DECISION SUPPORT SYSTEMS

Time series in machine learning: time series clustering

Part 1

Co-funded by the Erasmus+ Programme of the European Union

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein

Author: Łukasz Paśko | Academic year: 2021/2022

Objectives of the classes

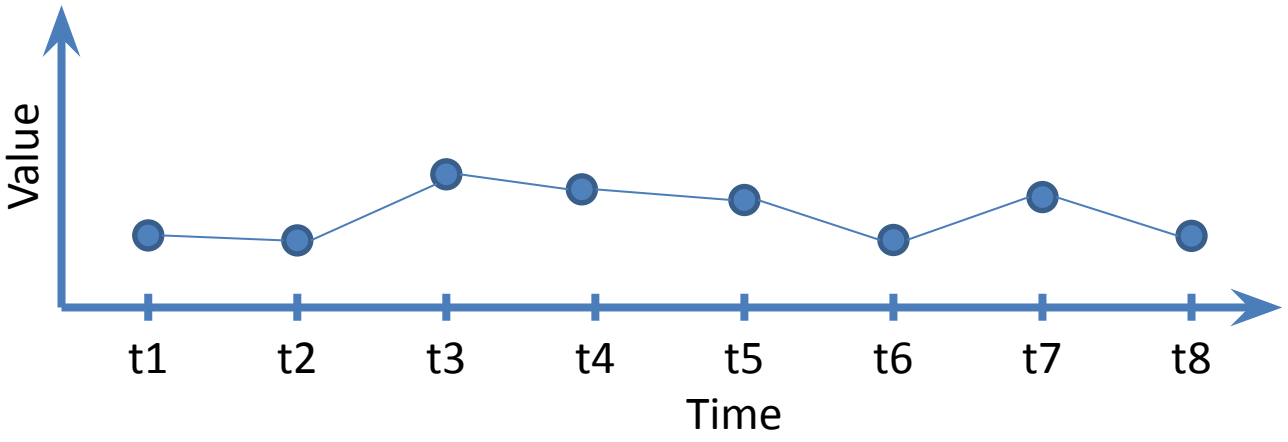
Presentation of the concept of time series clustering using machine learning techniques

Preprocessing of time series data

Application of hierarchical methods

Time series

A sequence of data (observations) that are ordered in time (the domain is time).

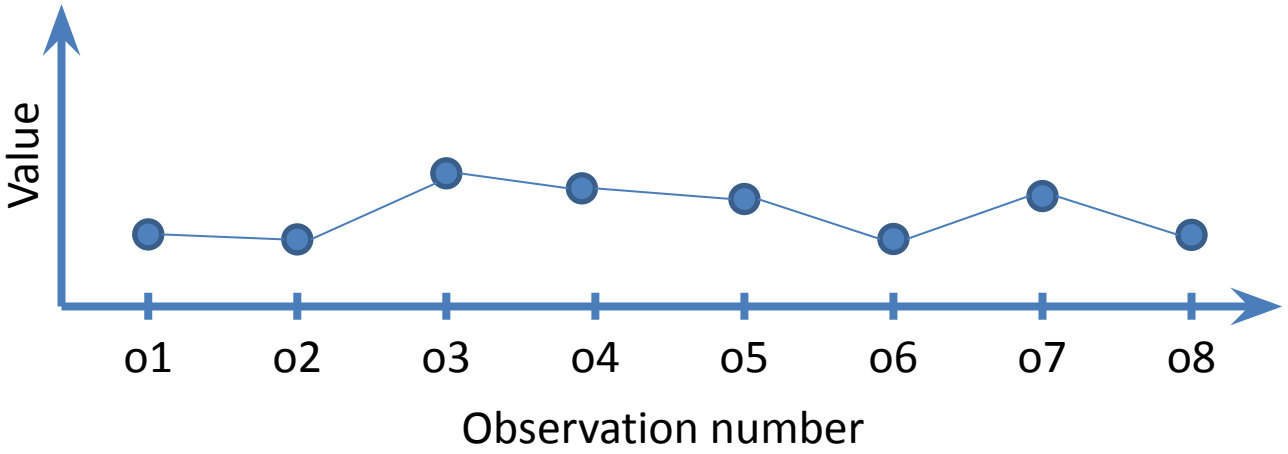


Examples:

- Values of certain parameters recorded every second by sensors during operation of a certain device (e.g., speed, vibration),
- The temperature in the production hall recorded every 10 minutes,
- The number of products produced during each working shift,
- Daily electricity consumption,
- The number of employees absent each day,
- Altitude above the ground during flight measured every second,
- ECG,
- Weekly store revenue,
- Daily bank account balance,
- Poland's unemployment rate by quarter,
- Stock market index values at the end of each day,
- and many more...

"Non-time" series

A string of data (observations) that are ordered (the order of the data is important).



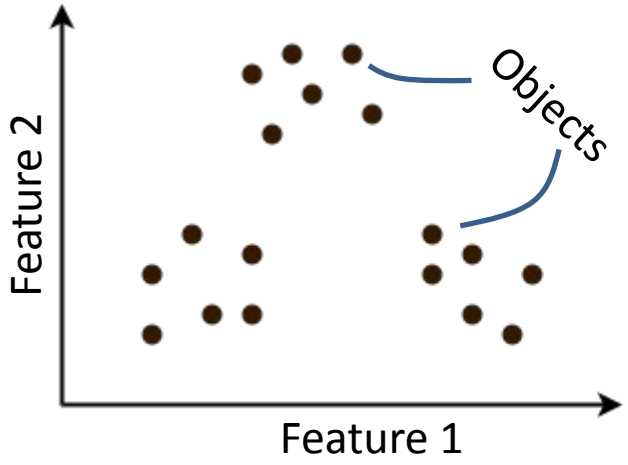
All of the time series analysis methods and techniques that will appear in today's class can also be used for "non-time" series.

Clustering

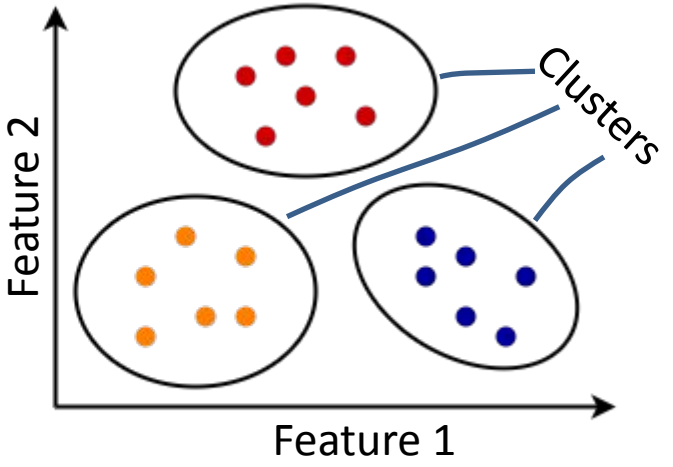
The task of finding clusters, also called groups, in a set of objects.

The goal: objects included in the same cluster must be similar to each other in terms of the characteristics under study, while objects from different clusters should be as different from each other as possible. Clustering is otherwise known as clustering or patternless classification.

