CS-541
Wireless Sensor Networks

Lecture 10: Time-series analysis

Spring Semester 2017-2018

Prof Panagiotis Tsakalides, Dr Athanasia Panousopoulou, Dr Gregory Tsagkatakis

Overview

- Time series analysis
- Intro to Machine learning

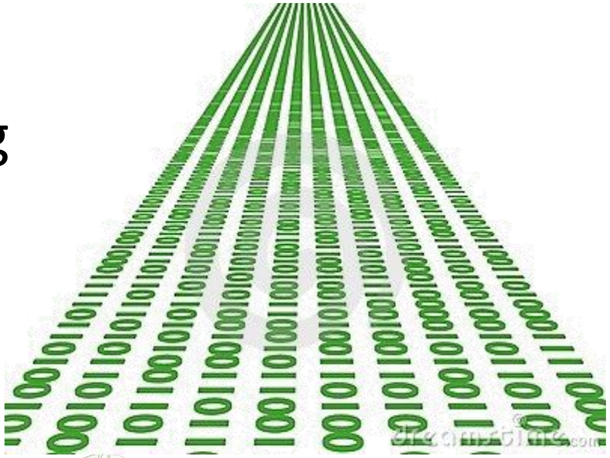
Stream Data Processing

Data streams—continuous, ordered, changing, fast, huge amount

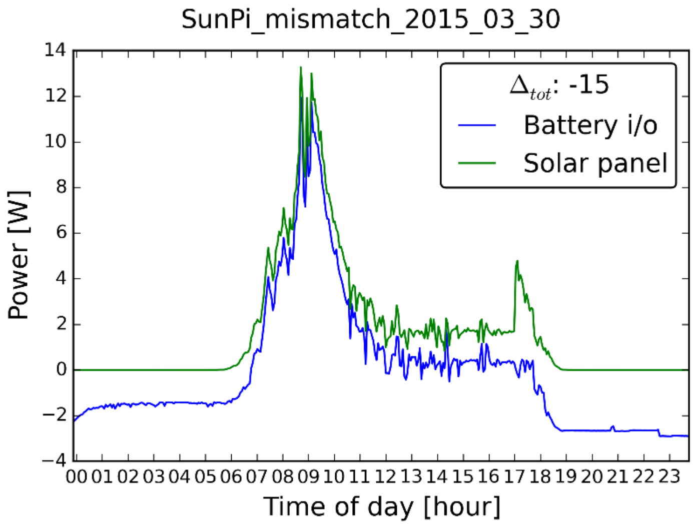
- Huge *volumes* of continuous data, possibly infinite
- Fast *changing* and requires fast, real-time response

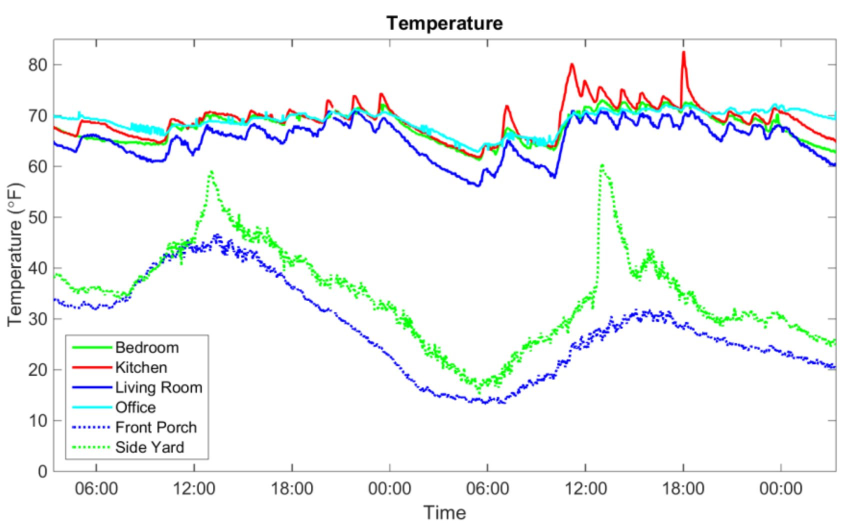
Applications

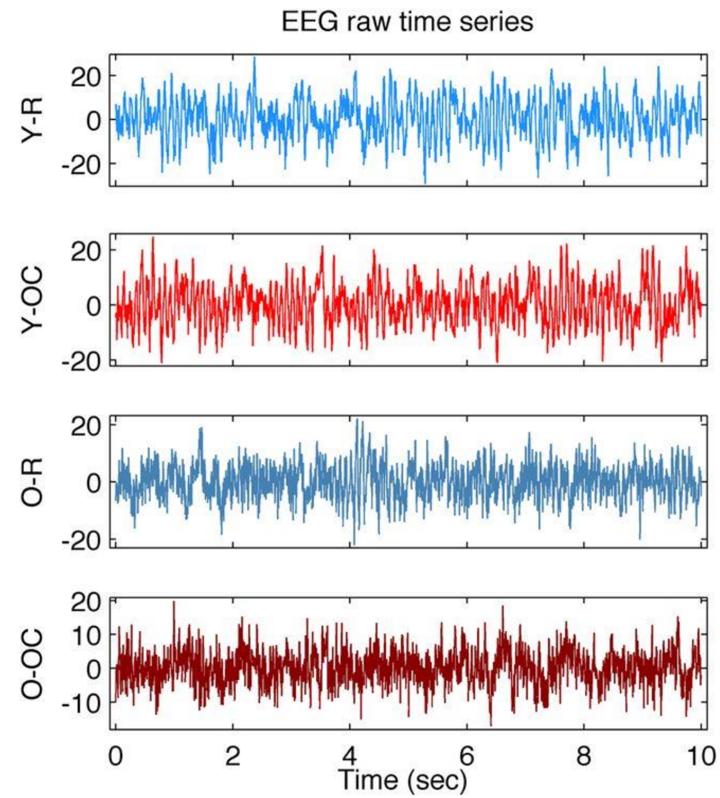
- Telecommunication records
- Network monitoring and traffic engineering
- Industrial processes: power & manufacturing
- Sensor, monitoring & surveillance



Time-series in WSN







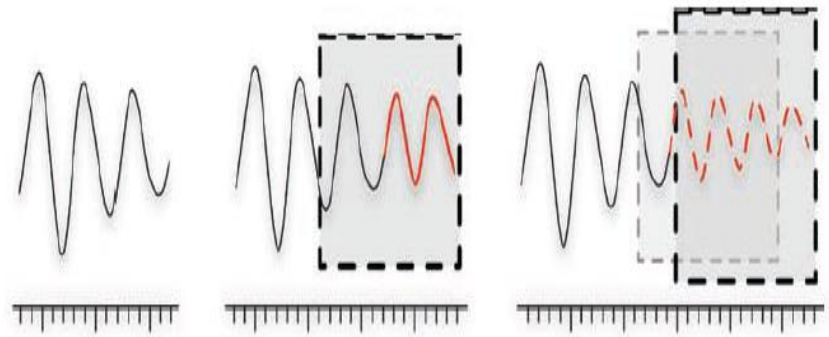
Problems

- *Type 1*: patterns, periodicities, and/or compress
  - Wearable, Smart city
- *Type 2*: forecast, find motifs, quantify similarity
  - Activity recognition
- *Type 3*: Multiple time series analysis
  - Sensor networks

