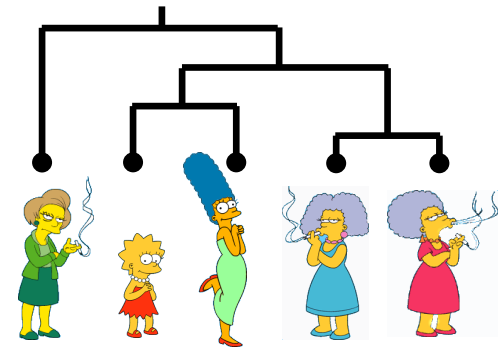
Consider all possible merges...



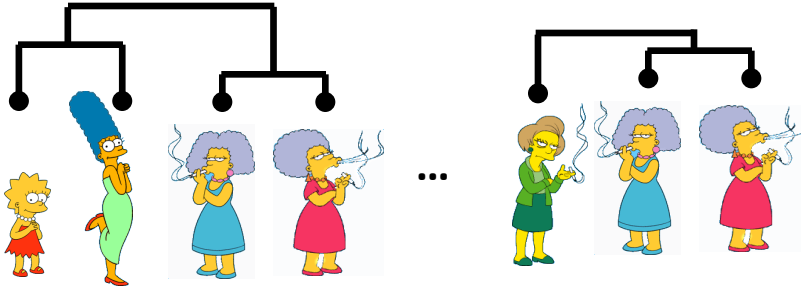
Choose the best



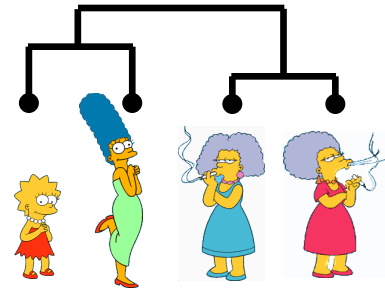
Bottom-Up (Agglomerative)



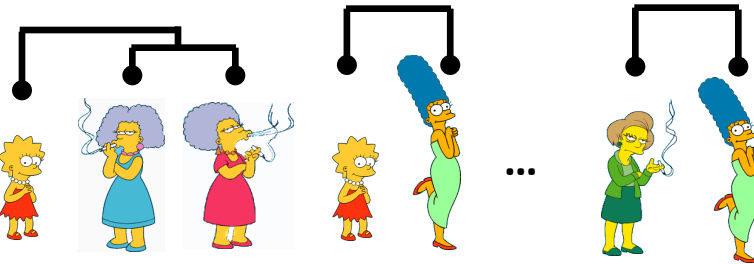
Consider all possible merges...



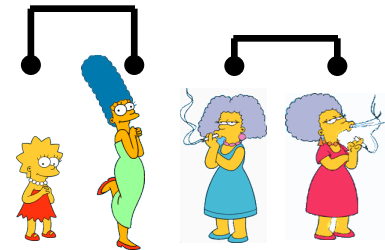
Choose the best



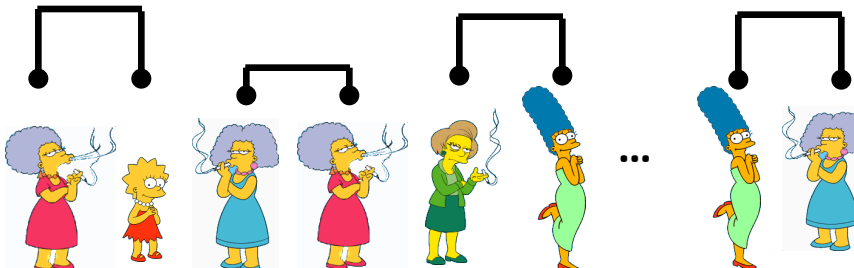
Consider all possible merges...



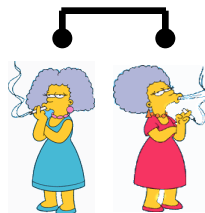
Choose the best



Consider all possible merges...



Choose the best



Distance between clusters: *Linkage*

We know how to measure the distance between two objects, but defining the distance between an object and a cluster, or defining the distance between two clusters is non obvious.

Complete linkage : In this method, the distances between clusters are determined by the greatest distance between any two objects in the different clusters (i.e., by the "furthest neighbors").

Single linkage : In this method the distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters.

Group average linkage: In this method, the distance between two clusters is calculated as the average distance between all pairs of objects in the two different clusters.

Distance between clusters

More formally

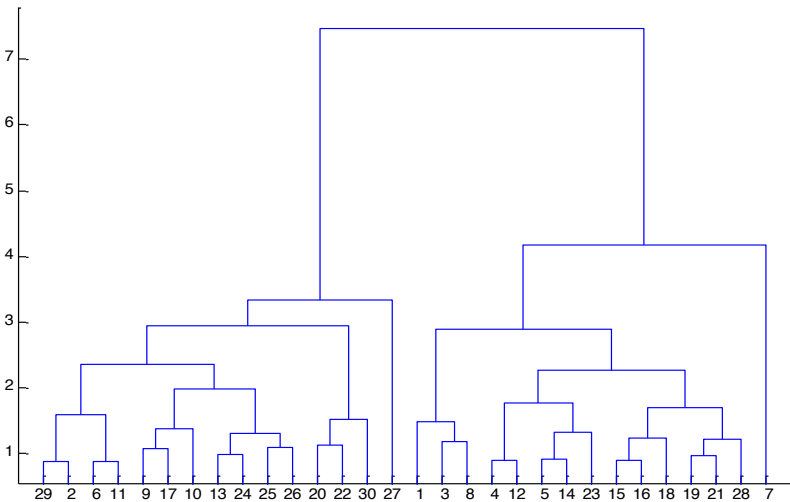
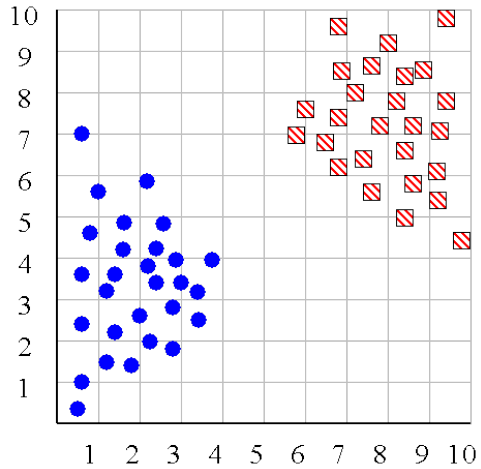
Let $D := N \times N$ be the dissimilarity matrix between individual objects (also called connectivity matrix), C_1, C_2 two clusters and $c_{1i} \in C_1, c_{2j} \in C_2$