Before clustering



After clustering



Time series clustering

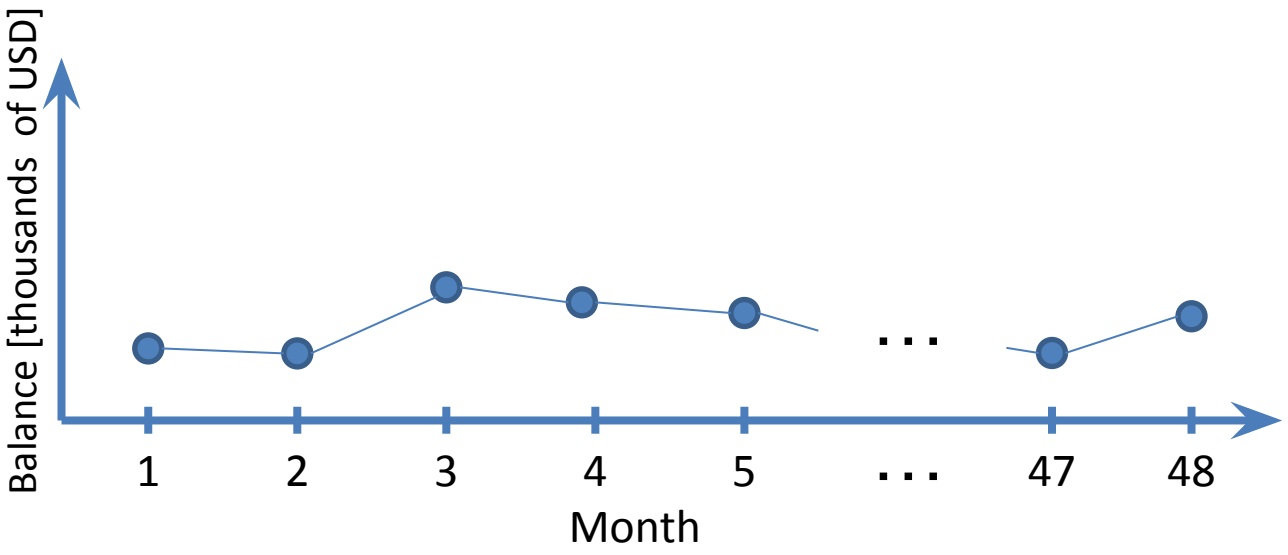
Time series clustering is an important task because:

- Facilitates the discovery of patterns present in the data - generating clusters for a certain set of data makes it easier to understand the structure of the data, allows you to more easily spot anomalies or other types of patterns present in the time series,
- Makes it easier to review large amounts of data - datasets containing time series can be huge (especially nowadays), making it difficult for an analyst to review such a large amount of data,
- Clustering is the most commonly used exploratory technique - it is part of the more complex tasks of:
  - Discovering rules (patterns) in time series,
  - Classification of time series,
  - Detection of anomalies in time series.

The task

Clustering of bank customers

Customer clustering (dividing customers into groups) is important for various industries and institutions. It is used, among others, in banking, where customer clustering improves, for example marketing campaigns or risk management.



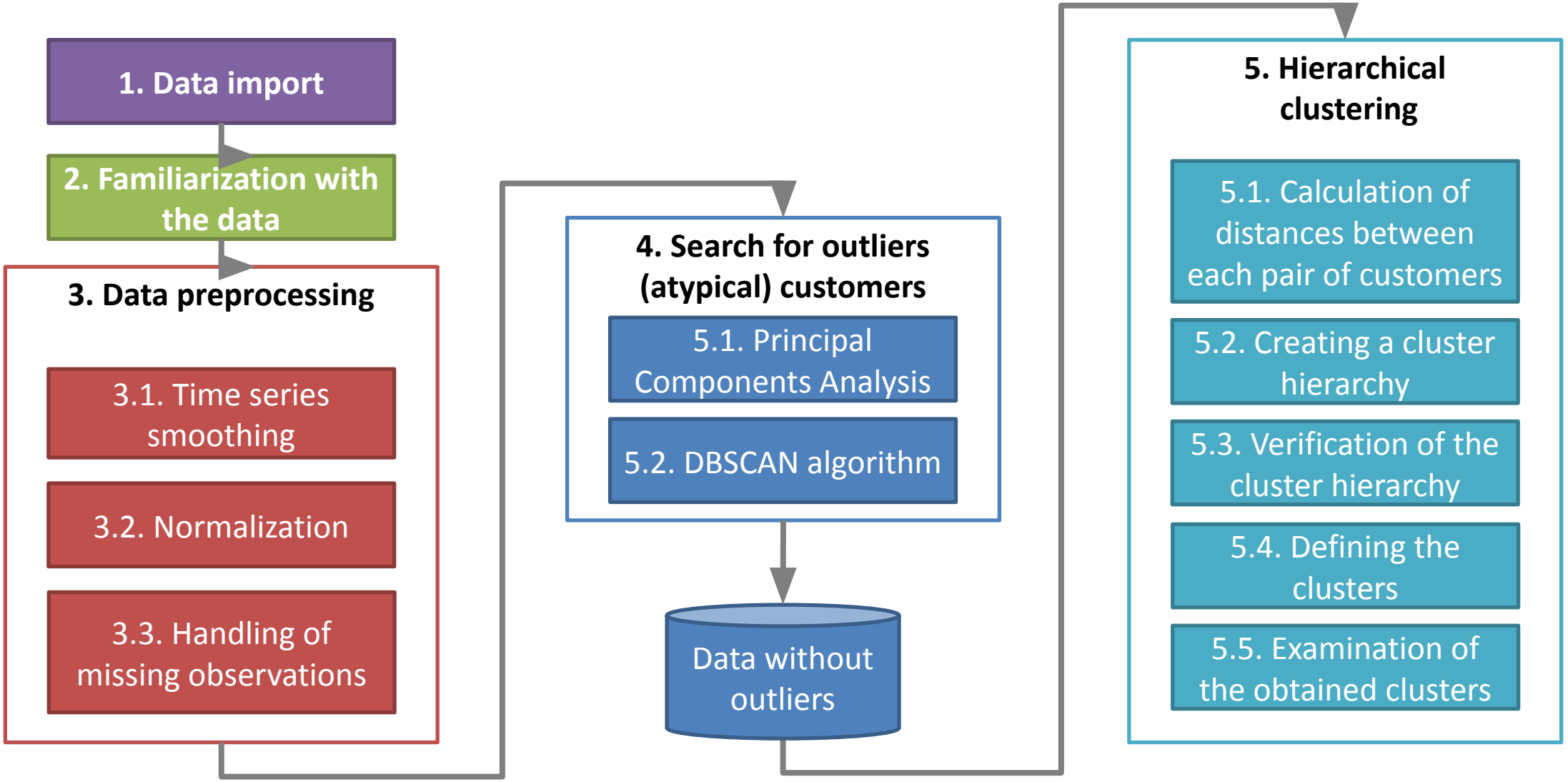
Each customer is described by **48** numerical values - the balance of the bank account (balance) at the end of the month.

The data covers **4** years.

We have **5869** customers in the dataset, **or 5869 time series**.

