

# Pattern Recognition Lecture

## Hierarchical Clustering

Prof. Dr. Marcin Grzegorzek

Research Group for Pattern Recognition  
Institute for Vision and Graphics  
University of Siegen, Germany

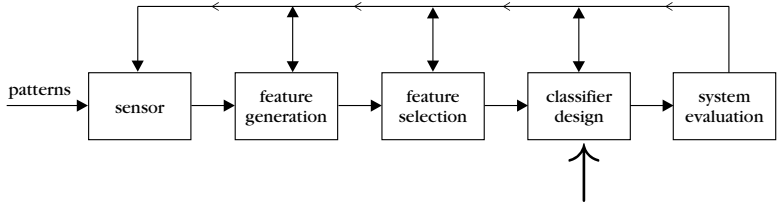


# Pattern Recognition Chain

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms



# Overview

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- 1 Introduction
- 2 Agglomerative Algorithms
- 3 Divisive Algorithms

# Overview

## Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- 1 Introduction
- 2 Agglomerative Algorithms
- 3 Divisive Algorithms

# General Idea and Applications

## Introduction

### Agglomerative Algorithms

### Divisive Algorithms

- Instead of producing a single clustering (like sequential algorithms), hierarchical algorithms produce a hierarchy of clusterings.
- This kind of algorithm is usually found in the social sciences and biological taxonomy.
- Further fields of application are: medicine, archaeology, computer science, and engineering.

# Initial Definitions

## Introduction

## Agglomerative Algorithms

## Divisive Algorithms

- Let us recall that  $X = \{\mathbf{x}_i, i = 1, \dots, N\}$  is a set of  $l$ -dimensional vectors that are to be clustered.
- Also recall the definition of clustering  $\mathbb{R} = \{C_j, j = 1, \dots, m\}$  where  $C_j \subset X$ .
- A clustering  $\mathbb{R}_1$  containing  $k$  clusters is said to be nested in the clustering  $\mathbb{R}_2$  which contains  $r < k$  clusters, if each cluster in  $\mathbb{R}_1$  is a subset of a set in  $\mathbb{R}_2$ . Note that at least one cluster of  $\mathbb{R}_1$  is a proper subset of a set in  $\mathbb{R}_2$  ( $\mathbb{R}_1 \neq \mathbb{R}_2$ ).
- If  $\mathbb{R}_1$  is nested in  $\mathbb{R}_2$  we denote it by  $\mathbb{R}_1 \sqsubset \mathbb{R}_2$ .

# Examples for the Term “Nested Clusterings”

## Introduction

### Agglomerative Algorithms

### Divisive Algorithms

- For example,  $\mathbb{R}_1 = \{\{\mathbf{x}_1, \mathbf{x}_3\}; \{\mathbf{x}_4\}; \{\mathbf{x}_2, \mathbf{x}_5\}\}$  is nested in  $\mathbb{R}_2 = \{\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4\}; \{\mathbf{x}_2, \mathbf{x}_5\}\}$
- But,  $\mathbb{R}_1$  is not nested in  $\mathbb{R}_3 = \{\{\mathbf{x}_1, \mathbf{x}_4\}; \{\mathbf{x}_3\}; \{\mathbf{x}_2, \mathbf{x}_5\}\}$ .
- It is clear that a clustering is not nested to itself.

# Two Main Categories of Hierarchical Algorithms

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- Hierarchical algorithms produce a hierarchy of nested clusterings.
- More specifically, these algorithms involve  $N$  steps, as many as the number of data vectors.
- At each step  $t$ , a new clustering is obtained based on the clustering produced at the previous step  $t - 1$ .
- There are two main categories of these algorithms, the agglomerative and the divisive hierarchical algorithms.

# Agglomerative Algorithms - General Idea

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- The initial clustering  $\mathbb{R}_0$  for the agglomerative algorithms consists of  $N$  clusters, each containing a single element of  $X$ .
- At the first step, the clustering  $\mathbb{R}_1$  is produced. It contains  $N - 1$  sets, such that  $\mathbb{R}_0 \subset \mathbb{R}_1$ .
- This procedure continues until the final clustering,  $\mathbb{R}_{N-1}$ , is obtained. It contains a single set, that is, the set of data,  $X$ . Notice that for the hierarchy of the resulting clusterings, we have:

$$\mathbb{R}_0 \subset \mathbb{R}_1 \subset \cdots \subset \mathbb{R}_{N-1}$$

# Divisive Algorithms - General Idea

## Introduction

## Agglomerative Algorithms

## Divisive Algorithms

- The divisive algorithms follow the inverse path. The initial clusterings  $\mathbb{R}_0$  consists of a single set,  $X$ .
- At the first step the clustering  $\mathbb{R}_1$  is produced. It consists of two sets, such that  $\mathbb{R}_1 \subset \mathbb{R}_0$ .
- This procedure continues until the final clustering  $\mathbb{R}_{N-1}$  is obtained. It contains  $N$  sets, each consisting of a single element of  $X$ . In this case we have

$$\mathbb{R}_{N-1} \subset \mathbb{R}_{N-2} \subset \cdots \subset \mathbb{R}_0$$