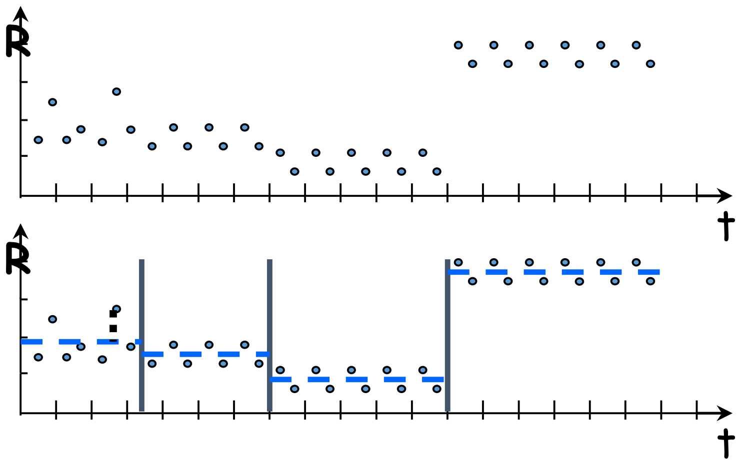
"Predictions are very difficult… especially about the future"
Niels Bohr

Applications

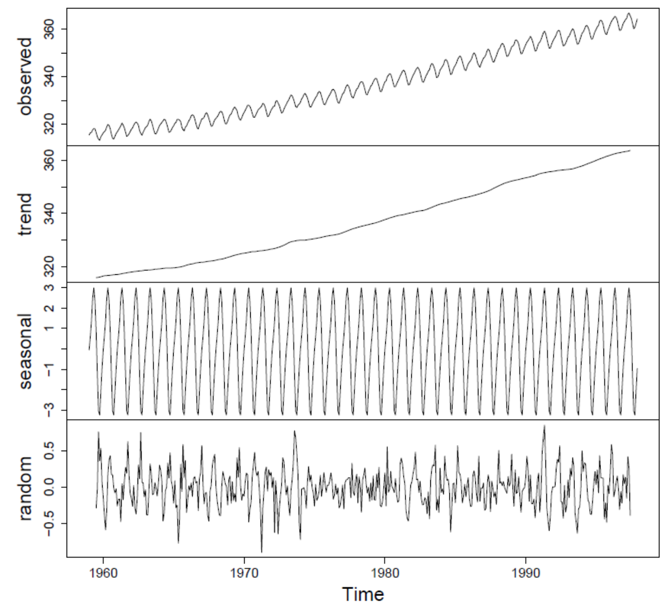
Prediction - Forecasting



Segmentation - Clustering



Analysis



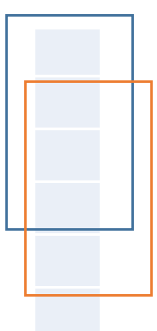
Time-Series data

Time series: sequence of observations $s_t \in R$ ordered in time t=1…N

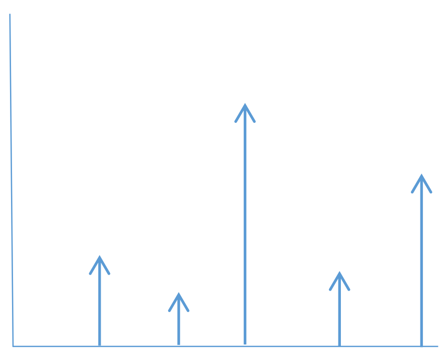
Applications

- Weather, economic, marketing, web, envirometrics, sensor networks

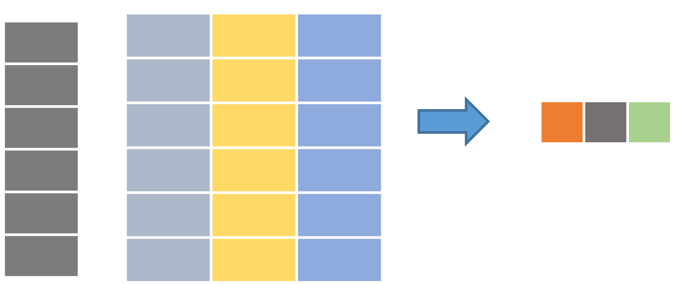
Representations



Sliding windows



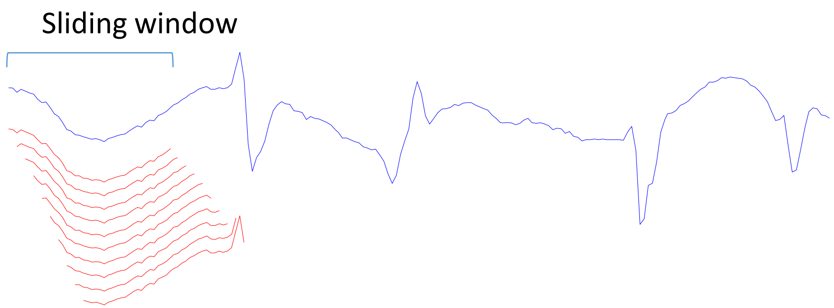
Histograms

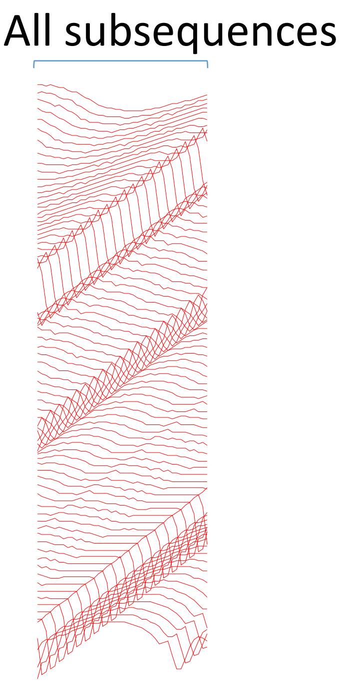


Transform coding

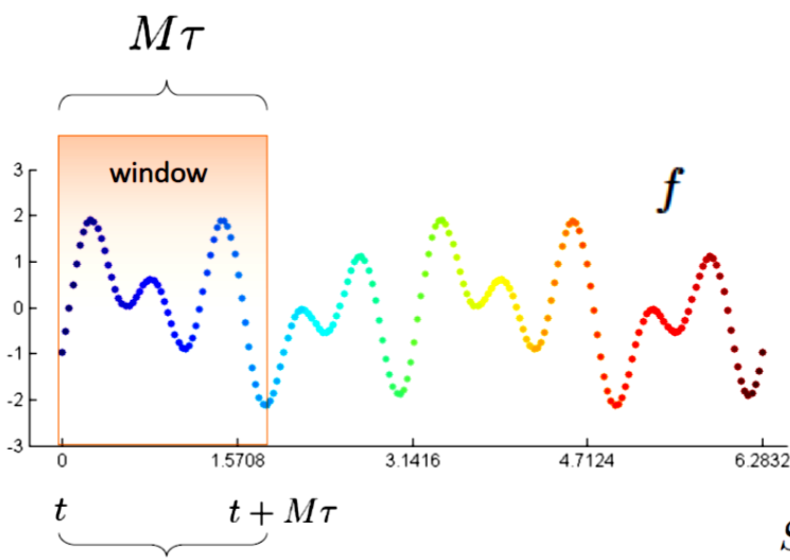
Sliding window

- Given a time series, individual subsequences are extracted with a sliding window