Complete linkage : $d(C_1, C_2) = \max(d(c_{1i}, c_{2j}) \in D)$

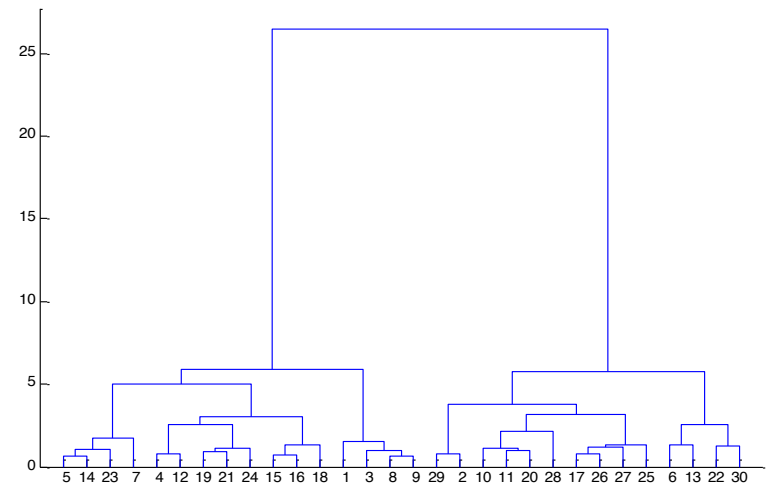
Single linkage : $d(C_1, C_2) = \min(d(c_{1i}, c_{2j}) \in D)$

Group average linkage:
$$d(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{i=1}^n |C_1| \sum_{j=1}^n |C_2| d(c_{1i}, c_{2j}) \in D$$

Distance between clusters



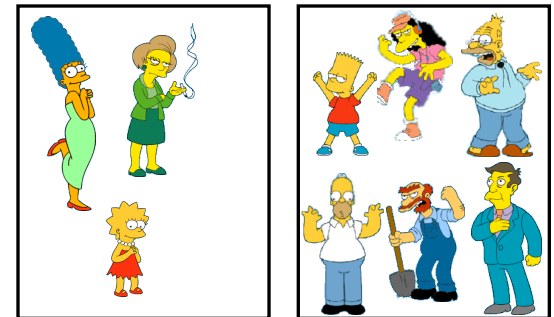
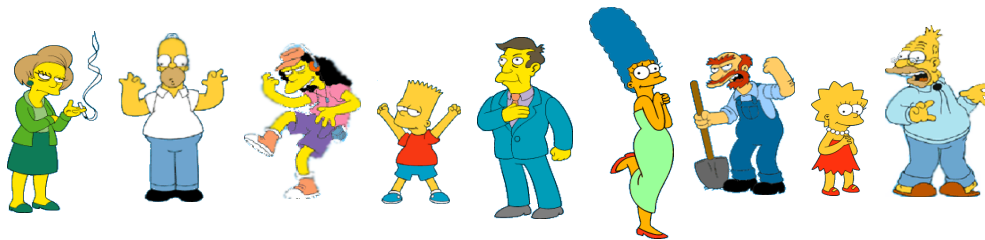
Average linkage



Wards linkage

Partitional Clustering

- Nonhierarchical, each instance is placed in exactly one of K non-overlapping clusters.
- Since only one set of clusters is output, the user normally has to input the desired number of clusters K .