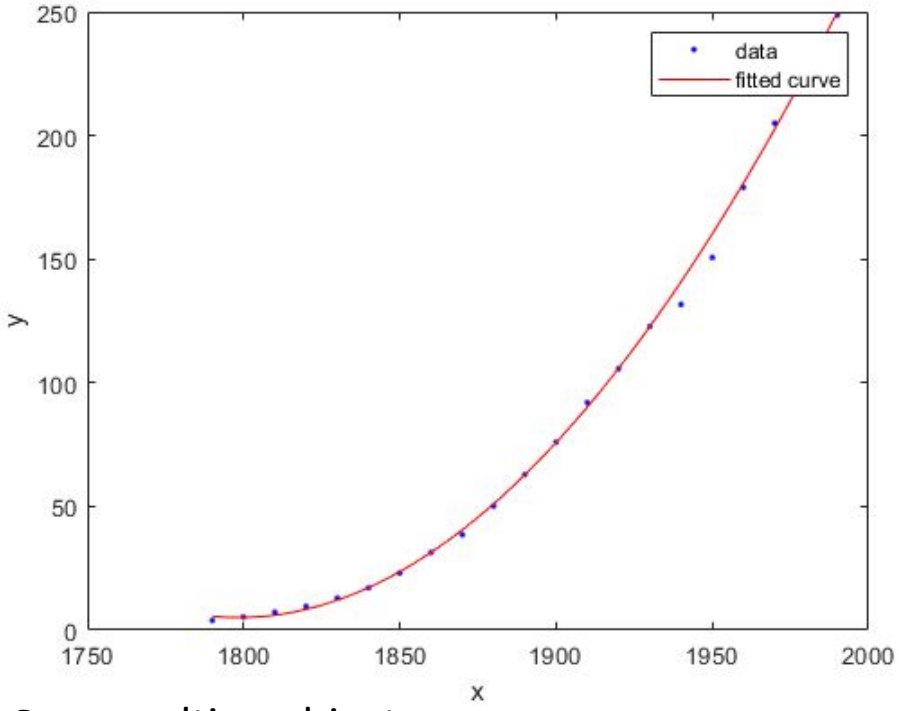
We are looking for groups consisting of similar customers (similar in terms of their account balance in consecutive months).

Course of the task



Fit function

Fits the curve to the data.




```
[f,goodness,output] =
      fit((1:w)', customersclustering(:,1),
'smoothingspline');
```

`((1:w)'` – a vector with numbers from 1 to 48 specifying the data to be matched

`customersclustering(:,1)` – A vector containing customer account balance data, to which the curve should be adjusted

`'smoothingspline'` – The type of model used for curve fitting

`f` – resulting object

`goodness` – parameters showing the goodness of fit of the curve to the data, including: sum of squares of errors (SSE), R2, root mean square error (RMSE)

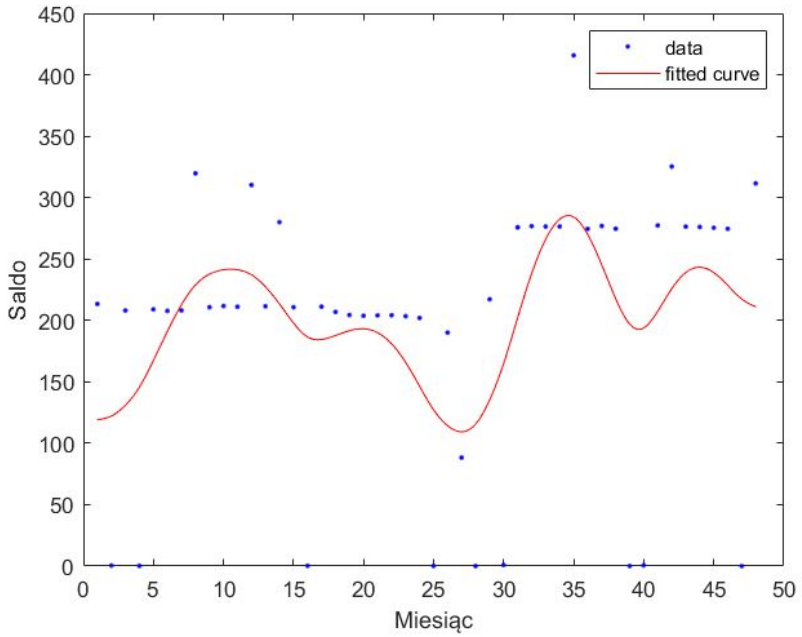
`output` – Information generated by the matching algorithm, including: number of observations (numobs), vector of residuals (residuals), parameter controlling the degree of smoothing (p)

Fit function

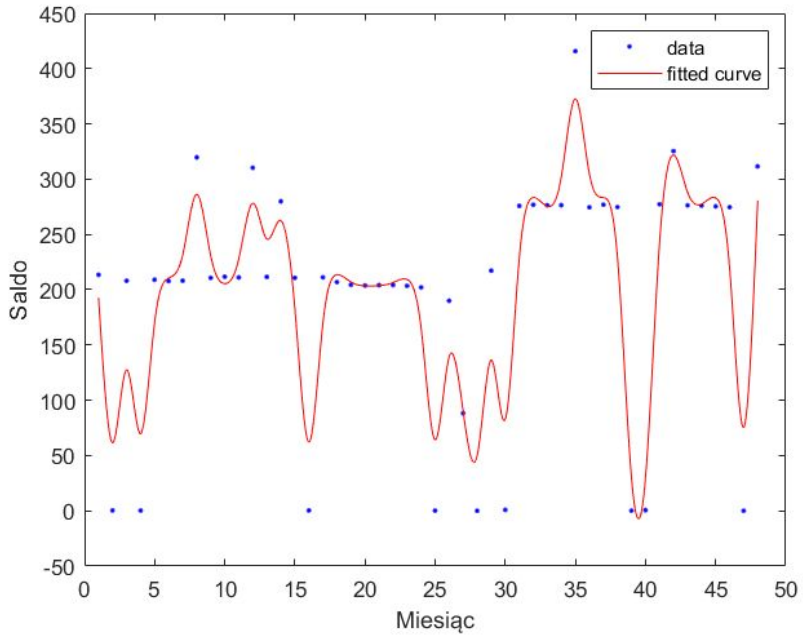
Fits the curve to the data.

To change the degree of curve fitting to the data, add the argument `'SmoothingParam'` with number (0, 1)

```
[f,goodness,output] =
 fit((1:w)', customersclustering(:,1), 'smoothingspline', 'SmoothingParam', 0.07);
```