Sliding windows embedding



$f(t), f(t+\tau), \ldots, f(t+M\tau)$

Fix

τ → **Step/delay**

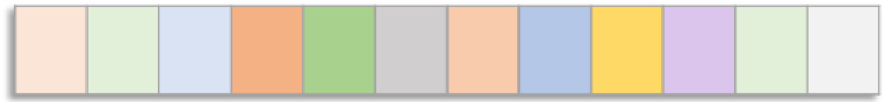
$M+1$ → **Dimension**

$$SW_{M,\tau}f(t) = \begin{bmatrix} f(t) \\ f(t+\tau) \\ \vdots \\ f(t+M\tau) \end{bmatrix} \in \mathbb{R}^{M+1}$$

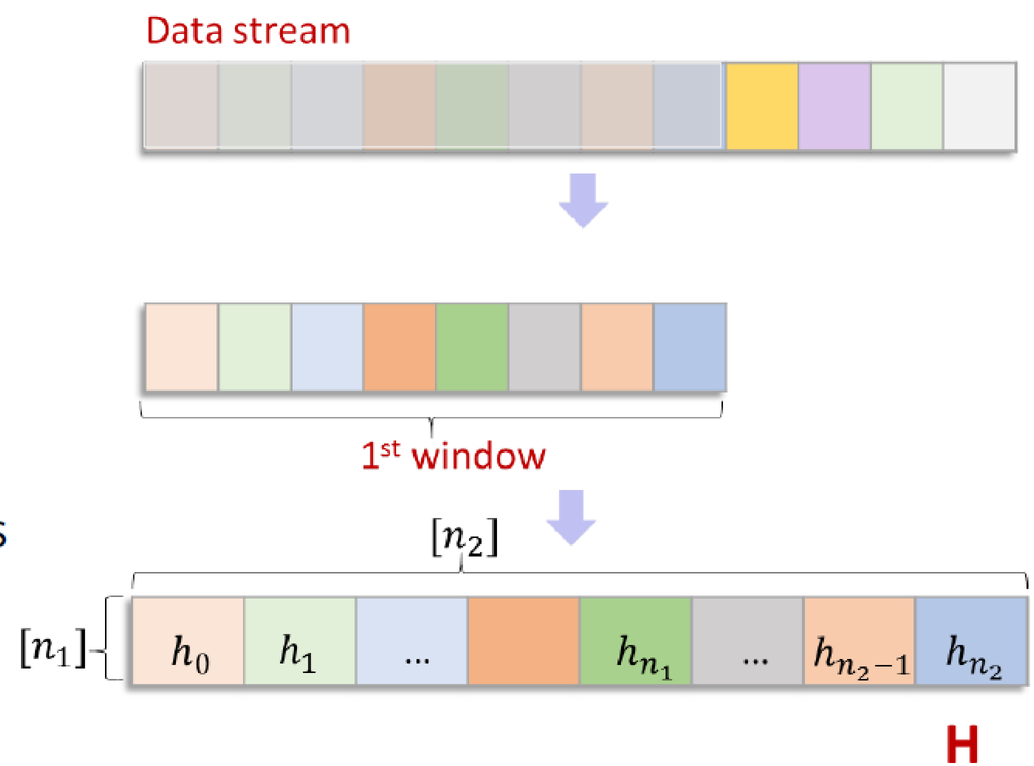
Sliding Windows and Persistence: An application of topology to signal analysis, J. Perea and J. Harer, 2015