# Overview

Introduction

**Agglomerative  
Algorithms**

Divisive  
Algorithms

1 Introduction

**2 Agglomerative Algorithms**

3 Divisive Algorithms

# Generalised Agglomerative Scheme (GAS)

Let  $g(C_i, C_j)$  be a function defined for all possible pairs of clusters of  $X$ . This function measures the proximity between  $C_i$  and  $C_j$ . Let  $t$  denote the current level of hierarchy. Then, the general agglomerative scheme may be stated as follows:

- Initialisation
  - Choose  $\mathbb{R}_0 = \{C_i = \{\mathbf{x}_i\}, i = 1, \dots, N\}$
  - $t = 0$
- Repeat:
  - $t = t + 1$
  - Among all possible pairs of clusters  $(C_r, C_s)$  in  $\mathbb{R}_{t-1}$  find the one, say,  $(C_i, C_j)$ , such that

$$g(C_i, C_j) = \begin{cases} \min_{r,s} g(C_r, C_s) & \text{if } g \text{ is a dissimilarity function} \\ \max_{r,s} g(C_r, C_s) & \text{if } g \text{ is a similarity function} \end{cases}$$

- Define  $C_q = C_i \cup C_j$  and produce the new clustering  
 $\mathbb{R}_t = \{\mathbb{R}_{t-1} \setminus \{C_i, C_j\}\} \cup \{C_q\}$
- Until all vectors lie in a single cluster.

# GAS - the Nesting Property

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- GAS creates a hierarchy of  $N$  clusterings, so that each one is nested in all successive clusterings, that is,  $\mathbb{R}_{t_1} \sqsubset \mathbb{R}_{t_2}$ , for  $t_1 < t_2$ ,  $t_2 = 1, \dots, N - 1$ .
- If two vectors come together into a single cluster at level  $t$  of the hierarchy, they will remain in the same cluster for all subsequent clusterings.
- A disadvantage of the nesting property is that there is no way to recover from a “poor” clustering that may have occurred in an earlier level of the hierarchy.

# GAS - Algorithm Complexity

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- At each level  $t$ , there are  $N - t$  clusters. Thus, in order to determine the pair of clusters that is going to be merged at the  $t + 1$  level,

$$\binom{N-t}{2} \equiv \frac{(N-t)(N-t-1)}{2}$$

pairs of clusters have to be considered.

- Thus, the total number of pairs that have to be examined throughout the whole clustering process is

$$\sum_{t=0}^{N-1} \binom{N-t}{2} = \sum_{k=1}^N \binom{k}{2} = \frac{(N-1)N(N+1)}{6}$$

that is, the total number of operations required by an agglomerative scheme is proportional to  $N^3$ . However, the exact complexity of the algorithm depends on the definition of  $g$ .

# Pattern and Similarity Matrix

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- The pattern matrix  $D(X)$  is the  $N \times I$  matrix, whose  $i$ -th row is the transposed  $i$ -th vector of  $X$ .
- The similarity (dissimilarity) matrix  $P(X)$  is an  $N \times N$  matrix whose  $(i, j)$  element equals the similarity  $s(\mathbf{x}_i, \mathbf{x}_j)$  (dissimilarity  $d(\mathbf{x}_i, \mathbf{x}_j)$ ) between vectors  $\mathbf{x}_i$  and  $\mathbf{x}_j$ .

# Dendrogram

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- A dendrogram is an effective means of representing the sequence of clusterings produced by an agglomerative algorithm.

