Machine Learning

ITAM

Outline

- Unsupervised learning
- Clustering: Grouping and data segmentation
- Similarity measures
  - Transformation of ordinal, nominal and categorical variables
- Techniques
  - Partition methods
    - EM: k-medias
  - Density methods
  - Hierarchical methods

Objective

- Group data into categories or clusters such that instances that are more closely related belong to the same group
- Sometimes we also want a hierarchy that orders data accroding to their relatedness
  - E.g. A taxonomy

Unsupervised learning

- Clustering algorithms are unsupervised learning algorithms
  - There are no labeled examples from which the model is created
- Uses:
  - When labeling is expensive
  - When label change with time
  - When we want to find unsuspected relationships in the data that might be useful for classification
  - When we want to better understand the data
  - …..

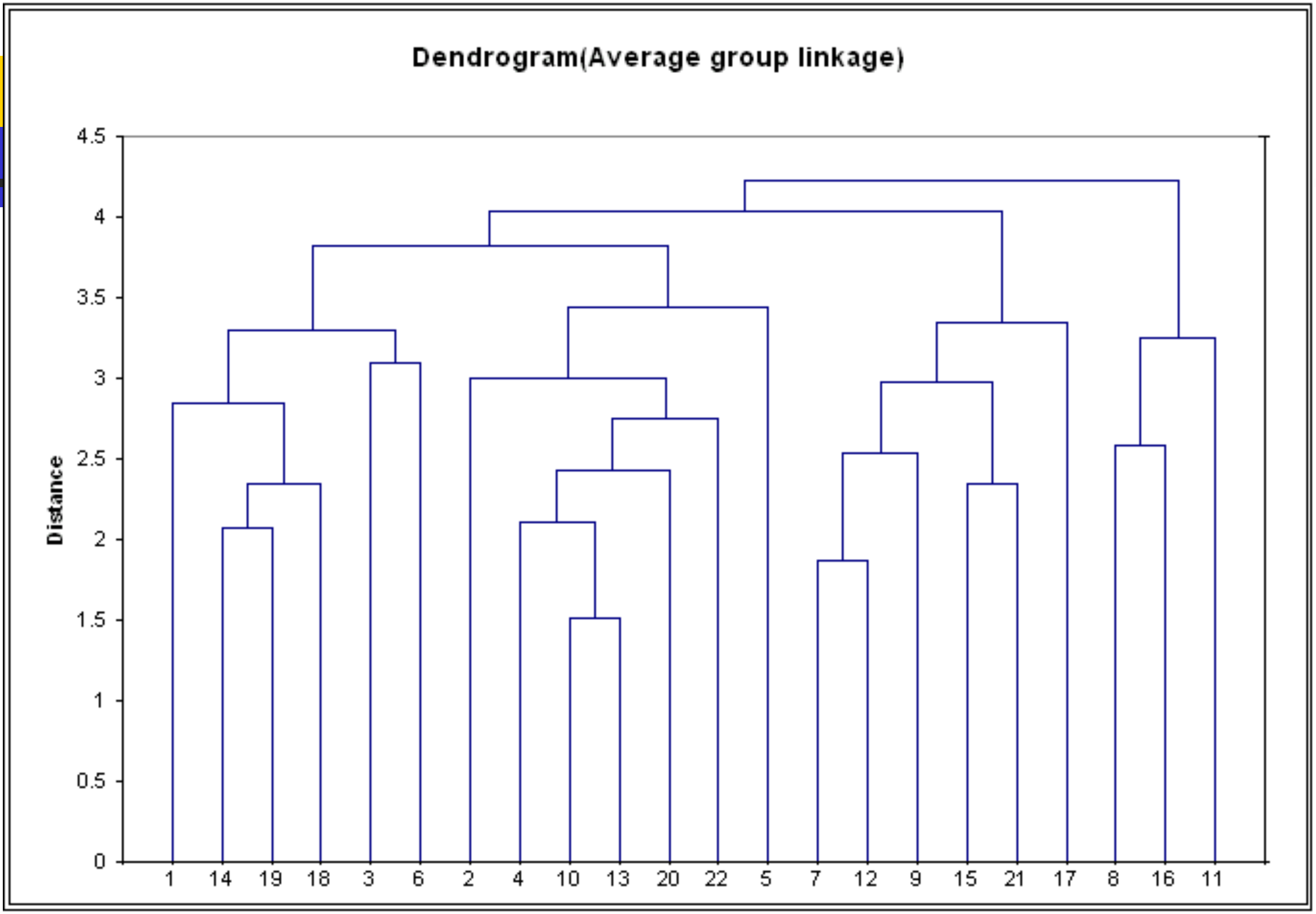
Hierarchical Methods

Clustering Algorithms
Hierarchical

- These algorithms create a hierarchy in which each level forms groups out of groups of a lower level
    - In the lowest level is the ungrouped data i.e., each data item belongs to its own group (every group is of size 1)
    - The highest level contains one group with all the data
- The resulting structure is a tree with the root being the group with all the data and the leaves the individaul data items. Inner node have different aggregations