1. Sensor stream

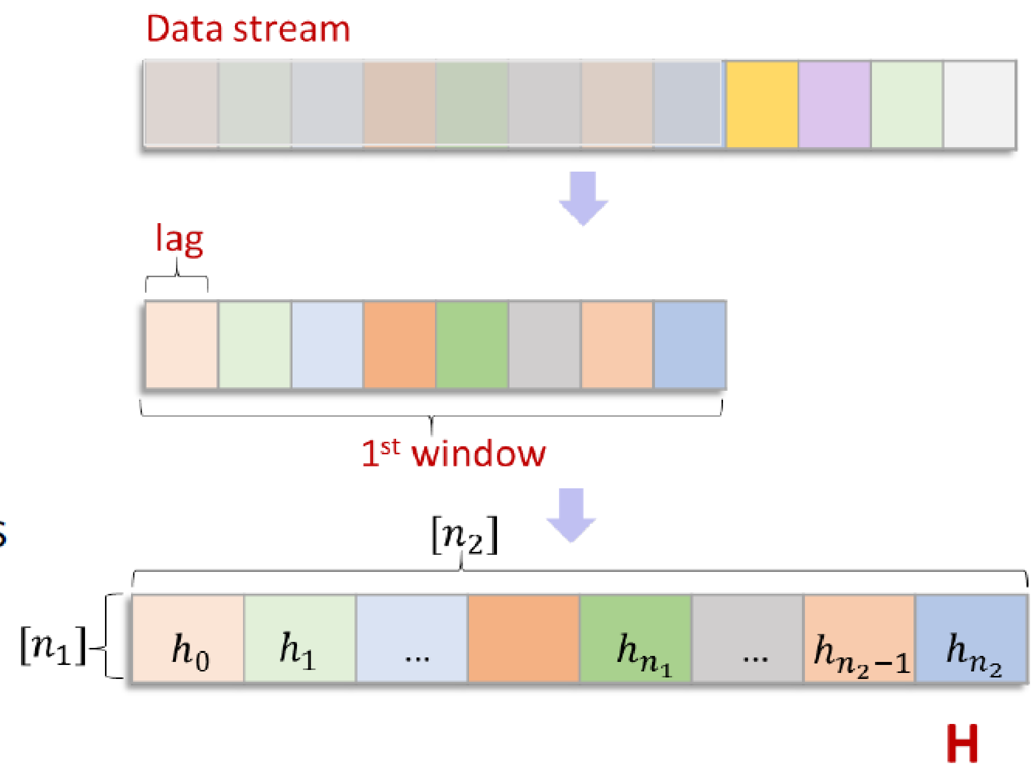
Data stream



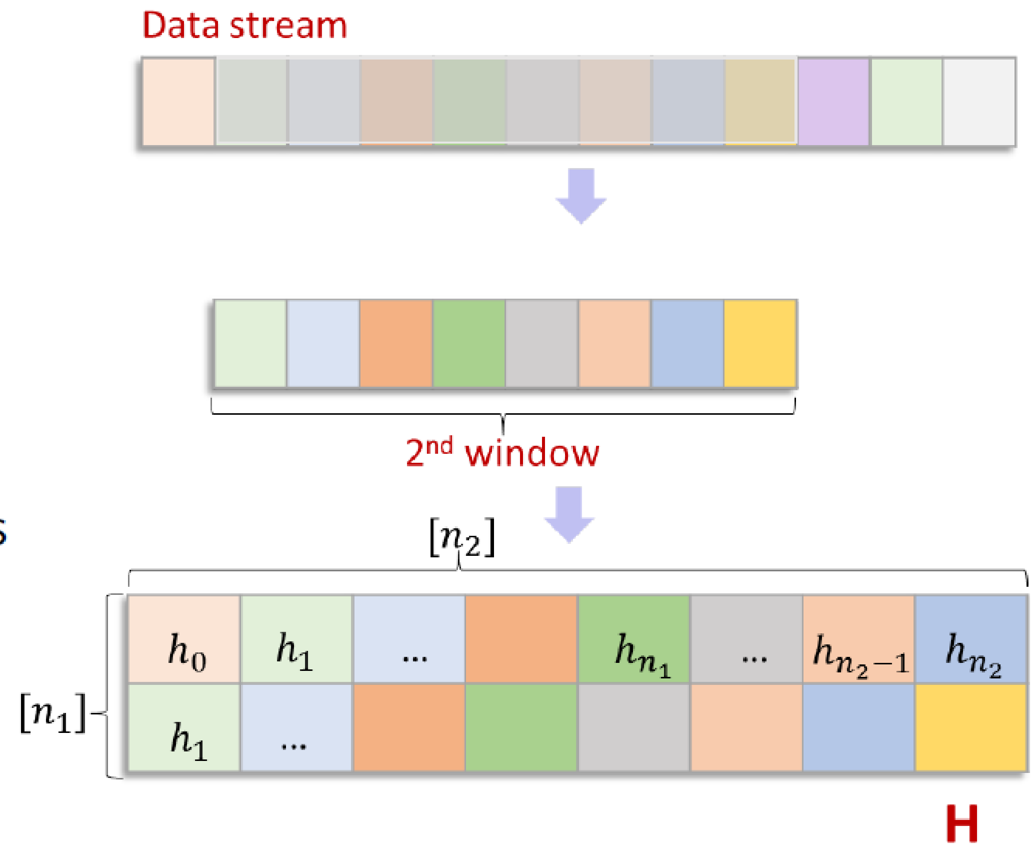
1. Sensor stream
2. Temporal windowing
3. Hankelization process **H**
   - ✔ $[n_1]$ lagged temporal windows of $[n_2]$ samples



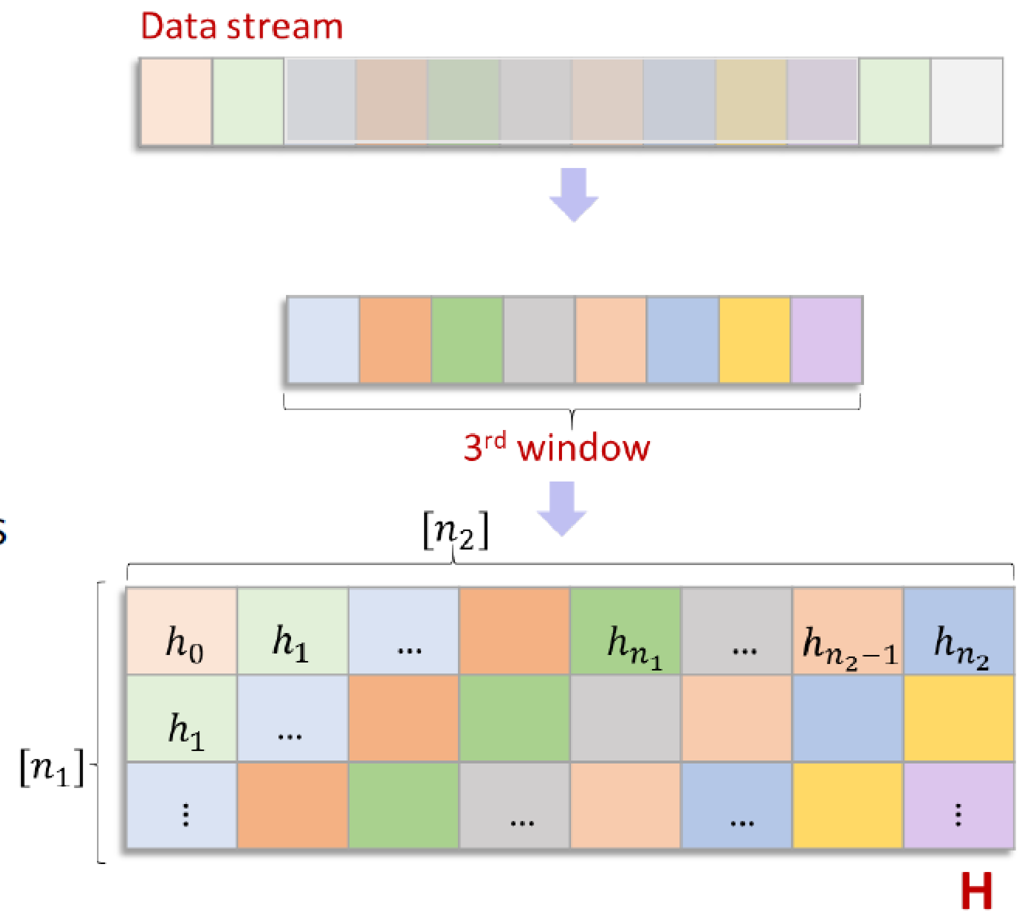
1. Sensor stream
2. Temporal windowing
3. Hankelization process **H**
   - ✔ $[n_1]$ lagged temporal windows of $[n_2]$ samples



1. Sensor stream
2. Temporal windowing
3. Hankelization process **H**
   - ✔ $[n_1]$ lagged temporal windows of $[n_2]$ samples



1. Sensor stream
2. Temporal windowing
3. Hankelization process **H**
   - ✔ $[n_1]$ lagged temporal windows of $[n_2]$ samples