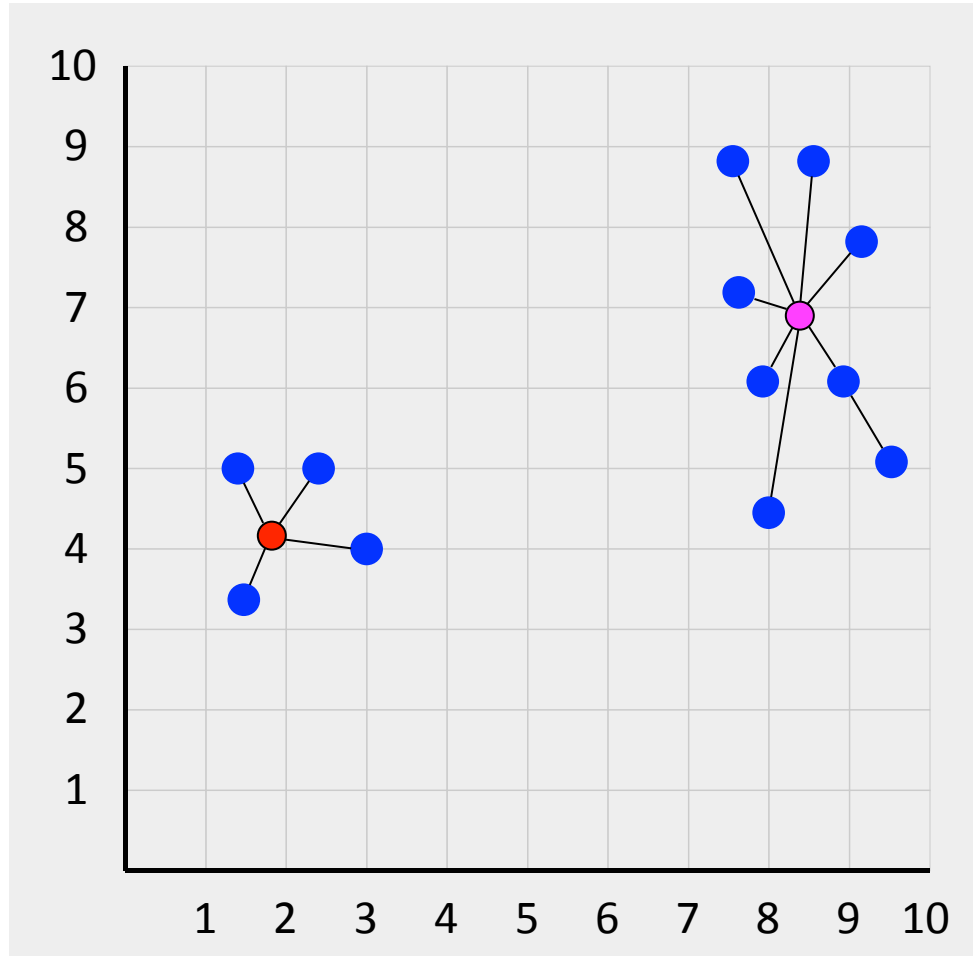
Partitional Clustering

- General characteristics of the approach:
 - Each of the k clusters C_j is represented by the mean (or weighted average) c_j of its objects, the centroid
 - The clusters are iteratively recomputed to achieve stable centroids
- A popular measure for the intra-cluster variance is the square-error criterion