Clustering Algorithms
Hierarchical

- The tree is drawn such that the height of each node is proportional to the dissimiliutud between the subgroups that it aggregates
- The resulting figure is know as a dendrogram and provides a descriptive image of the data



For example 1=Seattle 14=San Francisco, 19=LA, 18= Phoenix

Clustering Algorithms Hierarchical

- In contrast to k-means, the number clusters is not an intrinsic part of the algorithm
- It is the analyst's job to determine the level in the hierarchy that best represents the data's grouping
- Note:
  - These algorithms always find a hierarchy regardless of whether the data can truly be grouped
  - Different algorithms find different hierarchies
- Group formation is based on some similarity measure between groups

Dendrogram

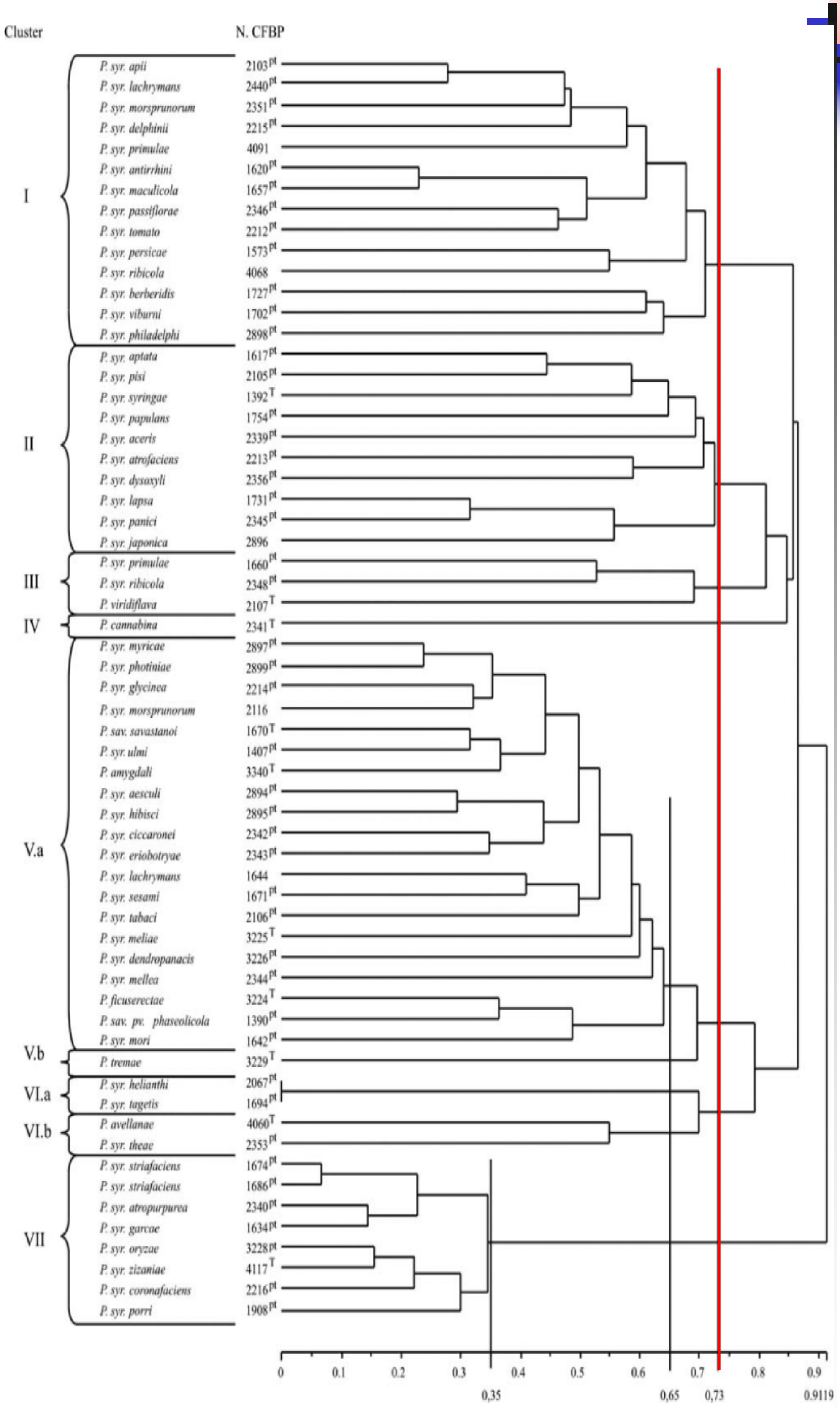


Figure 3 - Dendrogram obtained by comparison of BOX-PCR fingerprinting patterns from 61 bacterial type strains belonging to *Pseudomonas syringae* - *Pseudomonas viridiflava* large group (UPGMA analysis, Jaccard coefficient). Isolates obtained from the "Collection Française des Bactéries Phytopathogènes" (CFBP, Angers, France). T: species type strain, pt: pathotype strain.

