POLITECNICO
MILANO 1863

Machine Learning Methods for Communication Networks and Systems

Francesco Musumeci

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)

Politecnico di Milano, Milano, Italy

Part I – 5: Clustering

Outline

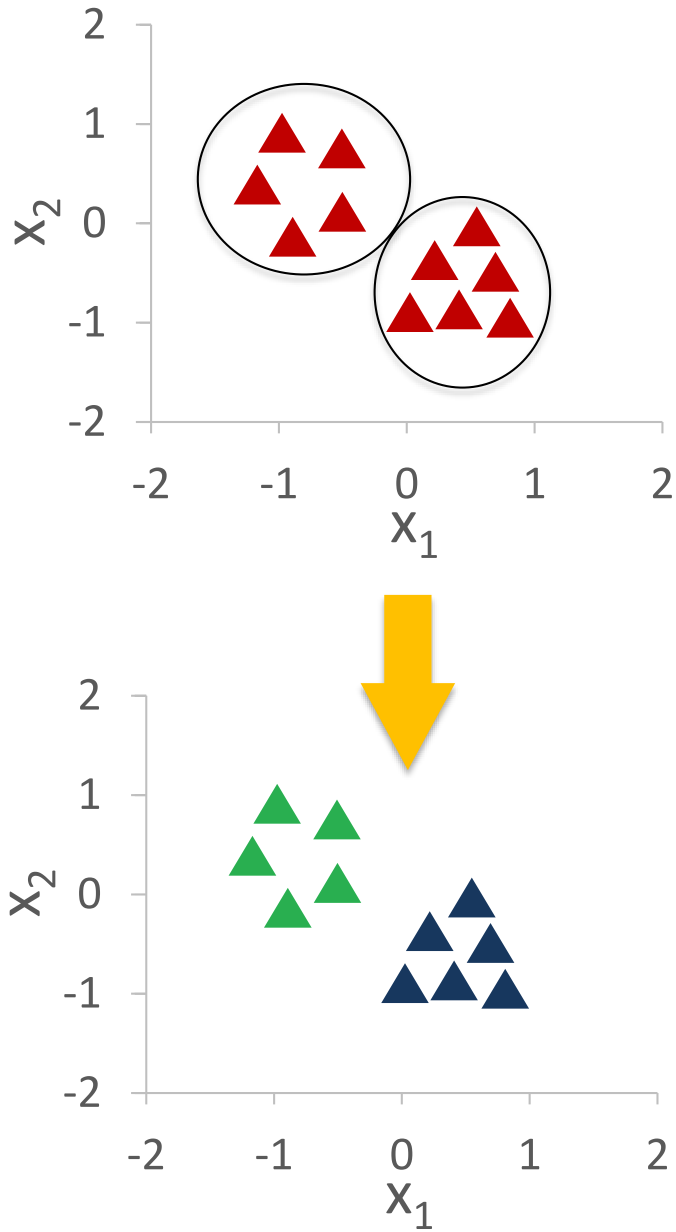
- Introduction
- K-means
- Hierarchical clustering
- Tips for clustering

Outline

- Introduction
- K-means
- Hierarchical clustering
- Tips for clustering

Introduction

- **Clustering** is part of *unsupervised* learning techniques
- Given a set of (unlabeled) examples $\underline{x}^{(i)}$, $i=1,2,\ldots,m$
- The objective is to find "structures" in the data
- Some examples:
  - Identify groups of similar users
  - Extract common traffic patterns from different cells in a mobile network
  - Market segmentation
- Often used before Classification