`'SmoothingParam', 0.07`

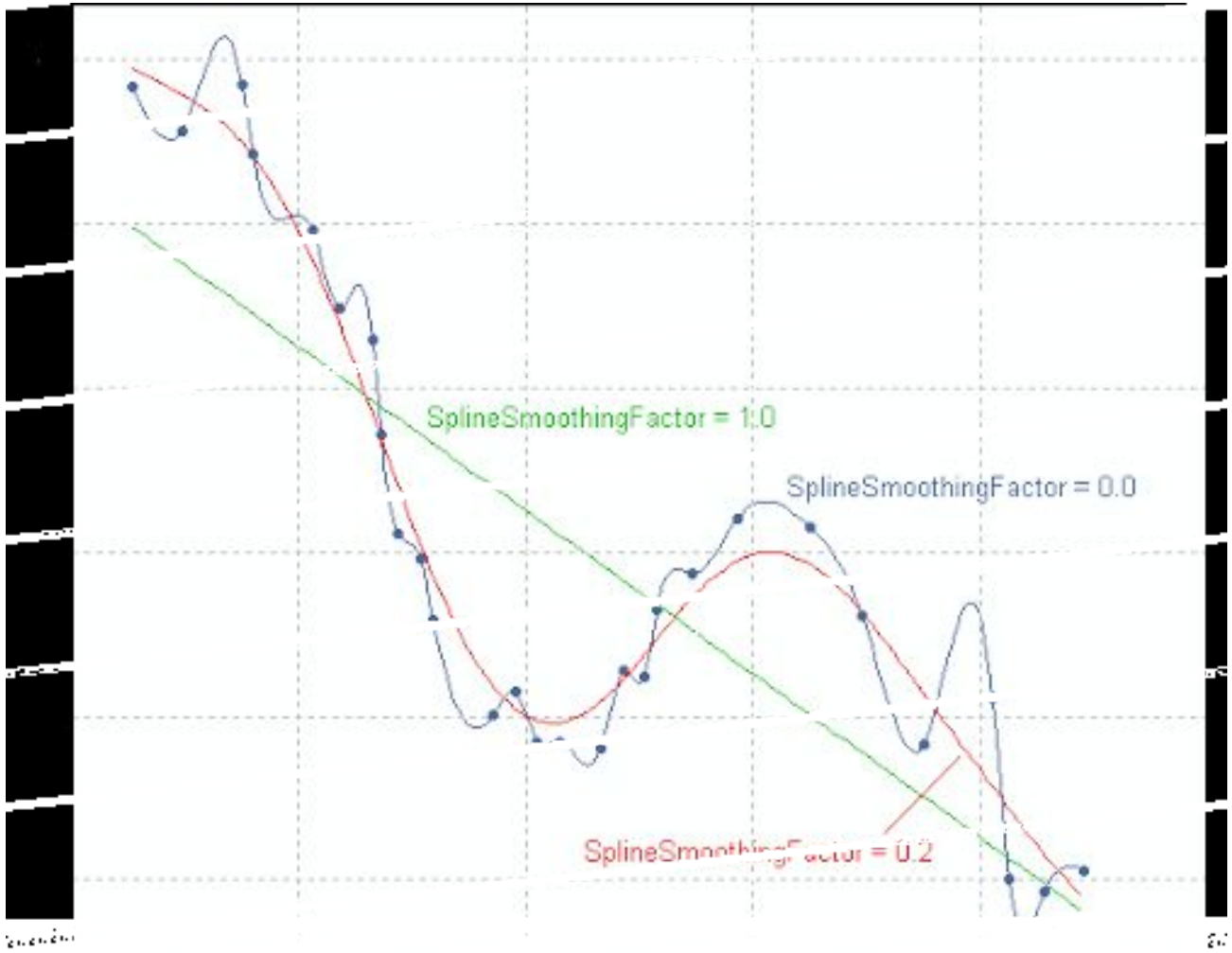


`'SmoothingParam', 0.95`

Spline

Functions used for interpolation, approximation and smoothing of curves or surfaces.
The domain of the function under consideration is divided into parts, and in each part a suitable glued function is used, such as a polynomial of as low a degree as possible, so that predetermined conditions are met, especially continuity and differentiability in the entire domain.

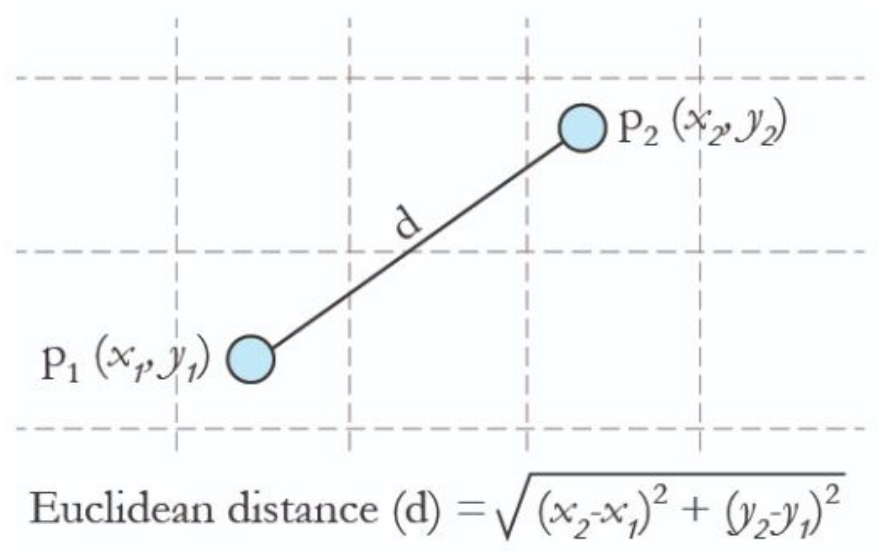
p must be in the range between 0 and 1.
When p = 0, then we fit a straight line to the data using the least squares method.
When p = 1, we create a cubic spline interpolant.
If we do not specify a smoothing parameter, it will be automatically chosen in the range close to 1/(1+h3/6), where h is the average distance between data points.