Clustering Algorithms Hierarchical

- There are two strategies to create these structures
    - Aggregation
        - Start at the lowest level (individual instances) and recursively form larger and larger groups, until you have only one
    - Disgregation
        - Start with the group with all the data and recursively divide it until there is only one item per group

Clustering Algorithms Hierarchical

- Aggregation algorithms
  - Merge the two most similar groups
  - Repeat until there is only one
- We need a way to compute the similarity between groups

Clustering Algorithms
Hierarchical

- Single Linkage
  - The distance between group A and B is the minimum distance between their individual items
  - Example (using squared Euclidean distance for the distance between items):
    - A={<1,2>,<3,2>}, B={<5,6>,<5,7>,<6,6>}
    - d(<1,2>,<5,6>)=32, d(<1,2>,<5,7>)=41, d(<1,2>,<6,6>)=41
    - d(<3,2>,<5,6>)=20, d(<3,2>,<5,7>)=29, d(<3,2>,<6,6>)=25
    - The distance between A and B is 20

Clustering Algorithms Hierarchical

- Complete Linkage
  - The distance or similarity between two groups is the distance between its most dissimilar items
  - From the above example
    - The distance between A and B is 41
- Group average
  - The distance between to groups is the average distance between every pair of items

$$\frac{1}{N_k N_L}\sum_{i\in C_k}\sum_{C_L} d(\mathbf{x_i},\mathbf{x}_j)$$

Clustering Algorithms
Notes

- Single Linkage
  - Compact groups due to chaining
  - Produces groups with large diameter
    - The diameter is the maximum distance between any pair of items
- Complete Linkage
  - The opposite from above. Groups with small diameter
  - It is more likely that noisy data cause inadequate groupings
- Group average
  - Its a good compromise from the above two

Algorithm

1. Start with each data item in its own group
2. Find the similarity between every pair of groups
3. Merge the two most similar groups
4. Repeat from 2 until there is only one group

Clustering algorithms

- Disgregation Algorithms
  1. Start with the group of all elements
  2. Take one of the groups with more than one element and partition into two or more groups
  3. Repeat from 2 until every element is in its own group
- The group partition must seek that elements within a group are more similar than elements in diferent groups
- Ideas?

Clustering algorithms

- Class excercise

Density methods

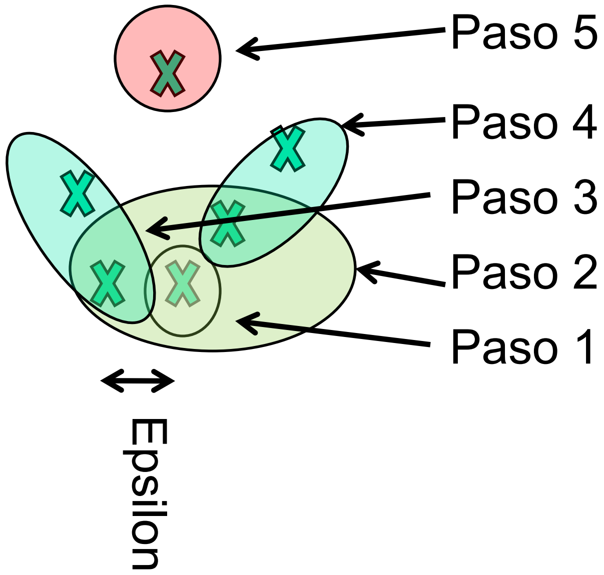
Clustering algorithms
Density (DBSCAN)

- Generate groups with a certain density
- It takes two parameters: Epsilon y Min
  - The distance between a data point and its closest neighbor in the same group is at most eplilos
  - Each group has at least Min elements

Clustering algorithms
DBSCAN

- D<-All the datas, i=0
- While (D <>{})
  - Extact any p from D (D<--D-{p})
  - V<--{p}, Ci<--{}
  - While(V<>{})
    - Extract any d from V (V<--V-{d})
    - O<--All of the data points from D at a distance Epsilon or less to d
    - If |O|+1<Min
      - if Ci={}, d is classified as noise
    - Else
      - V<--V U O, D<--D-V
      - Insert V and d in Ci: Ci<--Ci U V U {d}
  - If (Ci<>{})
    - i<--i+1

DBSCAN



Paso 5

Paso 4

Paso 3

Paso 2

Paso 1

Epsilon

Clustering algorithms
Example DBSCAN

- Epsilon= 1
- Min =2

Datos
23
8
12
13
4
22
14
3

Algoritmos de Agrupamiento
Ejercicio DBSCAN

- Del que hicieron para jerárquico ahora háganlo con DBSCAN usando
  - Probar con distintas Epsilons comezando con 0.2 (el valor de Epsilon dependerá de la medida de distancia que utilicen) Min=2

Clustering algorithms
Density

- No need to specify the number of groups
- Finds noisy data
- Groups have arbitrary shapes
- But defining epsilon is an art. Espilon is defined once per run so it can't find groups with different densities
  - Possible solution?