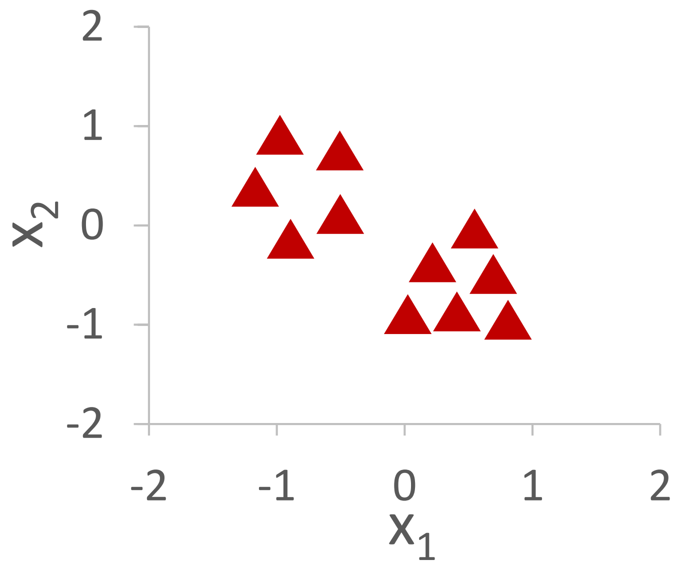


Outline

- Introduction
- K-means
- Hierarchical clustering
- Tips for clustering

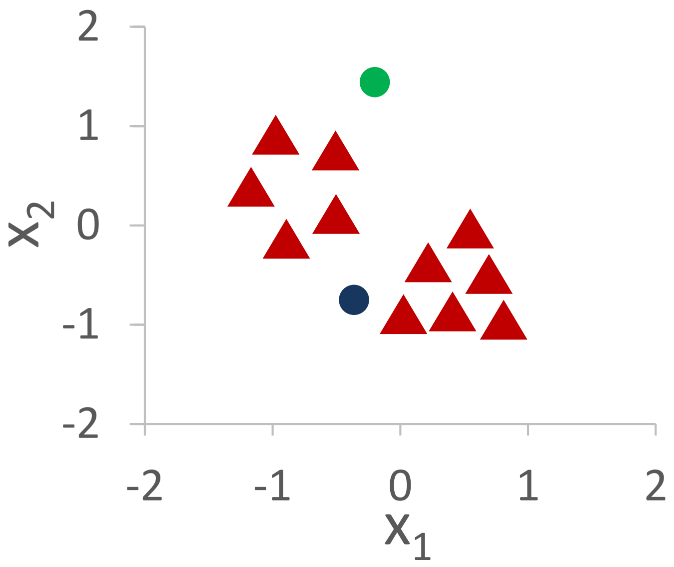
K-means

- Most popular clustering algorithm
- Iterative approach: randomly choose clusters *centroids*
  - assign examples to clusters and recalculate *centroids*
  - repeat until convergence



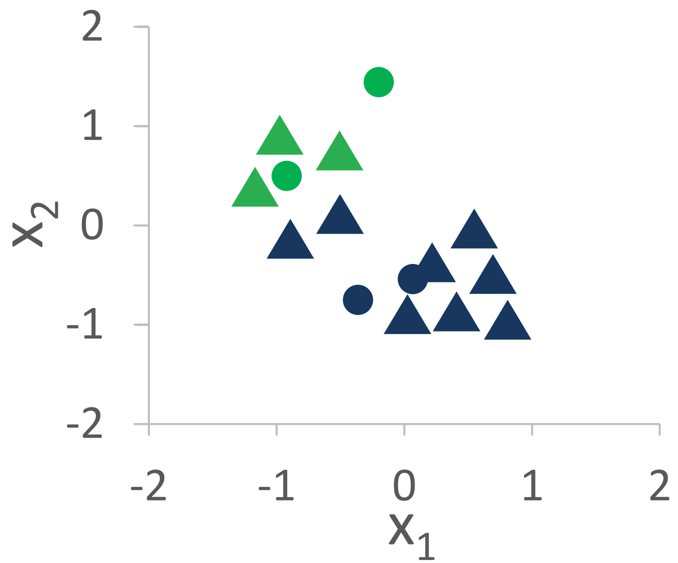
K-means

- Iterative approach: randomly choose clusters *centroids*
  - assign examples to clusters and recalculate *centroids*
  - repeat until convergence



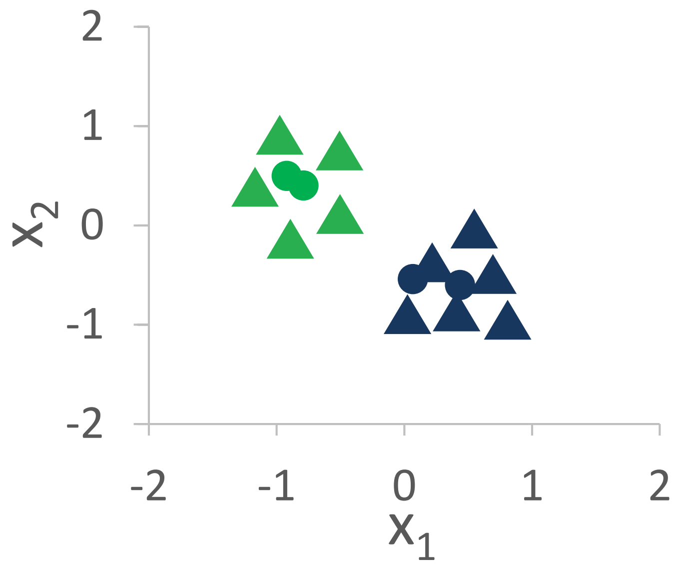
K-means

- Iterative approach: randomly choose clusters *centroids*
  - assign examples to clusters and recalculate *centroids*
  - repeat until convergence



K-means

- Iterative approach: randomly choose clusters *centroids*
  - assign examples to clusters and recalculate *centroids*
  - repeat until convergence