$$E = \sum_{i=1}^k \sum_{p \in C_i} ||p - m_i||^2$$

with m_i as the mean of cluster of C_i

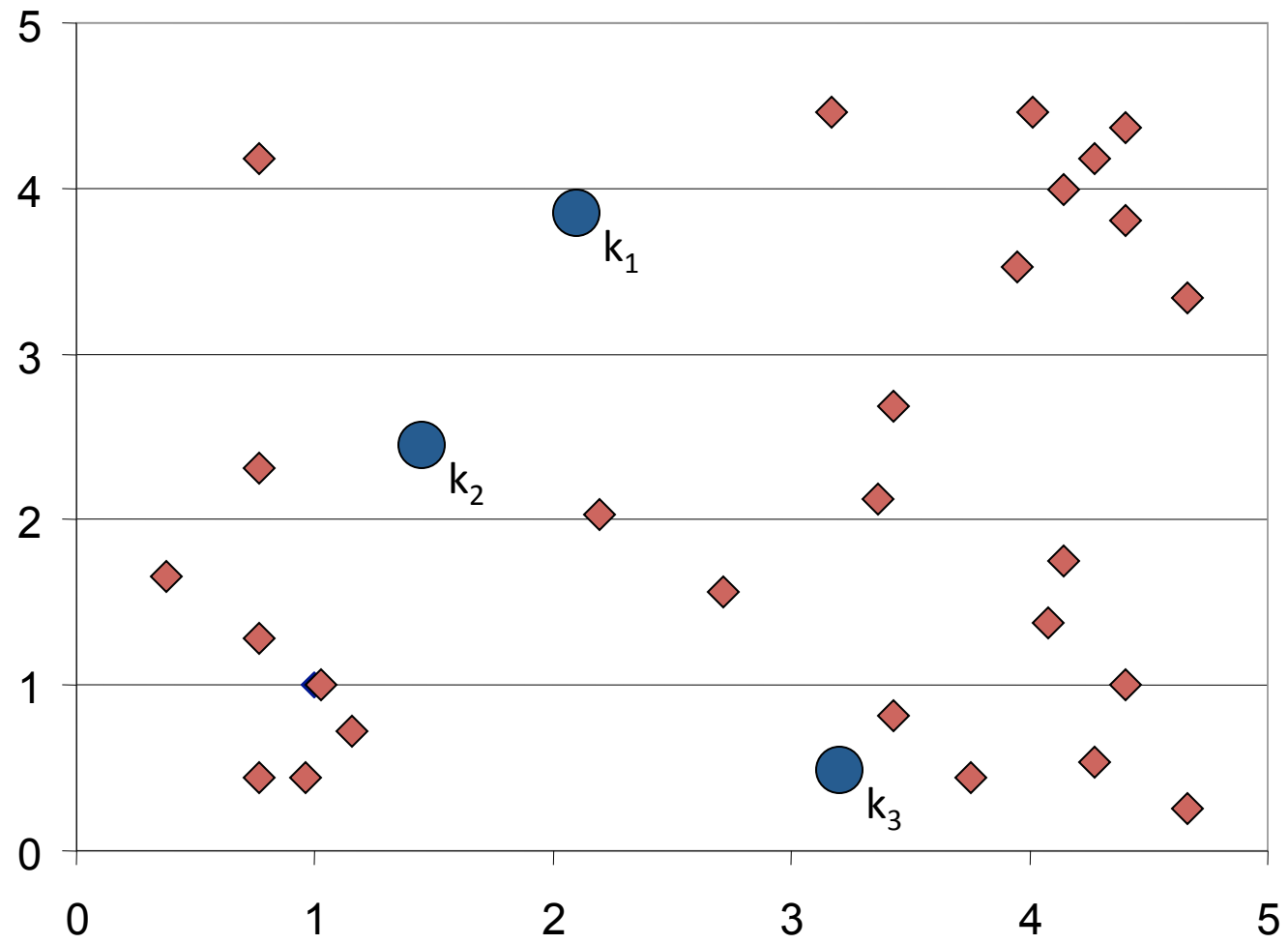
Partitional Clustering



K -means algorithm

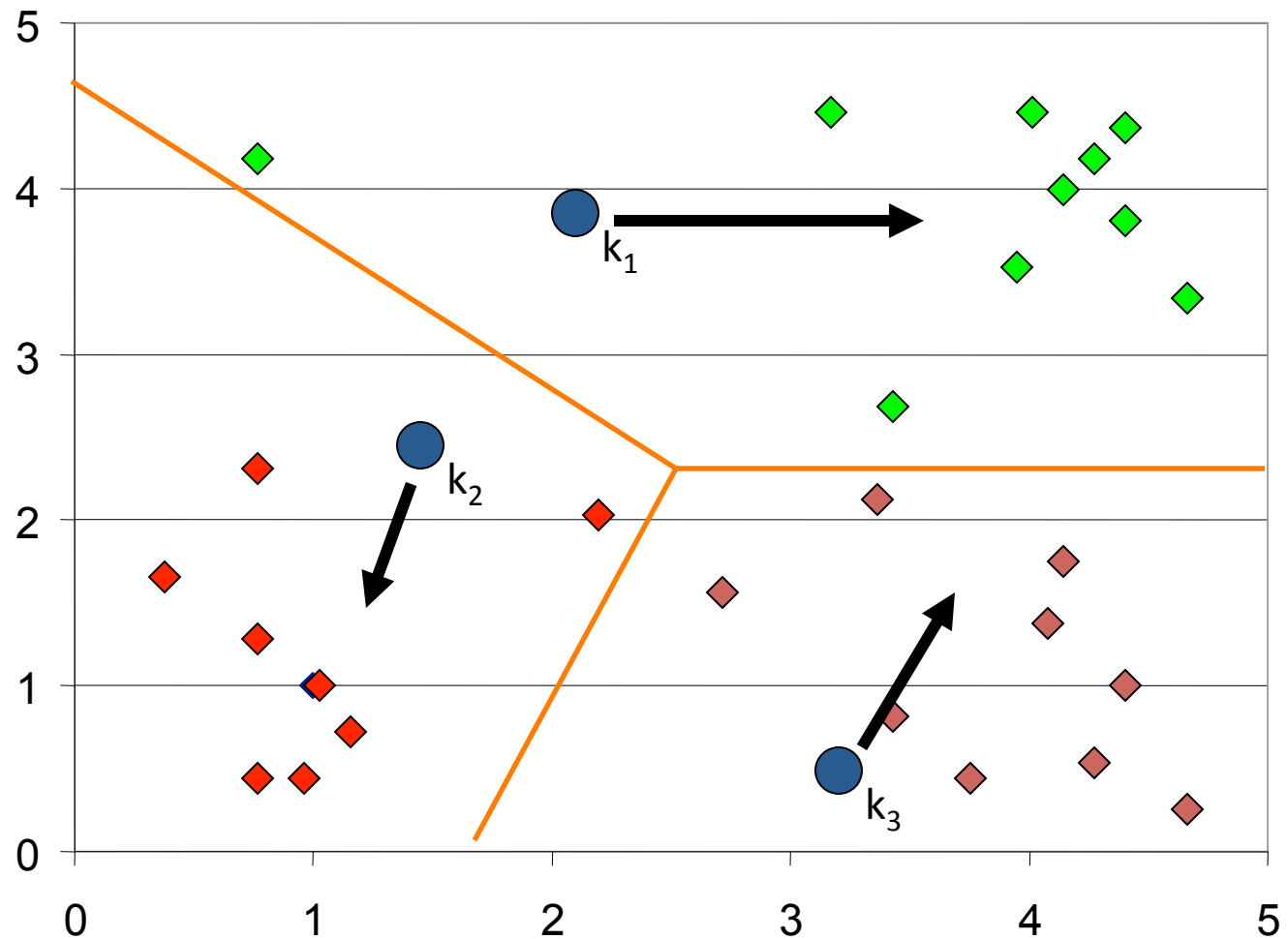
1. Decide a value for k .
2. Initialize the k cluster centers (randomly, if necessary).
3. Decide the class memberships of the N objects by assigning them to the nearest cluster center.