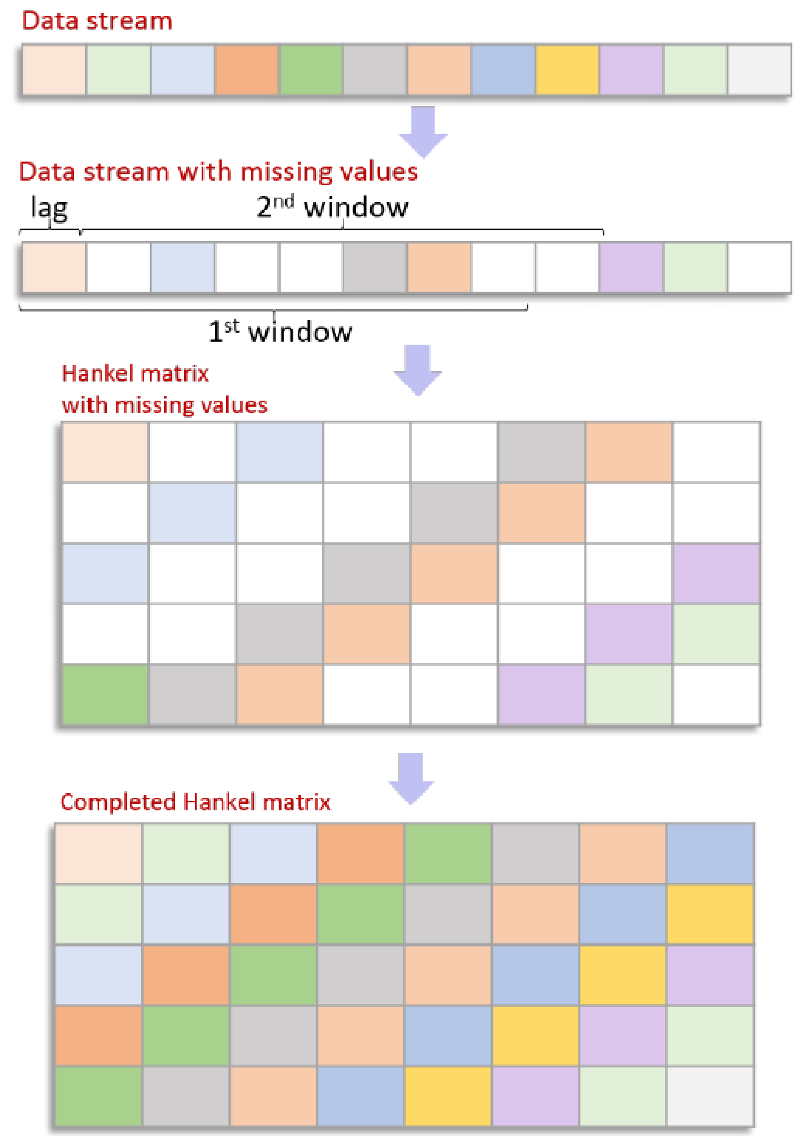


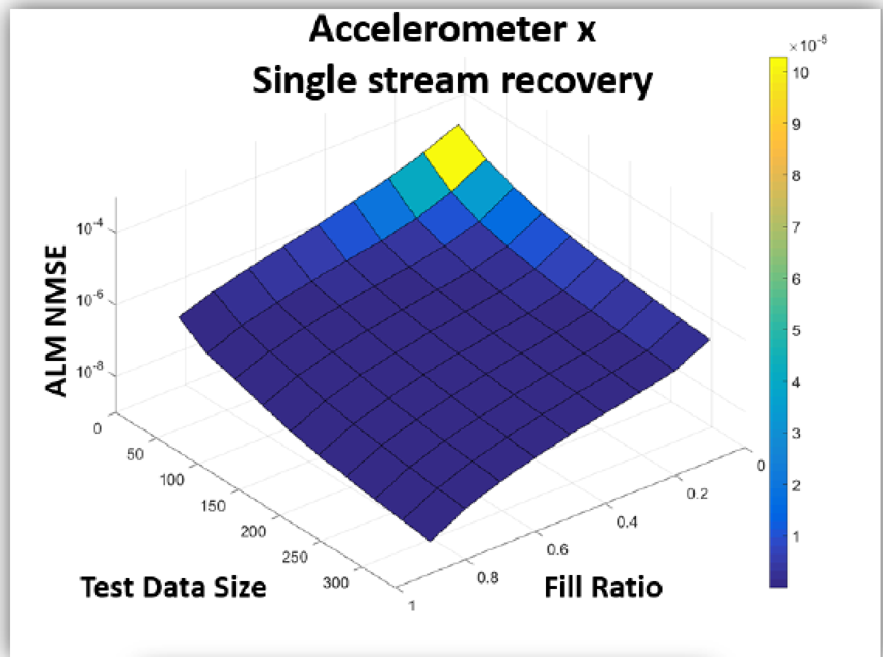
1. **Test** sensor stream
2. Introduction of missing values
3. Temporal windowing
4. Hankelization process **H**

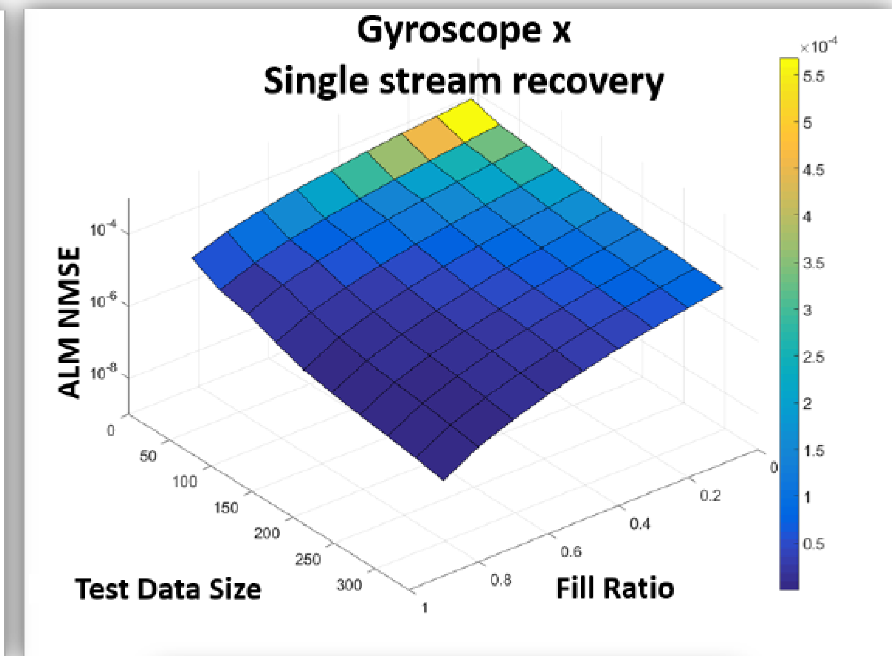
5. Undersampled Hankel matrices that need to be reconstructed!