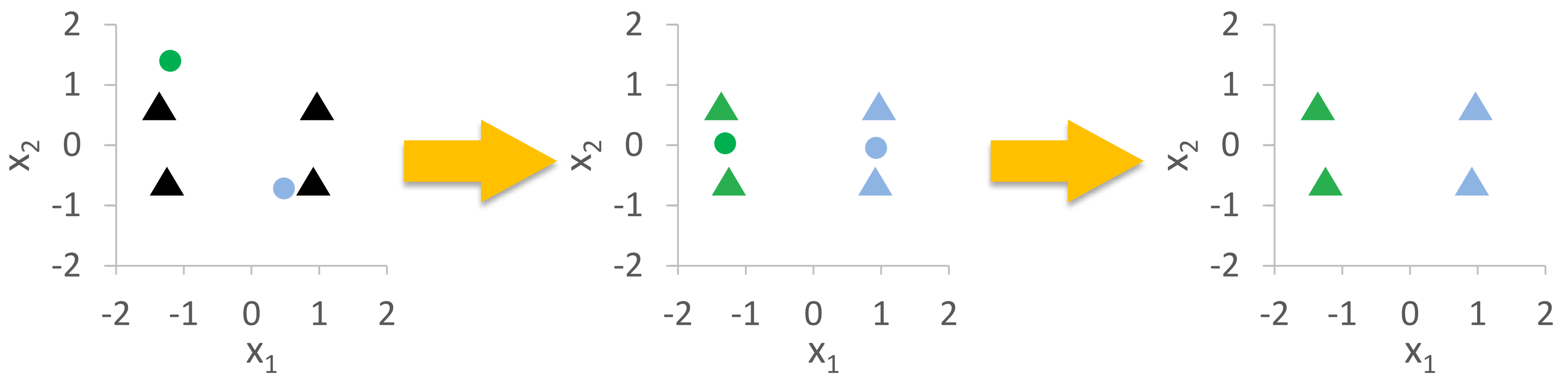
K-means

- More formally…
- Given:
  - training examples: $x^{(i)}=\{x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\}$ $i=1,2,\ldots,m$
  - K is the number of clusters (assumed!)
  - $c^{(i)}$: index of cluster for observ. $x^{(i)}$ ($c^{(i)}$ can be $=1, \ldots, K$)
- K-means algorithm:
  1. Randomly initialize clusters centroids $\mu_1, \mu_2, \ldots, \mu_K$
  2. Repeat until convergence:
     a. Cluster assignment: for $i=1,2,\ldots,m$, $c^{(i)}=\text{argmin}_j||x^{(i)}-\mu_j||$
     b. Update centroids: for $j=1,2,\ldots,K$, $\mu_j=1/n_j * \Sigma_{i:c(i)=j}\, x^{(i)}$

     where n_j is the n. of examples currently assigned to the j-th cluster
- *Cost function*: $J(c^{(1)}, c^{(2)}, \ldots, c^{(m)}; \mu_1, \mu_2, \ldots, \mu_K) = \frac{1}{m}\sum_{i=1}^{m}\left\| x^{(i)} - \mu_{c^{(i)}} \right\|$

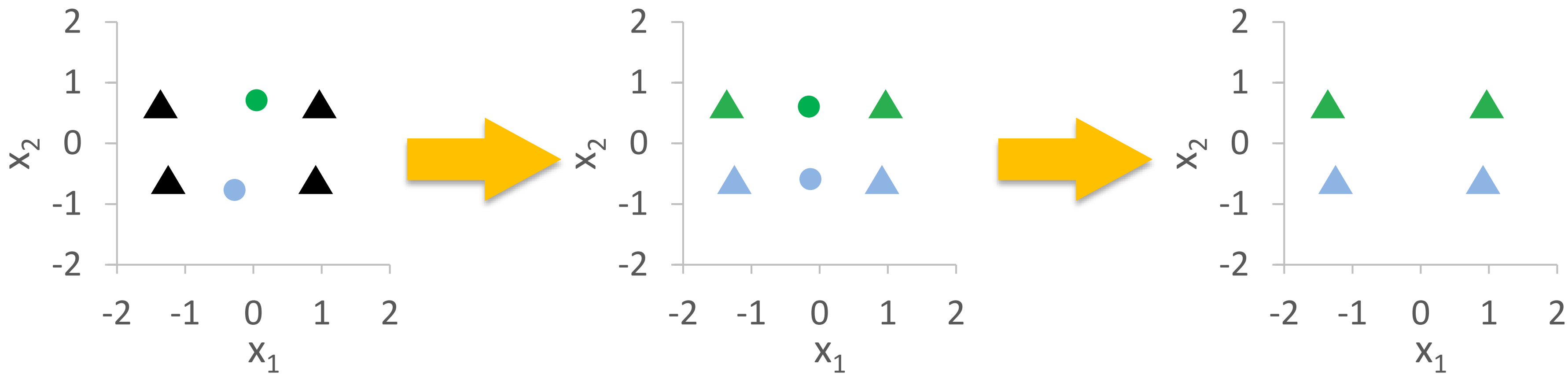
K-means
Issues

- <u>Random centroids initialization</u>
  - Different choices may lead to different clusters
    - Case 1