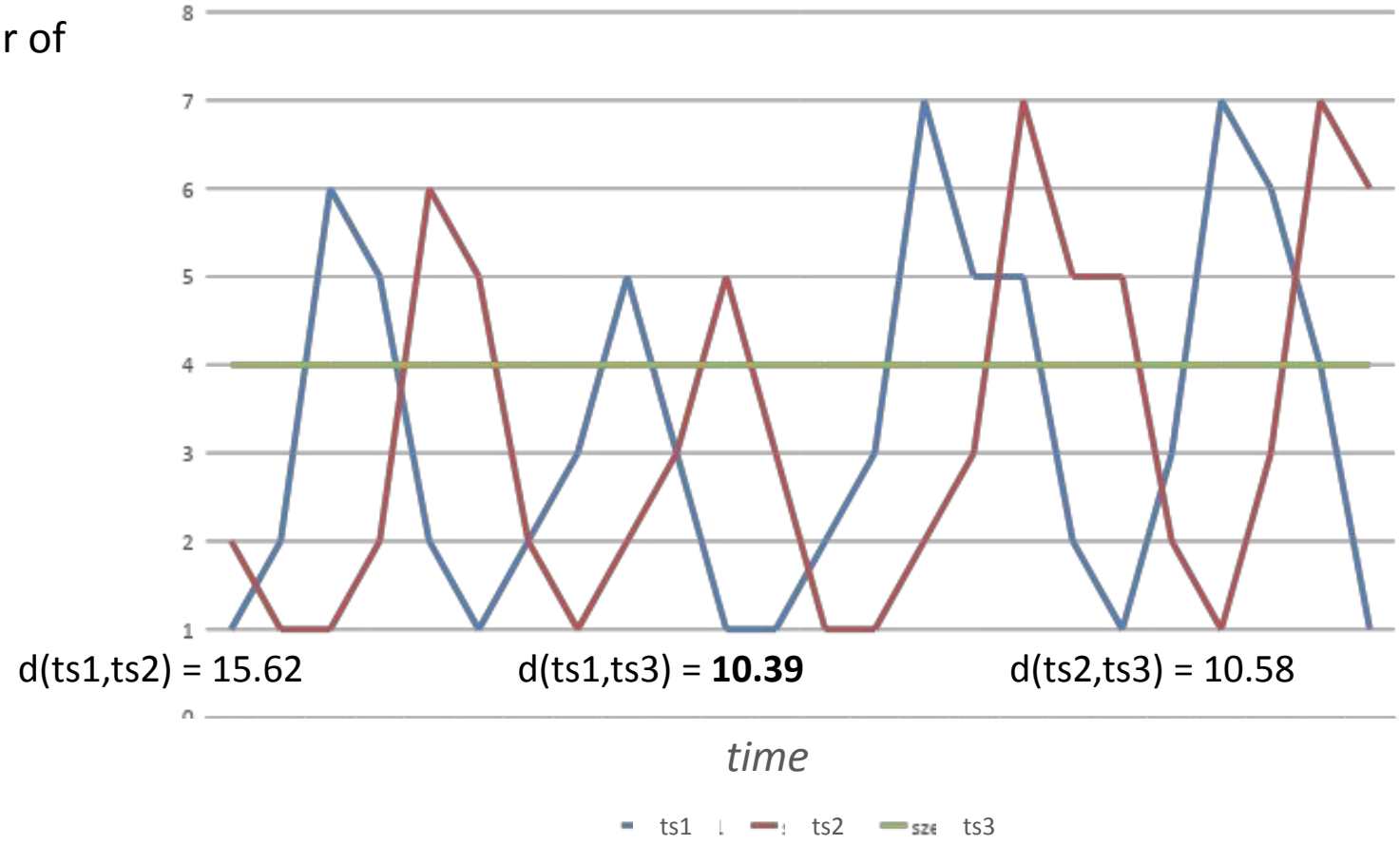
Measures of (dis)similarity

Distance measures are used to express how dissimilar the time series are to each other.

- We calculate the distances between each pair of observations.
- Euclidean distance is most commonly used.



However, in the case of time series, the use of Euclidean distances can lead to some paradoxes, as in the following situation.

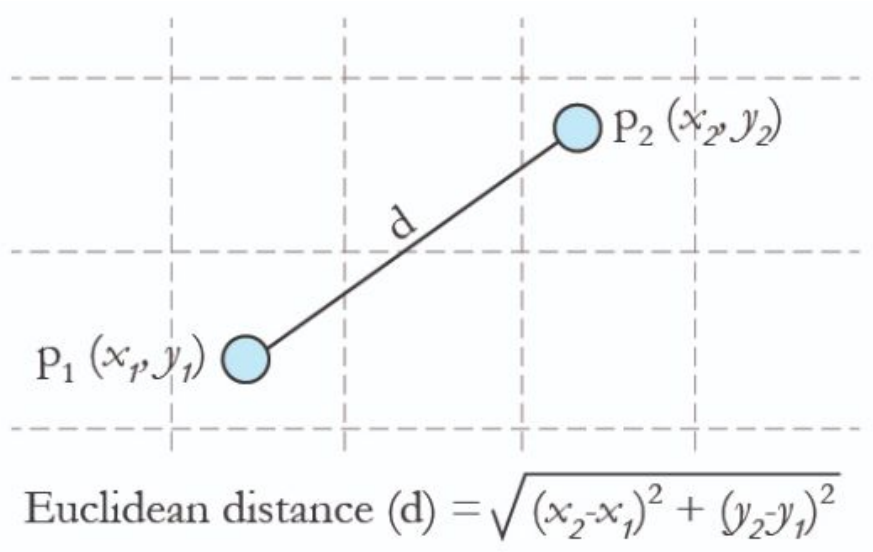


d(ts1,ts2) = 15.62 d(ts1,ts3) = **10.39** d(ts2,ts3) = 10.58

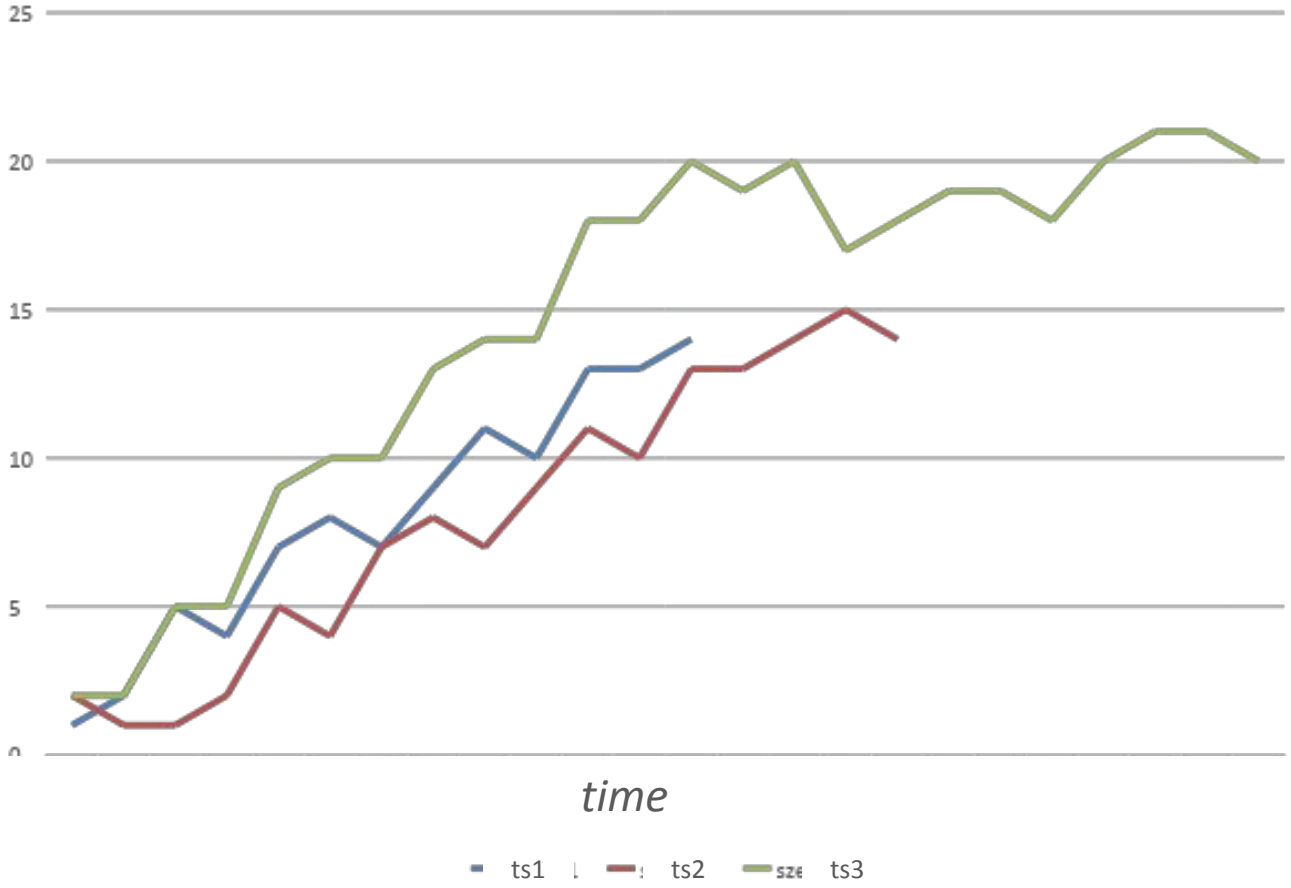
Measures of (dis)similarity

Distance measures are used to express how dissimilar the time series are to each other.

- We calculate the distances between each pair of observations.
- Euclidean distance is most commonly used.



The problem also arises when we want to apply the Euclidean distance for time series of different lengths.