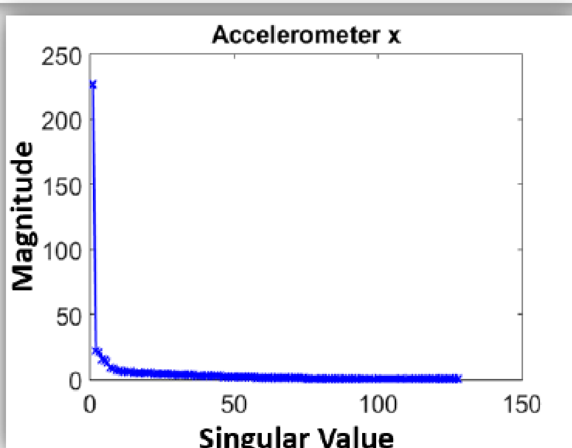
$\Downarrow$

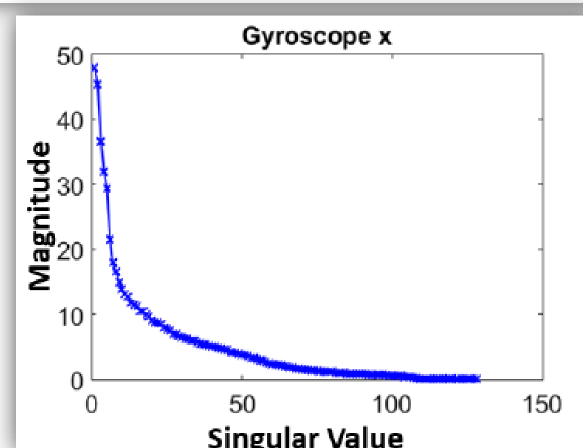
Matrix Completion



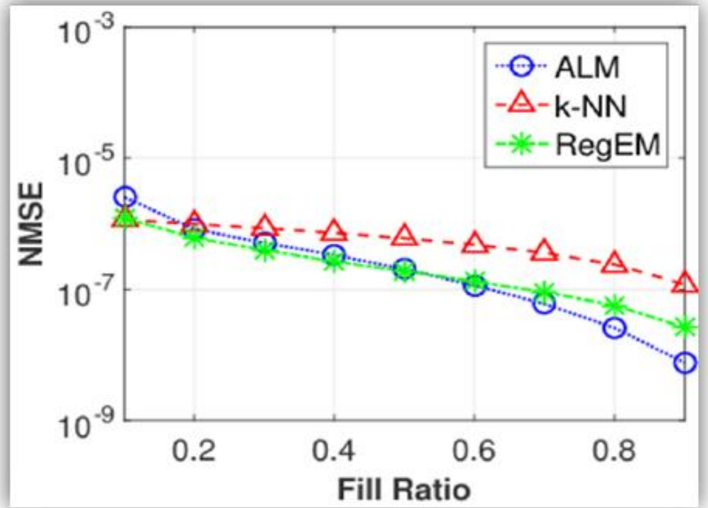

Accelerometer x
Single stream recovery
ALM NMSE
10^{-4}
10^{-6}
10^{-8}
Test Data Size
0
50
100
150
200
250
300
Fill Ratio
0
0.2
0.4
0.6
0.8
1
$\times 10^{-5}$
10
9
8
7
6
5
4
3
2
1

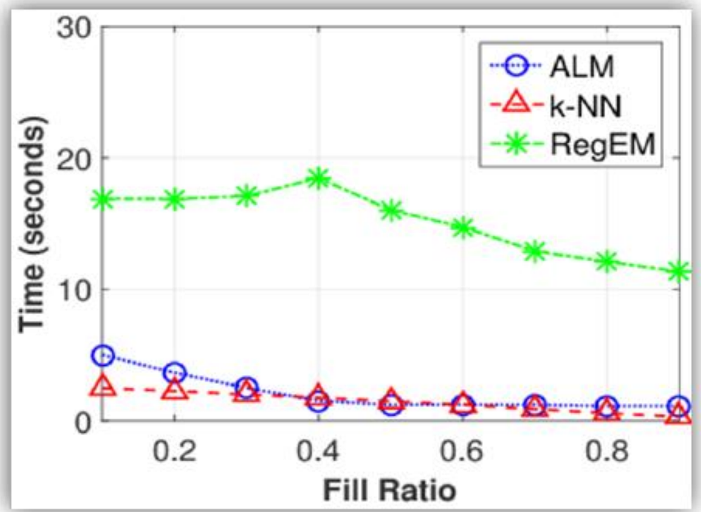

Gyroscope x
Single stream recovery
ALM NMSE
10^{-4}
10^{-6}
10^{-8}
Test Data Size
0
50
100
150
200
250
300
Fill Ratio
0
0.2
0.4
0.6
0.8
1
$\times 10^{-4}$
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5


Accelerometer x
Magnitude
250
200
150
100
50
0
Singular Value
0
50
100
150

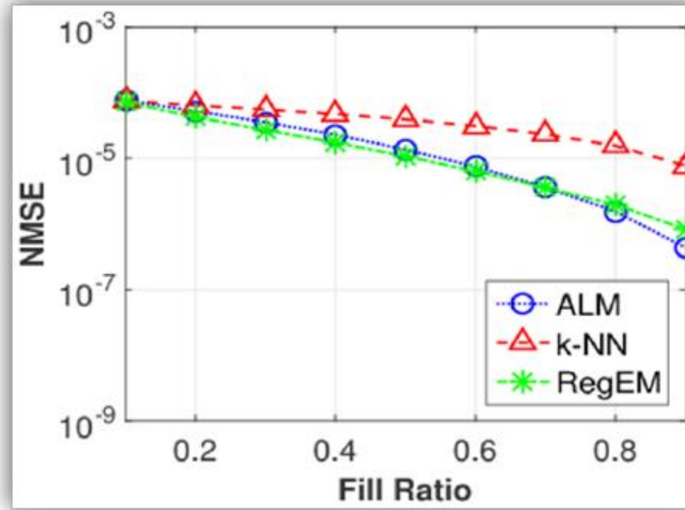

Gyroscope x
Magnitude
50
40
30
20
10
0
Singular Value
0
50
100
150

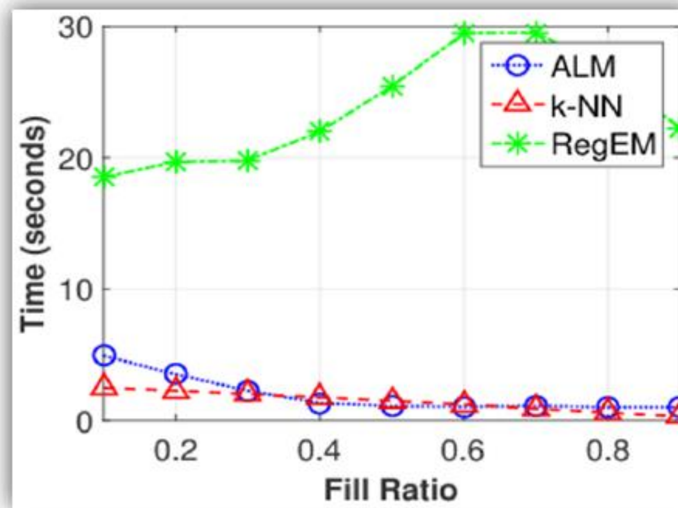
Accelerometer x
Single stream recovery





Gyroscope x
Single stream recovery





Autoregressive Models (AR)

Thus for *stationary* time series the mean value function is **constant** and the covariance function is only a **function of the distance in time** (*t* – *s*)

The "order" of the AR(*p*) models is the number of prior values used in the model.

Univariate AR model

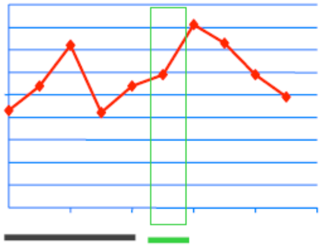
- **AR(1)→** $x_t = b_0 + b_1 x_{t-1} + \varepsilon_t$
- **AR(2)→** $x_t = b_0 + b_1 x_{t-1} + b_2 x_{t-2} + \varepsilon_t$
- **AR(p)→** $X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t.$

Solutions: Yule–Walker equations

Estimation of autocovariances, least squares regression

Matrix formulation

- $\mathbf{X}_{[N \times w]} \times \mathbf{a}_{[w \times 1]} = \mathbf{y}_{[N \times 1]}$



$$
\begin{array}{c} \text{Ind-var1} \quad \cdots \quad \text{Ind-var-w} \end{array}
$$

time

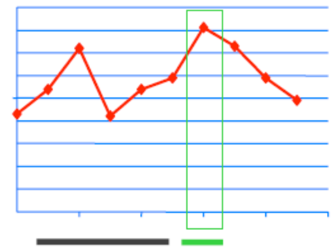
$$
\begin{bmatrix} X_{11}, X_{12}, \cdots, X_{1w} \\ X_{21}, X_{22}, \ldots, X_{2w} \\ \vdots \\ \vdots \\ \vdots \\ X_{N1}, X_{N2}, \ldots, X_{Nw} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_w \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_N \end{bmatrix}
$$

http://www.cs.kumamoto-u.ac.jp/
~yasuko/TALKS/15-SIGMOD-tut/

128

Matrix formulation

- $\mathbf{X}_{[N \times w]} \times \mathbf{a}_{[w \times 1]} = \mathbf{y}_{[N \times 1]}$