K-means
Issues

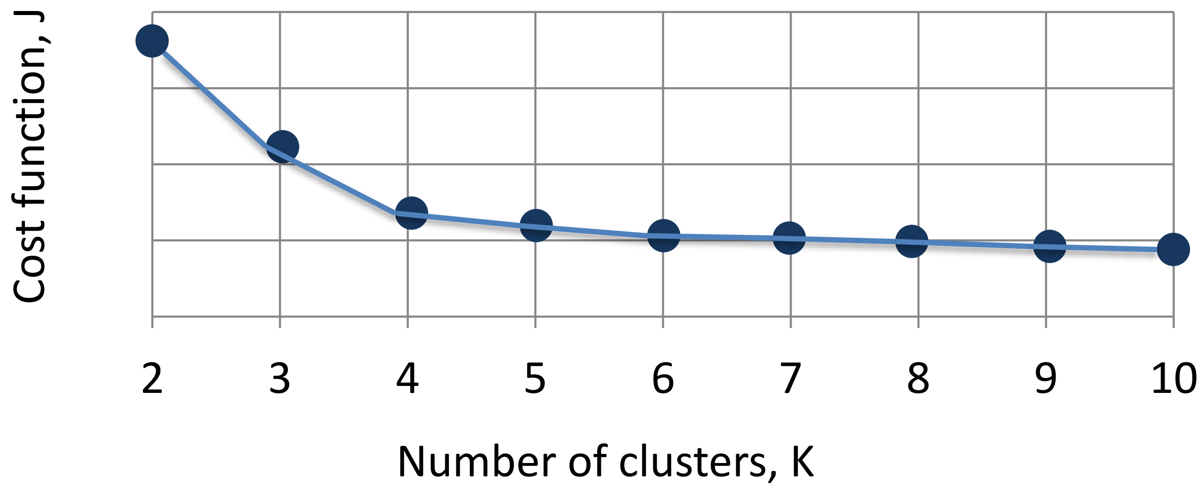
- <u>Random centroids initialization</u>
  - Different choices may lead to different clusters
    - Case 2



- Solution:
  - Perform several random initializations and calculate cost function *J(c;μ)*
  - Select clustering which provides the lowest *J(c;μ)*

K-means
Issues

- Selecting the number of clusters K
  - "Elbow" method



  - Maximize inter-cluster distance
  - Minimize intra-cluster distance
  - Combinations of inter/intra cluster distances
  - Silhouette coefficient
  - Minimize problem-specific cost function

Outline

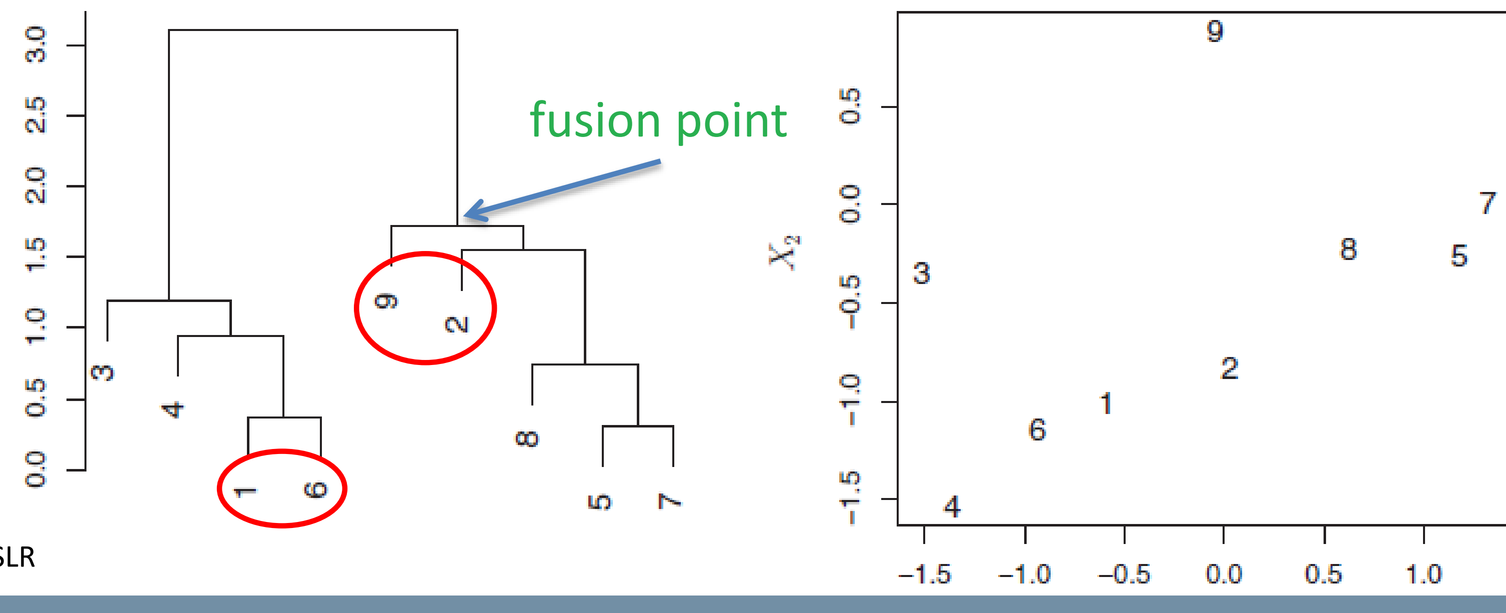
- Introduction
- K-means
- Hierarchical clustering
- Tips for clustering

Hierarchical clustering

- Clustering is performed according to the *dissimilarity* between (groups of) examples
  - Euclidean distance is the most used dissimilarity measure
- Iterative approach (*bottom-up*):
  1. Start considering every example as a single-element cluster
  2. Check dissimilarity between any pair of examples
  3. Group the least dissimilar (i.e., most similar) pair into a unique cluster
  4. Re-calculate dissimilarity between pairs of clusters
     - Need to specify inter-cluster dissimilarity (**linkage**)
     - N.B.: Linkage is different than dissimilarity of two examples
  5. Group least dissimilar (i.e., most similar) clusters into a unique cluster
  6. Repeat steps 4-5 until only one cluster remains
     - A *dendogram* can be drawn