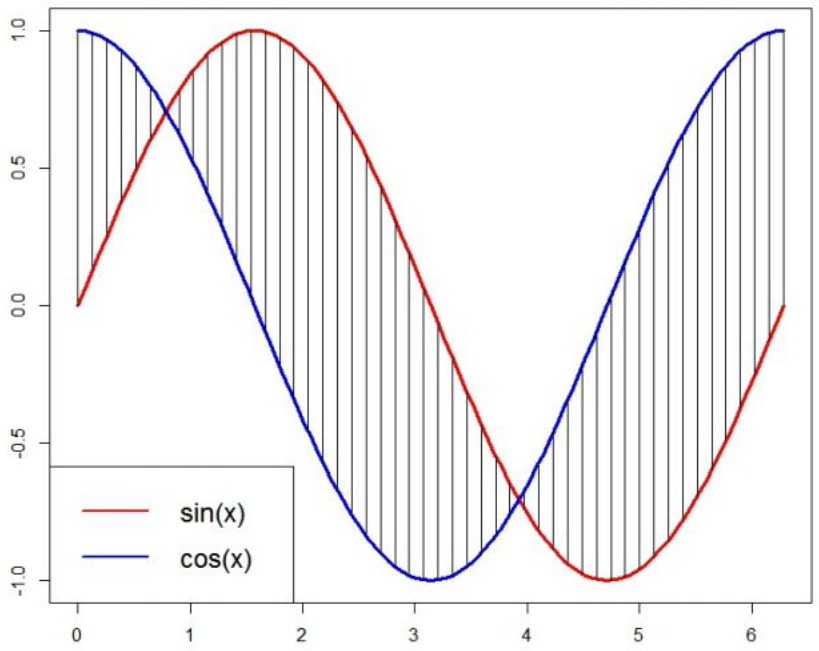


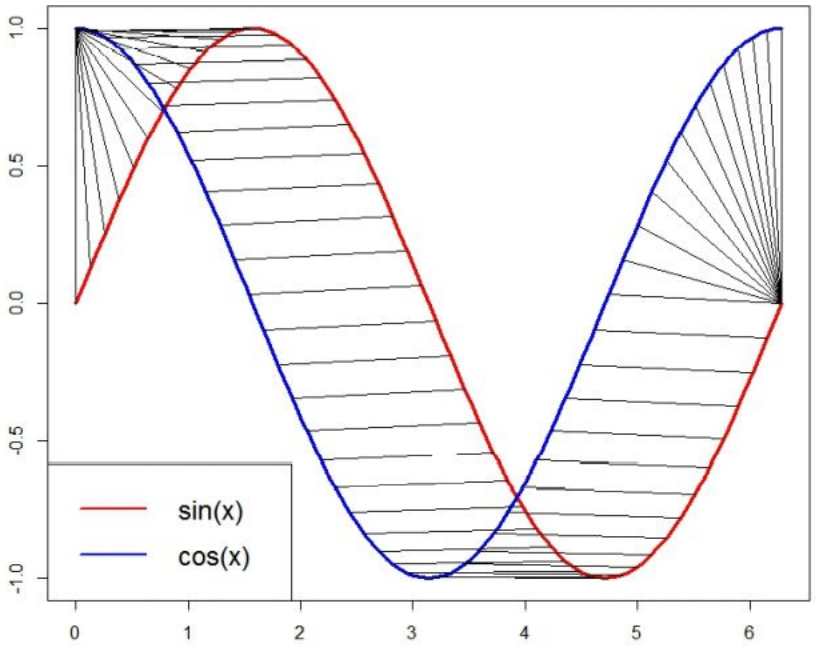
DTW measure

The problems mentioned in the previous two slides are handled by a measure determined by the Dynamic Time Warping (DTW) algorithm.

- The DTW algorithm can determine the optimal alignment of two time series.
- DTW-based measure is more suitable than Euclidean measure especially when comparing time series with similar structure but shifted in time (the following example).