Ind-var1 Ind-var-w

$$\text{time} \downarrow \quad \begin{bmatrix} X_{11}, X_{12}, \cdots, X_{1w} \\ X_{21}, X_{22}, \ldots, X_{2w} \\ \vdots \\ \vdots \\ \vdots \\ X_{N1}, X_{N2}, \ldots, X_{Nw} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_w \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_N \end{bmatrix}$$

//www.cs.kumamoto-u.ac.jp/
ko/TALKS/15-SIGMOD-tut/

129

Vector Autoregressive Models (VAR)

Vector AR (VAR) extension to multiple time series

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + e_t,$$

- least squares: $B_1=(Y_{t-1}^T Y_{t-1})^{-1} Y_{t-1}^T Y_t$ (under conditions)
- Determination of lag length is a trade-off

Granger causality: statistical hypothesis test for determining whether one time series X is useful in forecasting another time series Y, ('60)

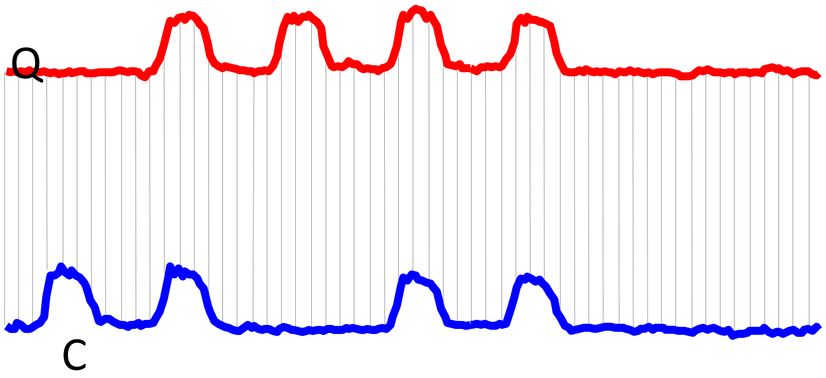
$$Y_t = \alpha + \phi_1 Y_{t-1} + \beta_1 X_{t-1} + e_t$$

"if β_1=0 then past values of X have no explanatory power for Y beyond that provided by past values of Y".