4. Re-estimate the k cluster centers, by assuming the memberships found above are correct.
5. If none of the N objects changed membership in the last iteration, exit. Otherwise go to 3.

K -means algorithm: Step 1



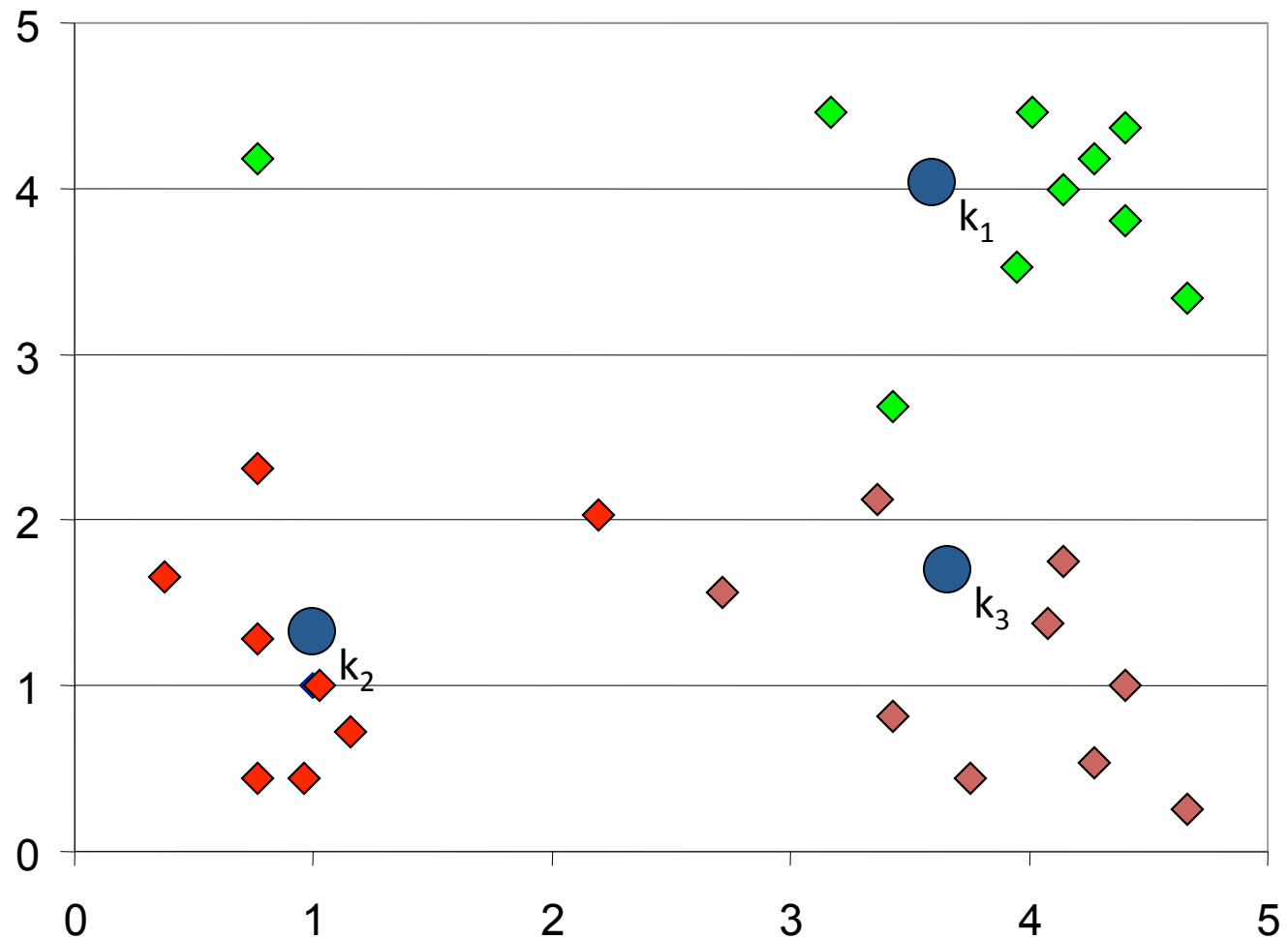
Algorithm: k -means, Distance Metric: Euclidean Distance