Euclidean distance

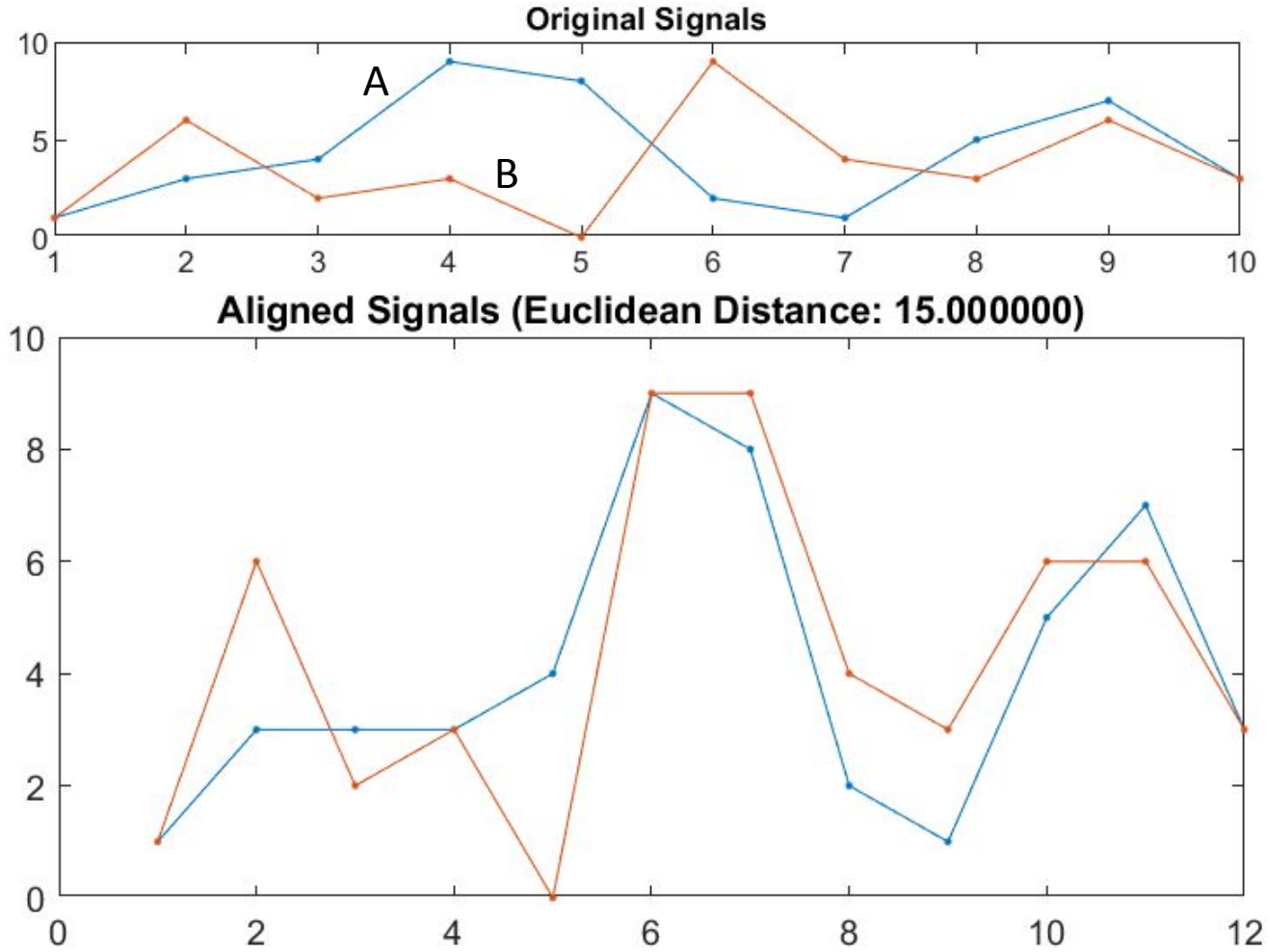




DTW

DTW measure

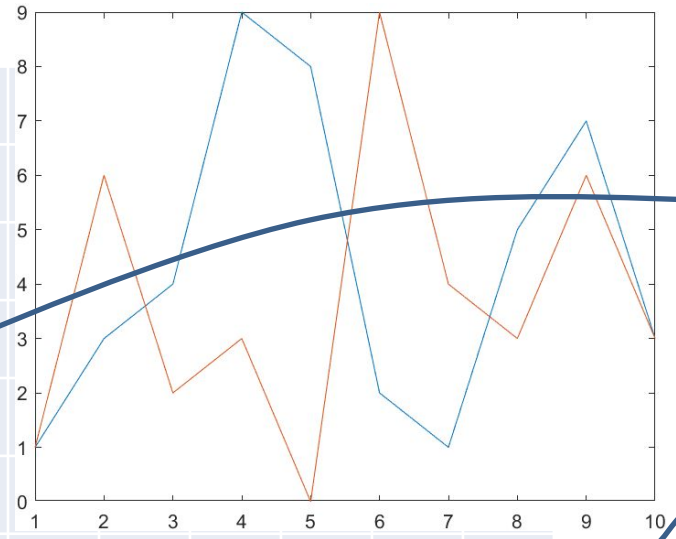
Determination of optimal alignment and DTW measure step by step:



DTW measure

Determination of optimal alignment and DTW measure step by step:

1. Create a distance matrix.



A										
3	33									
7	31									
5	25									
1	21									
2	21									
8	20									
9	13	7	11	11						
4	5	4	5	5	8	12	12	13	15	16
3	2	3	4	4	7	13	14	14	17	17
1	0	5	6	8	9	17	20	22	27	39
	1	6	2	3	0	9	4	3	6	3

B

= |Ai – Bj| + D[i-1, 0]
= |8 – 1| + 13
= 20

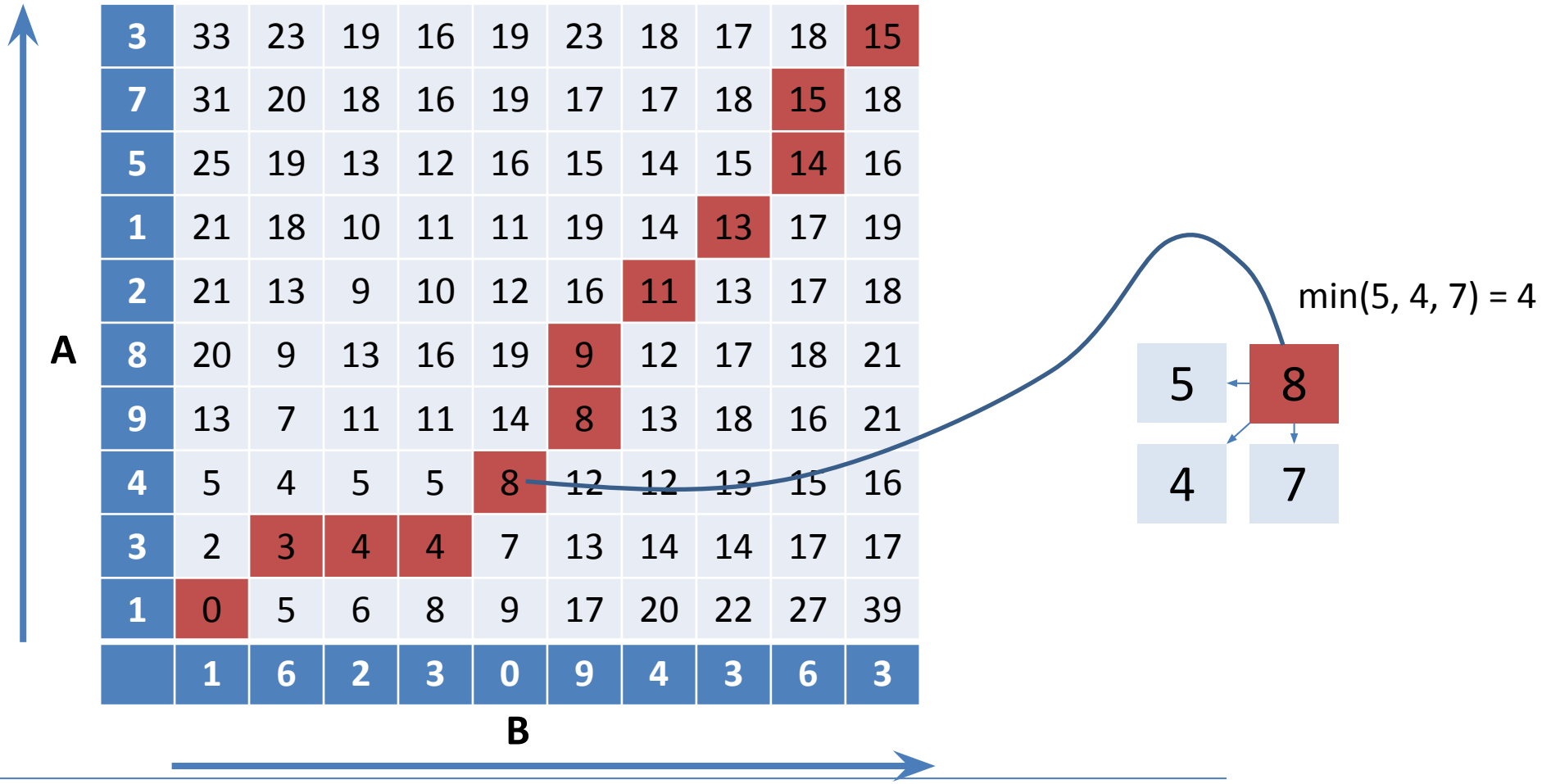
= |Ai – Bj| + min (
D[i-1, j-1],
D[i-1, j]
D[i, j-1])
= |9 – 3| + min(5, 5, 11)
= 6 +5 = 11

= |Ai – Bj| + D[0, j-1]
= |1 – 0| + 8
= 9

DTW measure

Determination of optimal alignment and DTW measure step by step :

1. Create a distance matrix.
2. Determine the line mapping the best fit of the series.



Principal Components Analysis

The main purpose of Principal Component Analysis (PCA) is to simplify the structure of input data. PCA transforms the original input variables into new variables, called principal components. The new variables are not actual observable variables, but are instead a linear combination of the input variables.