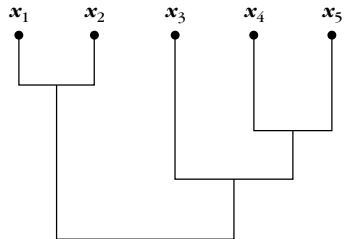
$\{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$

$\{\{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$

$\{\{x_1, x_2\}, \{x_3\}, \{x_4, x_5\}\}$

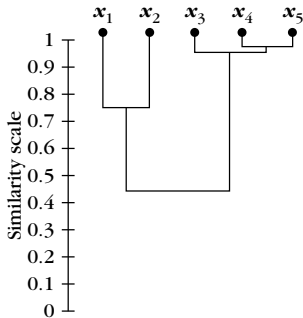
$\{\{x_1, x_2\}, \{x_3, x_4, x_5\}\}$

$\{\{x_1, x_2, x_3, x_4, x_5\}\}$

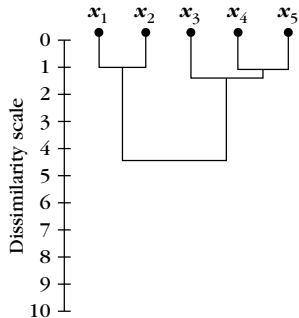


# Proximity Dendrogram

- A proximity dendrogram is a dendrogram that takes into account the level of proximity where two clusters are merged for the first time. We distinguish between similarity and dissimilarity dendrograms.



(a)



(b)

# Matrix Updating Algorithmic Scheme (MUAS)

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- Initialisation

- $\mathbb{R}_0 = \{\{\mathbf{x}_i\}, i = 1, \dots, N\}$
- $P_0 = P(X)$
- $t = 0$

- Repeat:

- $t = t + 1$
- Find  $C_i, C_j$  such that  $d(C_i, C_j) = \min_{r,s=1,\dots,N; r \neq s} d(C_r, C_s)$
- Merge  $C_i, C_j$  into a single cluster  $C_q$  and form
$$\mathbb{R}_t = \{\mathbb{R}_{t-1} \setminus \{C_i, C_j\}\} \cup \{C_q\}$$
- Define a proximity matrix  $P_t$  from  $P_{t-1}$  by<sup>1</sup>

- Until  $\mathbb{R}_{N-1}$  clustering is formed, that is, all feature vectors lie in the same cluster.

---

<sup>1</sup>(a) deleting the two rows and columns that correspond to the merged clusters and  
(b) adding a new row and a new column that contain the distances between the newly formed cluster and old clusters.

# Overview

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

1 Introduction

2 Agglomerative Algorithms

3 Divisive Algorithms

# General Idea

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- The divisive algorithms follow the reverse strategy from that of the agglomerative schemes.
- At the first step, we search for the best possible partition of  $X$  into two clusters. The straightforward method is to consider all possible  $2^{N-1} - 1$  partitions of  $X$  into two sets and to select the optimum according to a prespecified criterion.
- This procedure is then applied iteratively to each of the two sets produced in the previous stage.
- The final clustering consists of  $N$  clusters, each containing a single vector of  $X$ .

# Generalised Divisive Scheme (GDS) - Assumptions

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- The  $t$ -th clustering contains  $t + 1$  clusters.
- $C_{tj}$  denotes the  $j$ -th cluster of the  $t$ -th clustering  $\mathbb{R}_t$ .
- $g(C_i, C_j)$  is a dissimilarity function<sup>2</sup> defined for all possible pairs of clusters.
- The initial clustering  $\mathbb{R}_0$  contains only the set of  $X$ .

---

<sup>2</sup>The GDS can be also formulated for a similarity function, of course.

# Generalised Divisive Scheme (GDS)

Introduction

Agglomerative  
Algorithms

Divisive  
Algorithms

- Initialisation
  - Choose  $\mathbb{R}_0 = \{X\}$  as the initial clustering, e. g.,  $C_{01} = X$ .
  - $t = 0$
- Repeat:
  - $t = t + 1$
  - For  $i = 1$  to  $t$ 
    - Among all possible pairs of clusters  $(C_r, C_s)$  that form a partition of  $C_{t-1,i}$  find the pair  $(C_{t-1,i}^1, C_{t-1,i}^2)$  that gives the maximum value for  $g$ .
  - Next  $i$
  - From the  $t$  pairs defined in the previous step choose the one that maximises  $g$ . Suppose that this is  $(C_{t-1,i}^1, C_{t-1,i}^2)$ .
  - The new clustering is  $\mathbb{R}_t = \{\mathbb{R}_{t-1} \setminus \{C_{t-1,i}\}\} \cup \{C_{t-1,i}^1, C_{t-1,i}^2\}$
  - Relabel the clusters of  $\mathbb{R}_t$
- Until each vector lies in a single distinct cluster.