K -means algorithm: Step 2



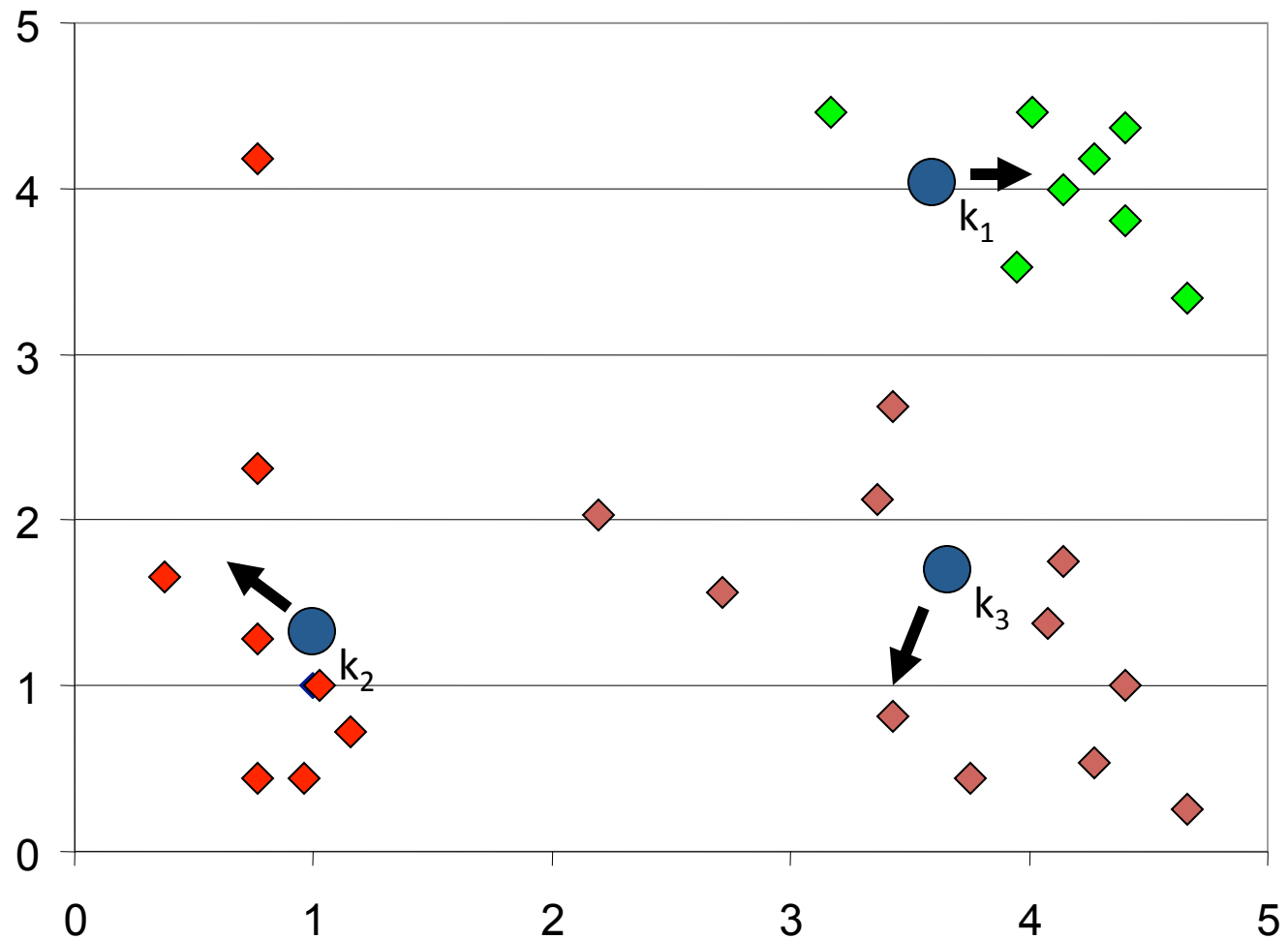
Algorithm: k -means, Distance Metric: Euclidean Distance