Hierarchical clustering
Dendogram

- Tree-based representation of examples and their clustering
  - The height of the fusion points is a measure of the *dissimilarity* between the fused clusters (the higher, the more dissimilar, i.e., the less similar)
    - E.g.: 1-6 are very similar; 9-(2-8-5-7) are very dissimilar
  - Final clustering is decided setting a threshold on the fusion points



Source: ISLR

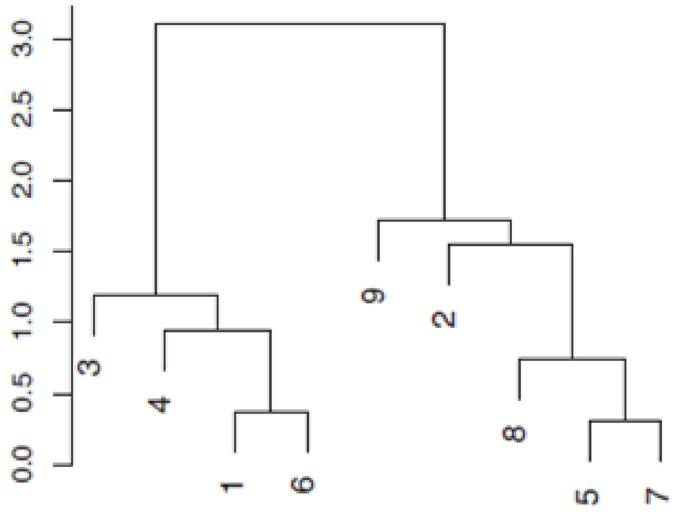
Hierarchical clustering
Linkage

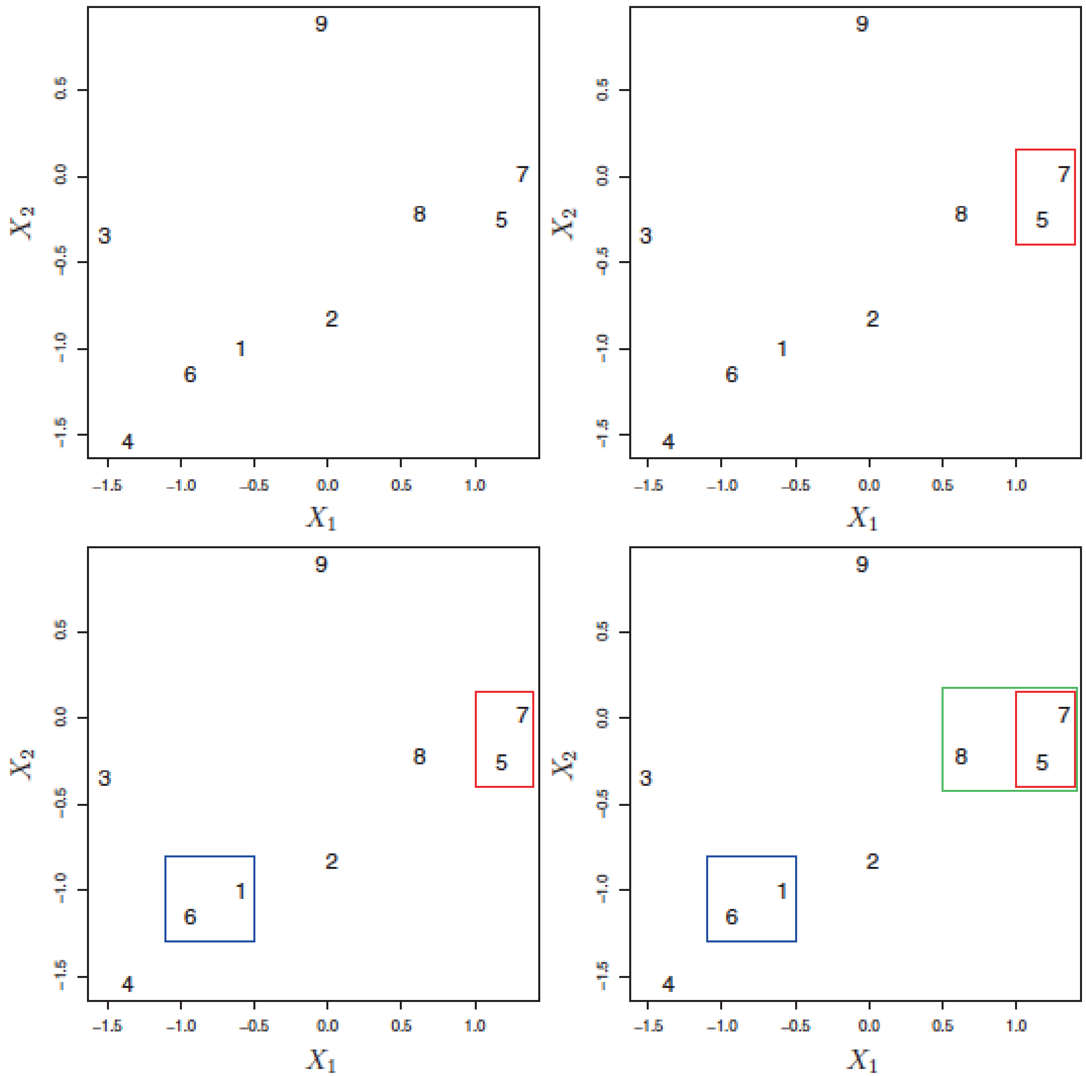
- How do we define inter-cluster (dis)similarity for clusters A & B?
  - *Complete linkage*: compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the largest of these dissimilarities
  - *Single linkage*: compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the smallest of these dissimilarities
  - *Average linkage*: compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the average of these dissimilarities.
  - *Centroid linkage*: dissimilarity between the centroid (mean vector) for cluster A and the centroid for cluster B