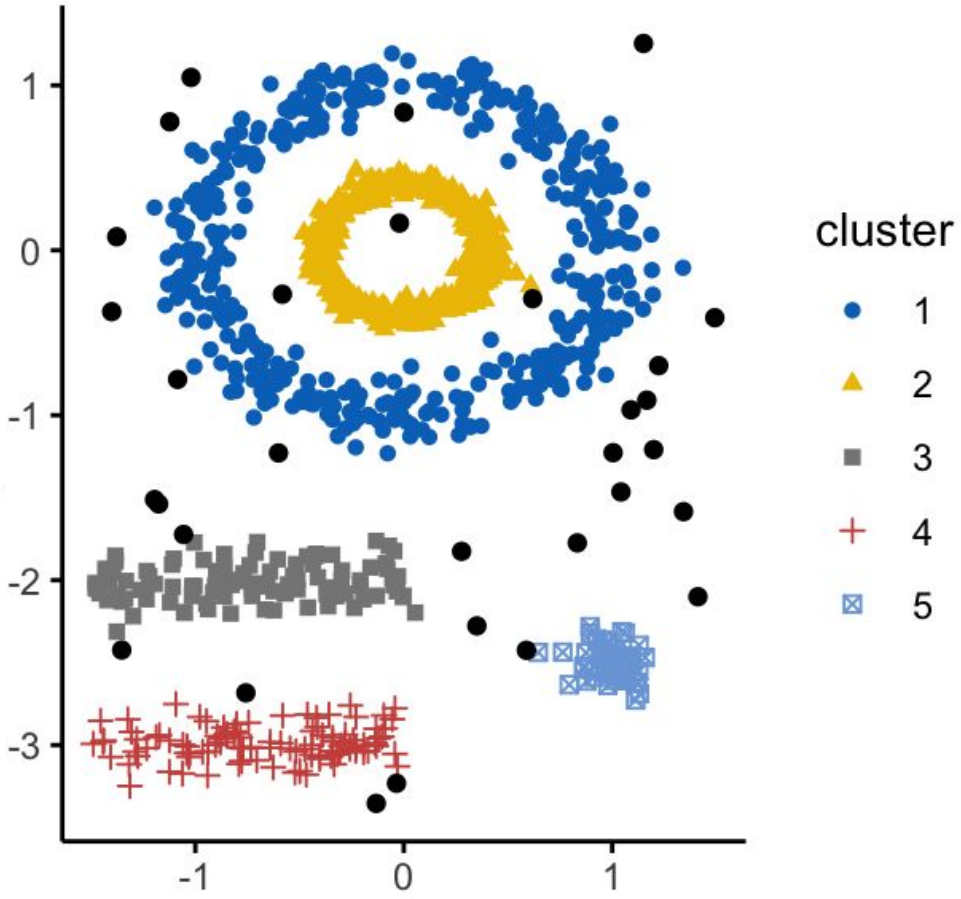
- The result of using PCA is to obtain as many components as the number of input variables analyzed.
- The main components are ordered in descending order of their variance.
- The variance is a measure of the information resources of the input dataset as reflected by a given component.
- The sum of the variances of all the components is always equal to the sum of the variances of all the original input variables. Therefore, it can be said that the use of PCA does not result in the loss of information stored in the dataset, but is able to record it in a different, orthogonal way.

An important result of PCA is the factor coordinates, i.e. the coordinates of objects from the input dataset in the space determined by the components. With these, it is possible to perform a projection of all objects onto the plane created for any two components. We will then subject the projection of objects onto the two principal components to clustering.

DBSCAN

The principle by which DBSCAN (Density-Based Spatial Clustering of Applications with Noise) works defines a cluster as a densely packed area filled with similar objects. Since the clustering result is influenced by the density of observations in space, the algorithm is included in the family of density-based algorithms.

The result of the algorithm depends on its two basic parameters:
the maximum radius of the neighborhood (denoted by Epsilon),
the minimum number of objects in the region defined by Epsilon (MinPts).
The values of these parameters can be chosen experimentally, performing several clustering and selecting those parameters that gave clusters of the best quality (the most homogeneous clusters and at the same time the best separated from each other).

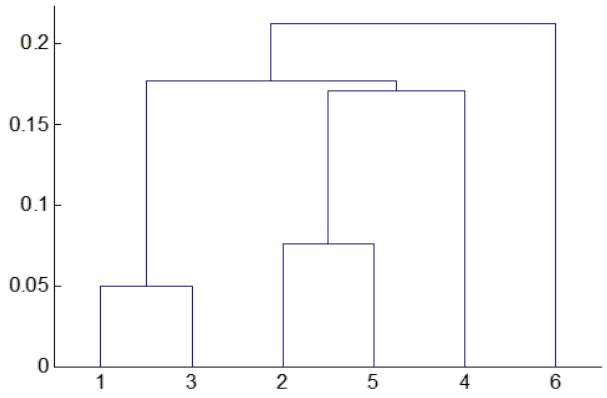


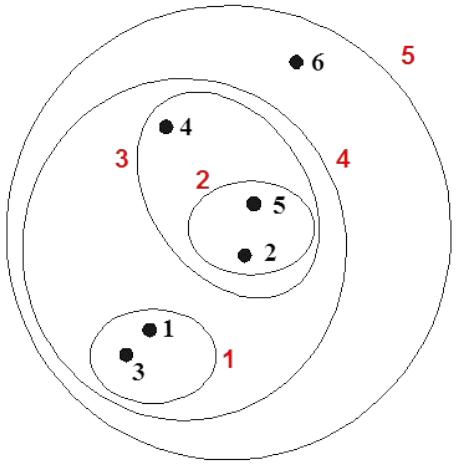
Hierarchical clustering

Hierarchical clustering methods allow:

- **divide objects into clusters consisting of similar objects,**
- **create a structure of clusters mapped in the form of a hierarchy of clusters (the so-called tree diagram, or icicle diagram).**

The advantage of this type of methods, is that it is not necessary to determine the number of clusters to be created before starting clustering.





Hierarchical clustering methods are divided into **agglomerative and deglomerative**. In agglomerative methods, each object is initially a one-element cluster. In subsequent iterations of the algorithm, the two most similar clusters are combined with each other to form a new cluster, which is treated as superior in the cluster hierarchy. The final step of the algorithm is to create a cluster consisting of all objects.

The deglomeration method uses the reverse approach, starting with a single cluster combining all objects, which in subsequent steps of the algorithm is divided into smaller and smaller clusters.

Hierarchical clustering

The similarity of objects at the first step of the agglomeration approach is calculated using the adopted **distance measure**. In the next steps of the algorithm, when more clusters are created, the corresponding **binding (agglomeration) rule** is applied. It determines when two clusters are similar enough to be combined into a single higher-level cluster. The main methods distinguished here are:

- Ward – uses a variance analysis approach to estimate the distance between clusters (the method aims to minimize the sum of squares of cluster deviations) - a method generally regarded as very effective,
- single linkage (nearest neighbor) - the distance between two clusters is defined as the shortest distance between two objects belonging to two different clusters (objects form clusters by joining in strings, and the resulting clusters form long "chains"),
- complete linkage (farthest neighbor) - the distance between two clusters is defined as the longest distance between two objects belonging to two different clusters (we usually use when objects form naturally separated "clumps", while this method is not suitable if the clusters are elongated),
- average linkage - the distance between two clusters is defined as the average distance between all pairs of objects belonging to two different clusters (suitable for clusters in the form of "chains" as well as "clumps"),
- weighted average linkage - as above, but in addition, we take into account the weight relating to the size of the clusters,
- centroid linkage - the distance between two clusters is defined as the distance between the centers of gravity of two different clusters,
- weighted centroid linkage (medians) - as above, but additionally we take into account the weight relating to the size of the clusters.

Thank You for your attention!