K -means algorithm: Step 3



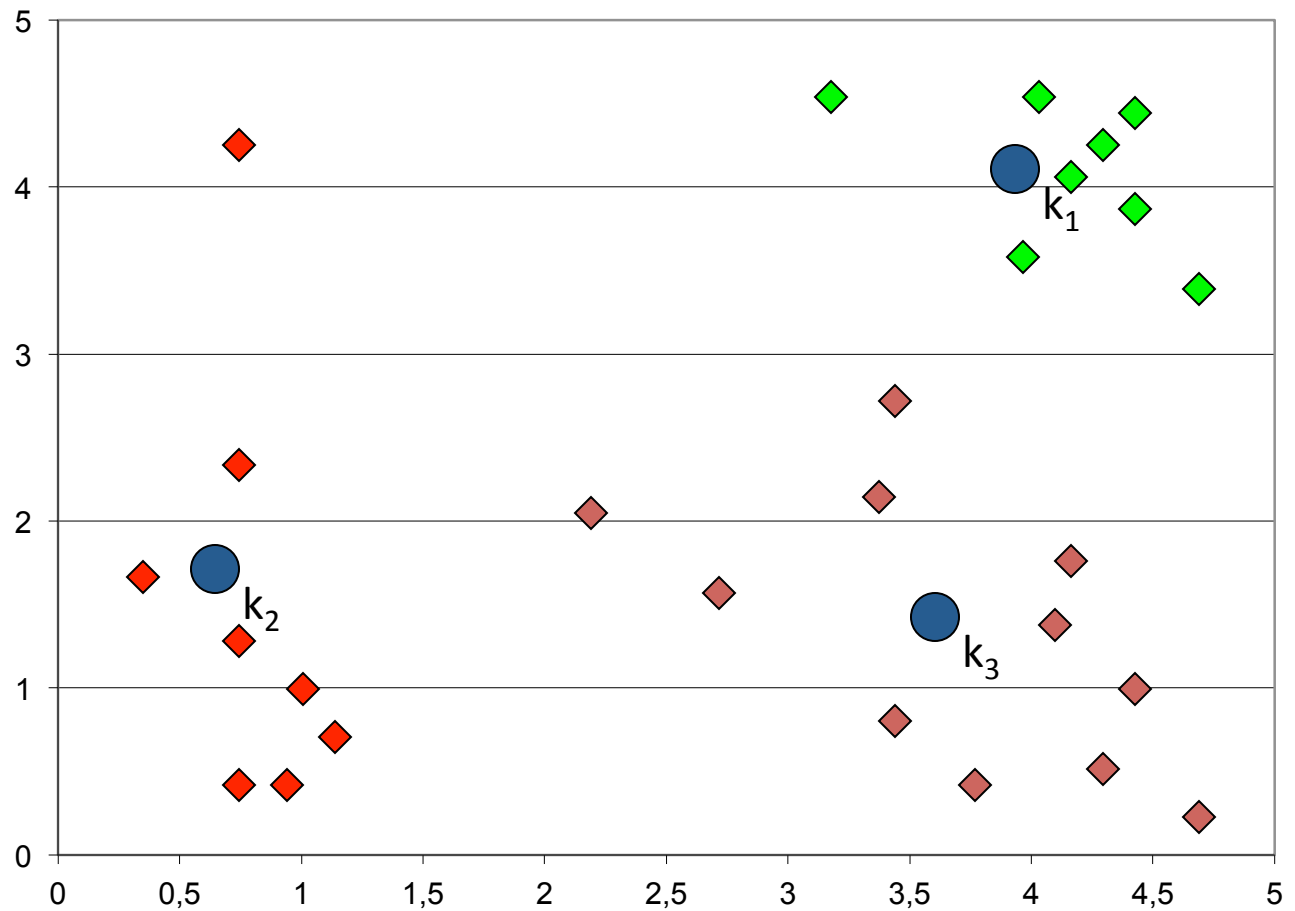
Algorithm: k -means, Distance Metric: Euclidean Distance

K -means algorithm: Step 4



Algorithm: k -means, Distance Metric: Euclidean Distance

K -means algorithm: Step 5



Algorithm: k -means, Distance Metric: Euclidean Distance

K -medoids algorithm

Medoids

A medoid is a representative object of a cluster whose average dissimilarity to all the data points in the cluster is minimal

$$M^* = \operatorname{argmin}_M \sum_i \min_k d(x_i, m_k)$$

Basic steps of K -medoids algorithm

1. Determine a set of k medoids
2. Construct k clusters by assigning each data point to its nearest medoid

K-medoids algorithm: issues

Two of the most difficult tasks in cluster analysis are:

- How to decide the appropriate number of clusters
- How to distinguish a bad cluster from a good one

The *K*-medoids algorithm family uses ***silhouette*** to address these tasks

Each cluster is represented by one *silhouette*, showing which data point lie within the cluster, and which merely hold an intermediate position. The entire clustering is displayed by plotting all silhouettes into a single diagram, which thus illustrates the quality of the clusters.

K-medoids algorithm: issues

Construction of a silhouette:

1. Consider any object i of the data set, and let A denote the cluster of i
2. Calculate the average distance a_i of i to all other objects in A

$$a_i = \frac{1}{N_A} \sum_{j \in A, j \neq i} d(i, j)$$

1. Consider any cluster C different from A and define the average distance $d(i, C)$ of i to objects of C

$$d(i, C) = \frac{1}{N_C} \sum_{c \in C} d(i, c)$$

2. Compute $d(i, C)$ for all clusters $C \neq A$, and then select the smallest of those: $b_i = \min d(i, C), C \neq A$

K-medoids algorithm: issues

The silhouette width of i is defined as:

$$sil_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

- Data point with large silhouette width are well-clustered, others tend to lie between clusters
- For a given number of clusters k , the overall average silhouette width for the clustering is the average over all objects I

$$sil_{av} = \sum_i \frac{sil_i}{N}$$

- The number of clusters k can be determined by choosing a k that yields the highest average silhouette width

Number of Clusters

In general, this is a unsolved problem.
However there are many approximate methods.

Number of Clusters

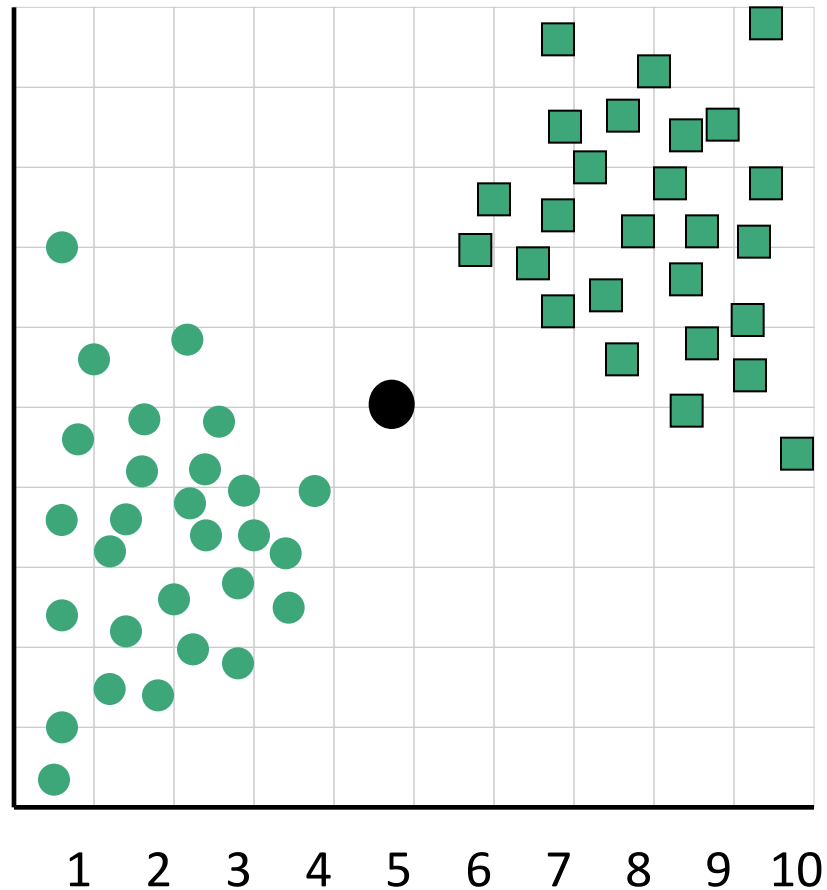
In general, this is a unsolved problem.