Hierarchical clustering
Example

- Maximal inter-cluster dissimilarity

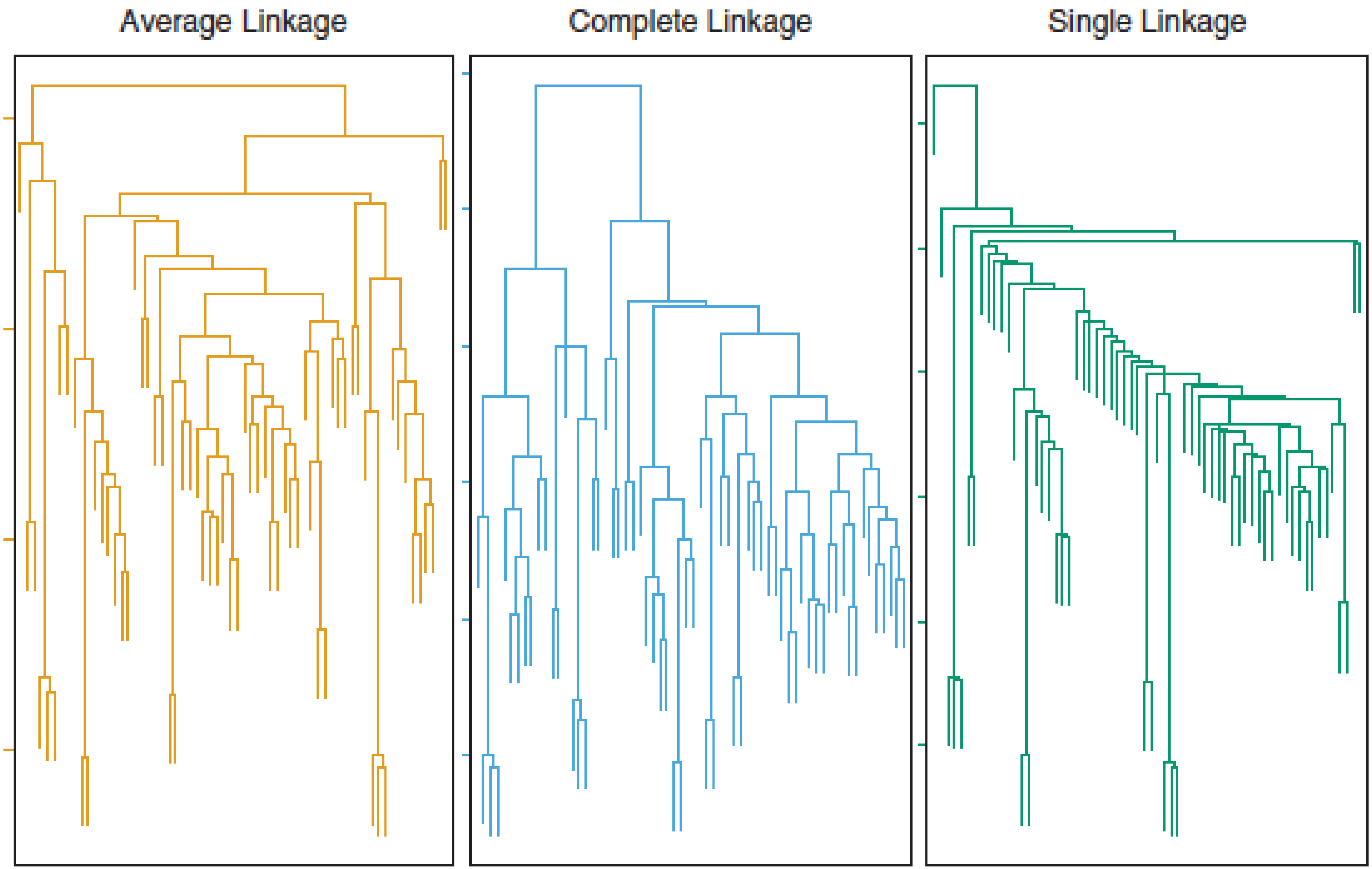
(*complete linkage*)





Source: ISLR

Hierarchical clustering
Effect of linkage on dendograms



Source: ISLR

Outline

- Introduction
- K-means
- Hierarchical clustering
- Tips for clustering

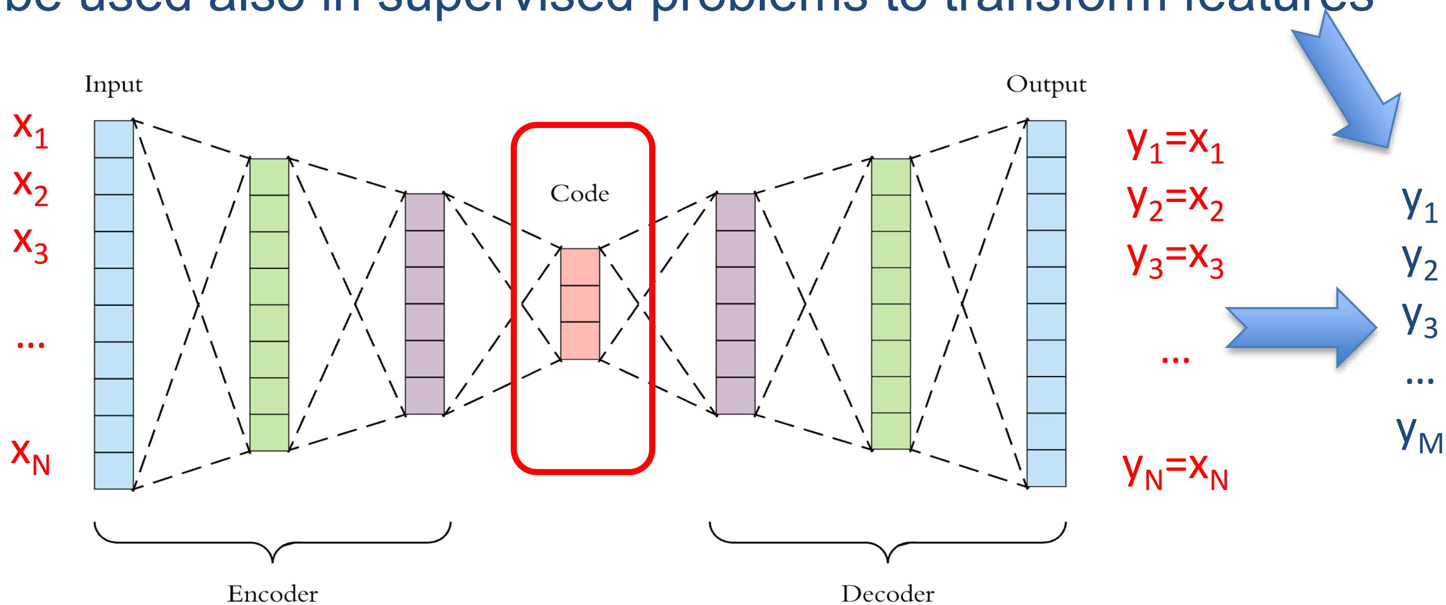
Tips for clustering

- *K*-means and hierarchical clustering force *every* observation into a cluster
  - clusters obtained may be heavily distorted due to the presence of outliers that do not belong to any cluster
  - density-based models (mean-shift, DBSCAN) are attractive approaches to handle with outliers
- Perform clustering with different choices of these parameters, and looking at the full set of results to see what patterns **consistently** emerge
- What is the proper cost function to minimize?
- What is the proper features set?
- Clustering subsets of the data in order to get a sense of the robustness of the clusters obtained
  - be careful about how to interpret the results of a clustering analysis
  - these results should not be taken as the absolute truth about a data set
  - starting point for the development of a scientific hypothesis and further study, preferably on an independent data set

Features selection for clustering

- As in classification, DNNs help in *automatic* features extraction…BUT…
- How can we use DNN in an **unlabelled** dataset?