However there are many approximate methods.

- Suppose to have an objective function to evaluate the quality of your clustering.
- Suppose that objective function' values range between 0 and 1000.
- Suppose that the objective function should be minimize.

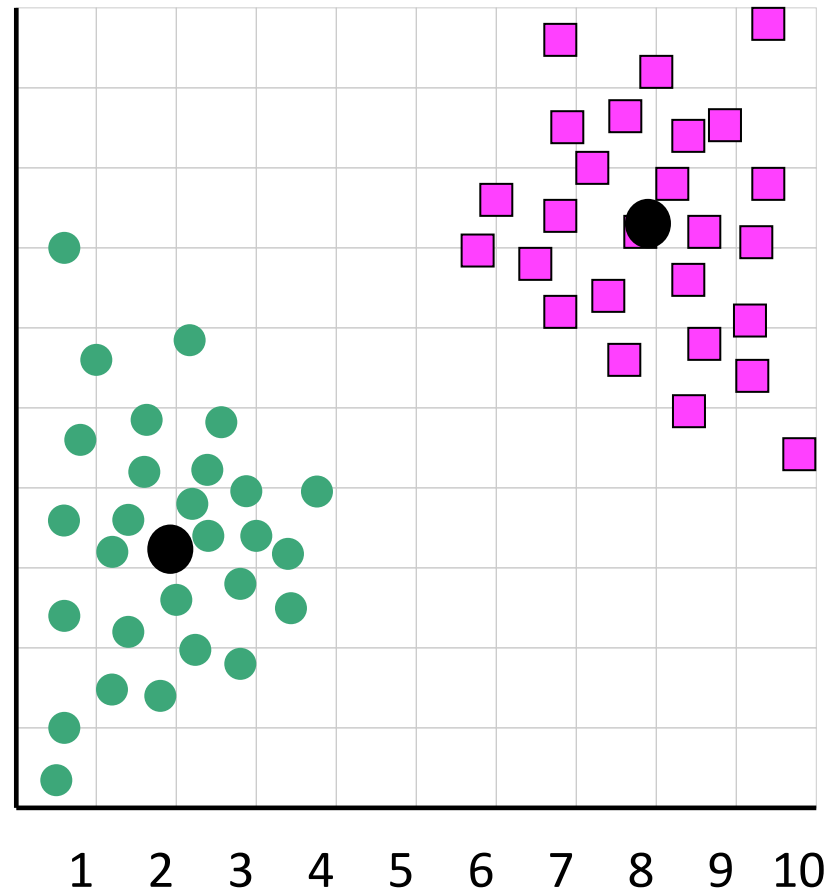
Number of Clusters

When $k = 1$, the objective function is 873.0



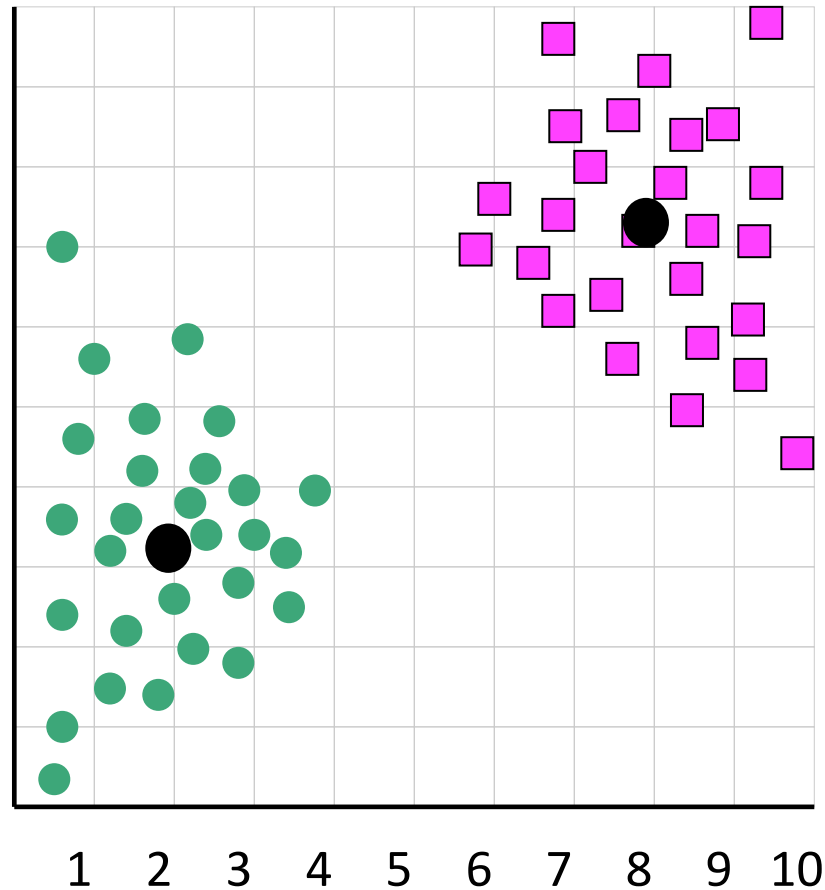
Number of Clusters

When $k = 2$, the objective function is 173.1



Number of Clusters

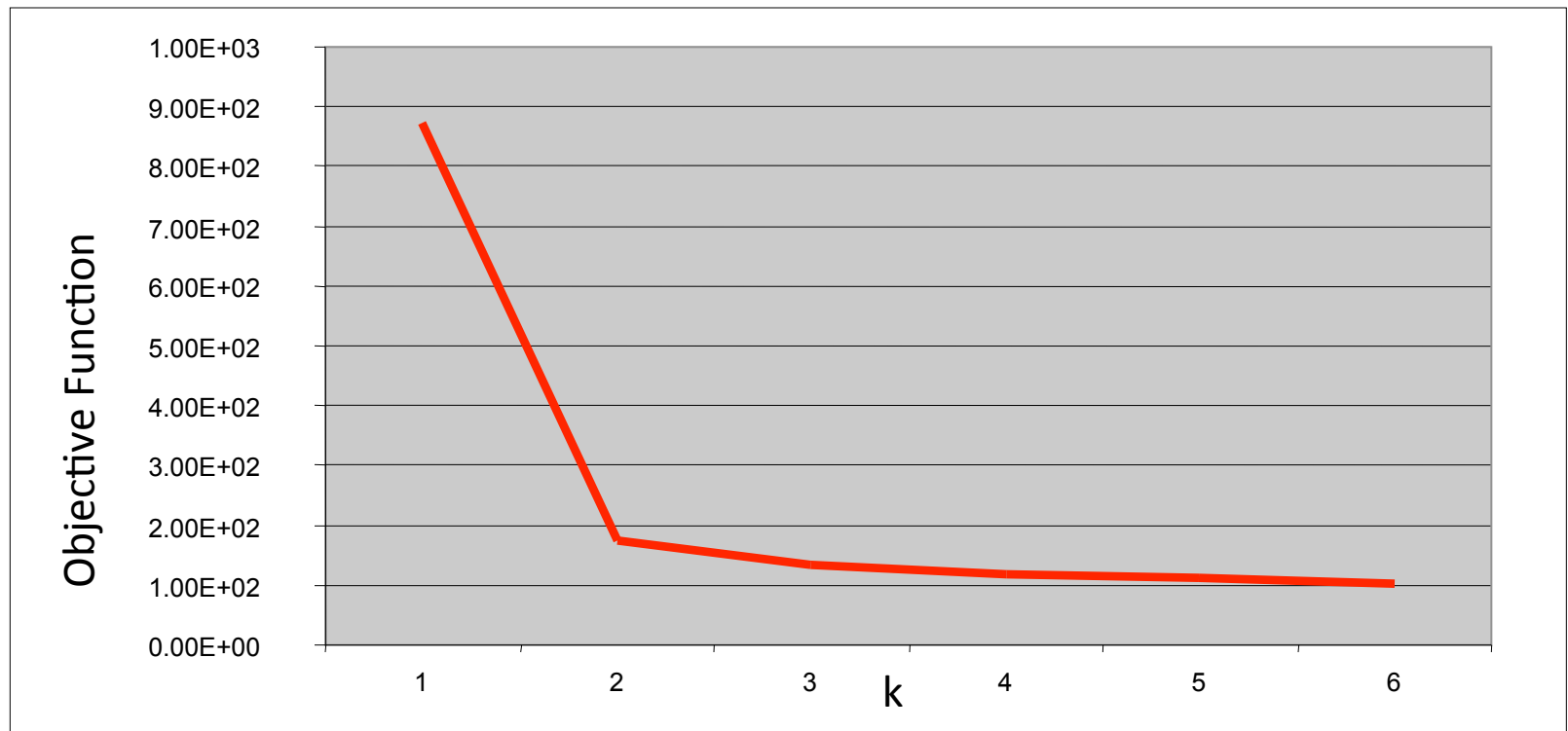
When $k = 3$, the objective function is 133.6



Number of Clusters

Plot the objective function values for k equals 1 to 6.

The abrupt change at $k = 2$, is highly suggestive of two clusters in the data.



Hackathon #2 – HTRU2 Data Set

Abstract.

Pulsar candidates collected during the HTRU survey.

Pulsars are a type of star, of considerable scientific interest.

Data Set Information:

HTRU2 is a data set which describes a sample of pulsar candidates collected during the High Time Resolution Universe Survey (South).

Pulsars are a rare type of Neutron star that produce radio emission detectable here on Earth. They are of considerable scientific interest as probes of space-time, the inter-stellar medium, and states of matter.

As pulsars rotate, their emission beam sweeps across the sky, and when this crosses our line of sight, produces a detectable pattern of broadband radio emission. As pulsars rotate rapidly, this pattern repeats periodically. Thus pulsar search involves looking for periodic radio signals with large radio telescopes.

Hackathon #2 – HTRU2 Data Set

Data Set Information:

Each pulsar produces a slightly different emission pattern, which varies slightly with each rotation. Thus a potential signal detection known as a 'candidate', is averaged over many rotations of the pulsar, as determined by the length of an observation. In the absence of additional info, each candidate could potentially describe a real pulsar. However in practice almost all detections are caused by radio frequency interference (RFI) and noise, making legitimate signals hard to find.

The data set shared here contains 16,259 spurious examples caused by RFI/noise, and 1,639 real pulsar examples. These examples have all been checked by human annotators.

Hackathon #2 – HTRU2 Data Set

Attribute Information

Each candidate is described by 8 continuous variables.

The first four are simple statistics obtained from the integrated pulse profile (folded profile). This is an array of continuous variables that describe a longitude-resolved version of the signal that has been averaged in both time and frequency. The remaining four variables are similarly obtained from the DM-SNR curve

These are summarised below:

1. Mean of the integrated profile.
2. Standard deviation of the integrated profile.
3. Excess kurtosis of the integrated profile.
4. Skewness of the integrated profile.
5. Mean of the DM-SNR curve.
6. Standard deviation of the DM-SNR curve.
7. Excess kurtosis of the DM-SNR curve.
8. Skewness of the DM-SNR curve.

LEARN BY DOING!
ENJOY!