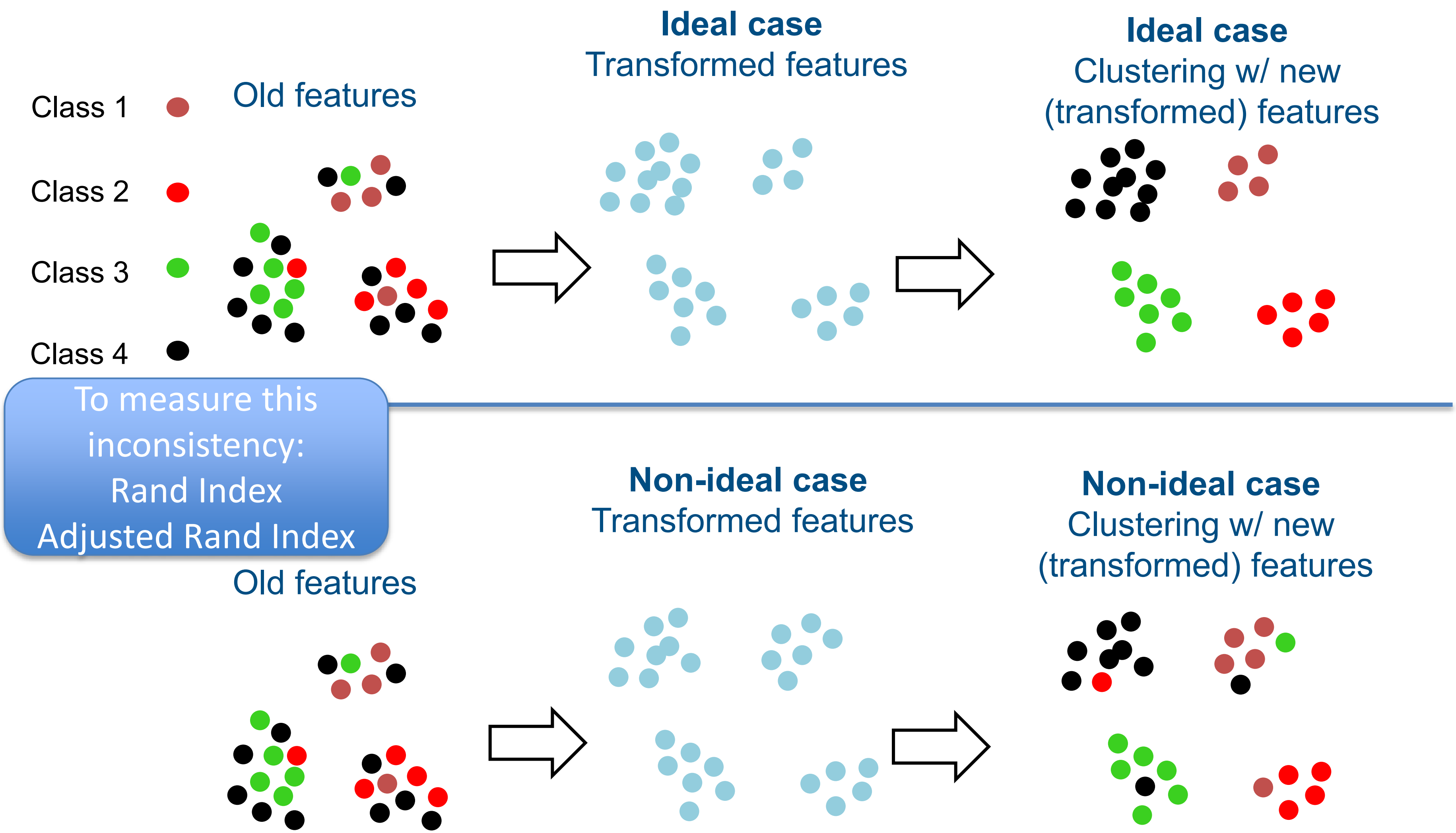
→ AUTOENCODERS: symmetrical DNNs
  - Trained using outputs = inputs
  - Encoder: maps inputs into coded features
  - Decoder: reconstructs the inputs from the encoded features
- Coded features can be used as a new features set
  - Can be used also in supervised problems to transform features



Clustering and supervised learning

- In some cases, clustering can be useful also for supervised problems
  - to obtain "better" features (separate classes more clearly)
  - to reduce dimensionality
    - another method is Principal Component Analysis (PCA), which uses ***linear*** transformation of original features
- It can be a basis for building an "automatic data labeler"
  - Semi-supervised learning and data augmentation
- How to evaluate if such features transformation is appropriate?
  - Usual clustering cost functions must be considered anyway, but…
  - We have knowledge of the labelled data:
    - how can we leverage this information?
    - are we changing the features space inappropriately? How to measure this ***numerically***?

Clustering and supervised learning



Clustering and supervised learning

- ## Rand index:

a: Number of pairs of elements that are in the same cluster in the original dataset and also after clustering

$$R = \frac{a+b}{\binom{n}{2}}$$

$\binom{n}{2}$
Total number of possible pairs on the dataset (without ordering)

b: The number of pairs of elements that are in different clusters in the original dataset but fall in the same cluster after doing clustering

$R \in [0, 1]$
$ARI \in [-1, 1]$

- ## Adjusted Rand index :
  - Given a set *S* of *n* elements, and two clustering:

Y: Clustering

$X = \{X_1, X_2, \ldots, X_r\}$ and $Y = \{Y_1, Y_2, \ldots, Y_s\}$

X: original dataset

the overlap between *X* and *Y* can be summarized in a **contingency table**, where each entry denotes the number of elements in common between X_i and Y_j : $n_{ij} = |X_i \cap Y_j|$

- **Contingency table**

$X \backslash Y$	Y_1	Y_2	$\ldots$	Y_s	Sums
X_1	n_{11}	n_{12}	$\ldots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\ldots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\ldots$	n_{rs}	a_r
Sums	b_1	b_2	$\ldots$	b_s	

- So the **ARI** is calculated as:

$$\overbrace{ARI}^{\text{Adjusted Index}} = \frac{\overbrace{\sum_{ij}\binom{n_{ij}}{2}}^{\text{Index}} - \overbrace{[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}^{\text{Expected Index}}}{\underbrace{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}]}_{\text{Max Index}} - \underbrace{[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}_{\text{Expected Index}}}$$

ARI: notable values

- **ARI = 1**
  n = 9

- **ARI = 0**
  n = 9

- **ARI → -1**
  n → ∞