Similarity between time-Series

Euclidean Distance

$$D(\vec{x},\vec{y}) = \sum_{i=1}^{n}(x_i - y_i)^2$$

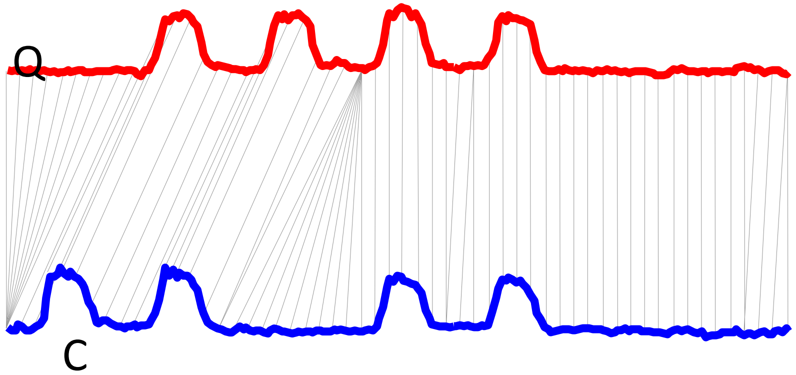


(+) *Efficient computation*

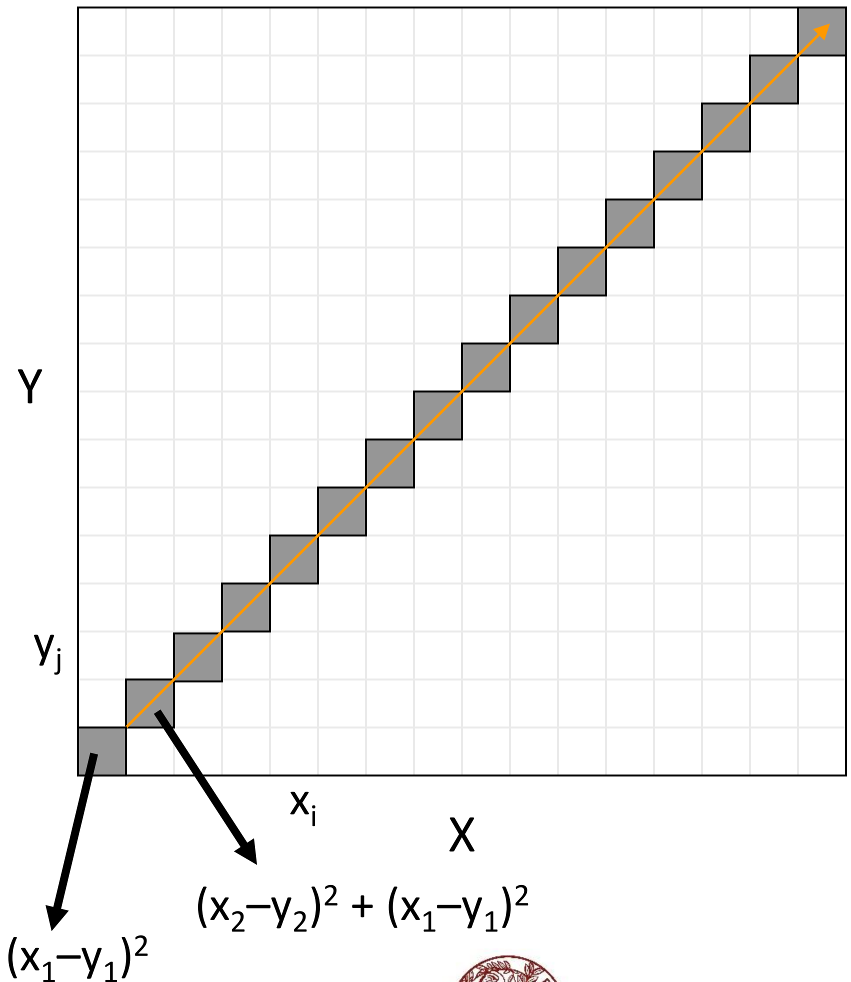
(-) *Time shift, scaling*

Dynamic Time Warping

- *Nonlinear alignments are possible.*

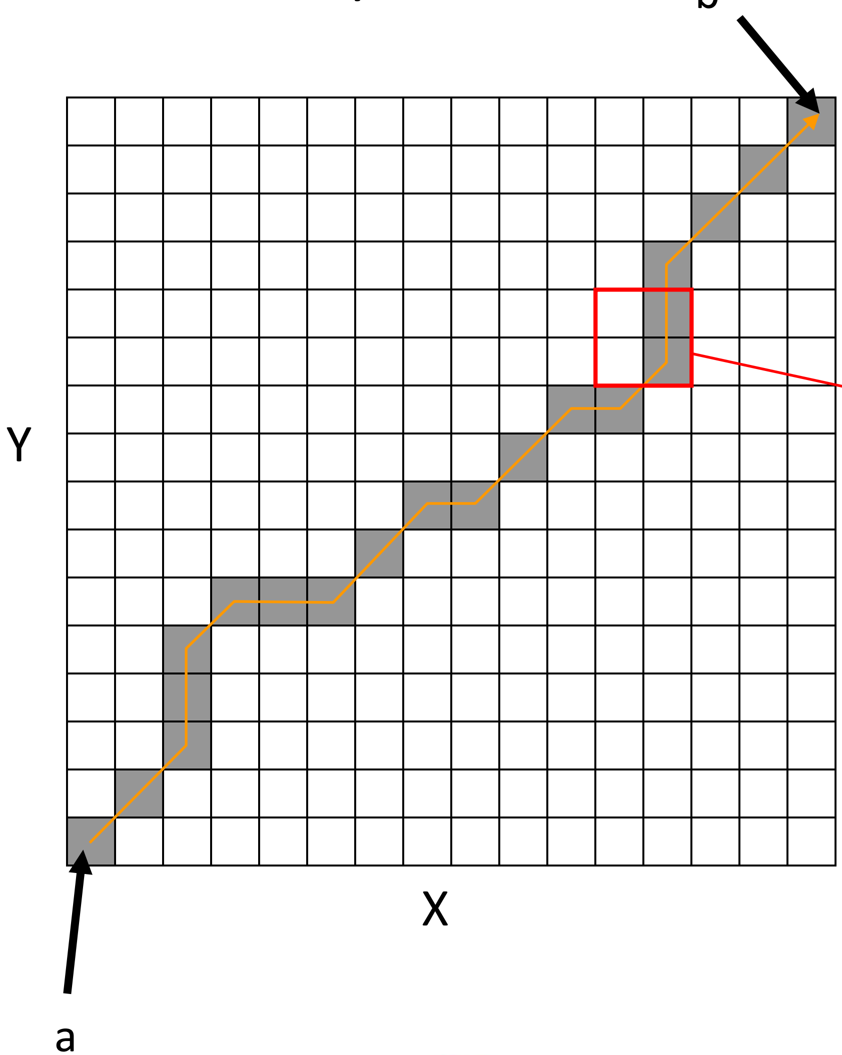


DTW: Euclidean Distance

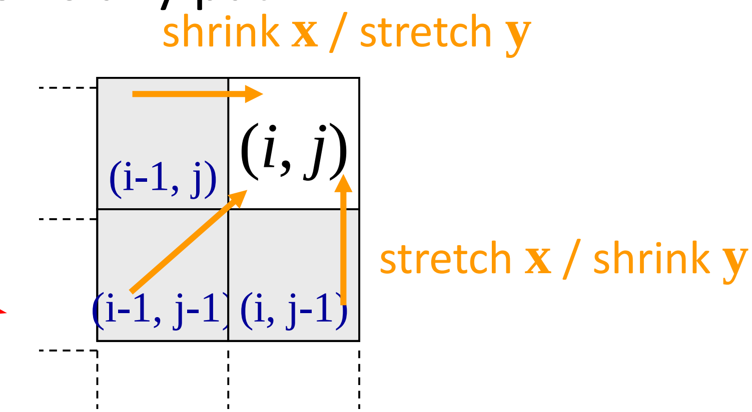


- Each cell *c* = (i, j) is a pair of indices whose corresponding values will be computed, $(x_i–y_j)^2$, and included in the sum for the distance.
- Euclidean path:
  - *i* = *j* always.
  - Ignores off-diagonal cells.

DTW: Dynamic time warping



DTW allows any path.

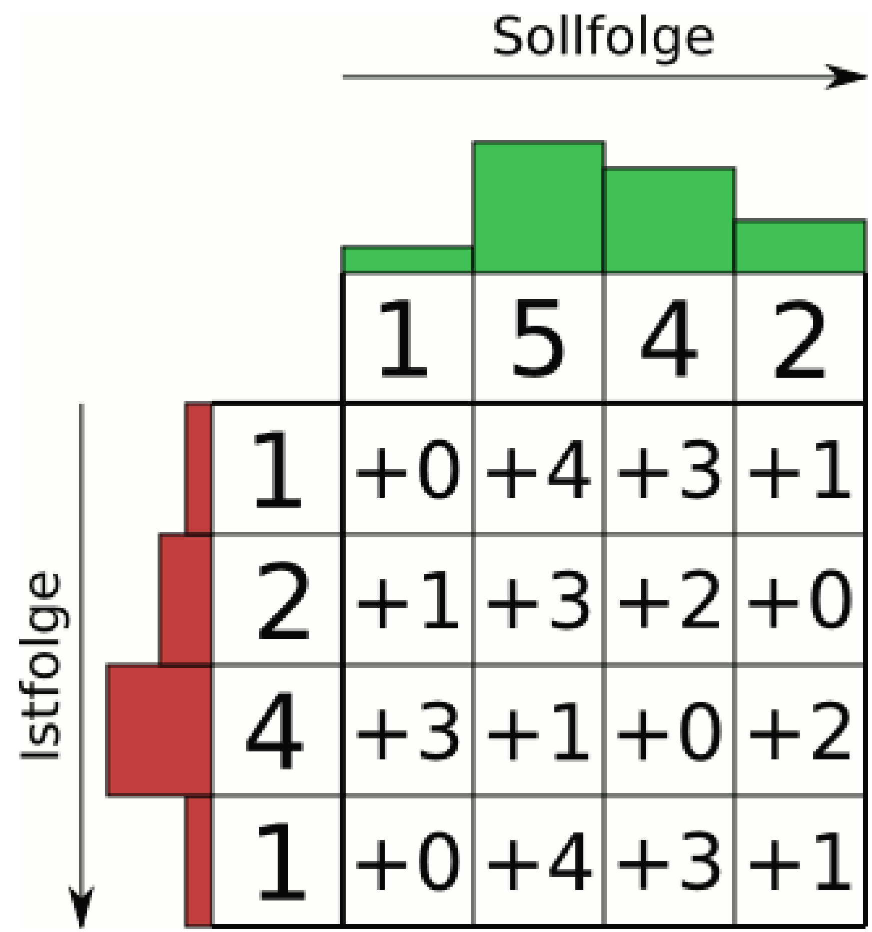


Dynamic Programming approach

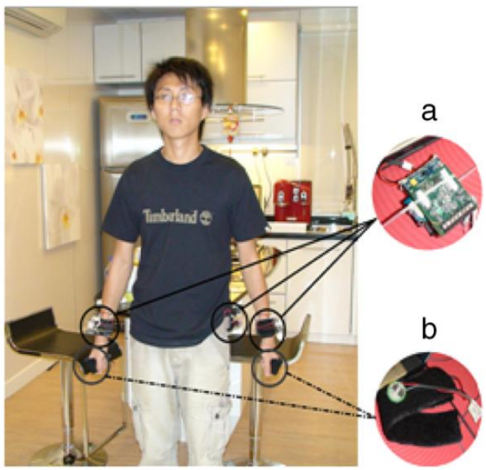
$$D(i, j) = |\, x_i - y_j \,| + \min \{\; D(i-1, j),\; D(i-1, j-1),\; D(i, j-1) \;\}$$

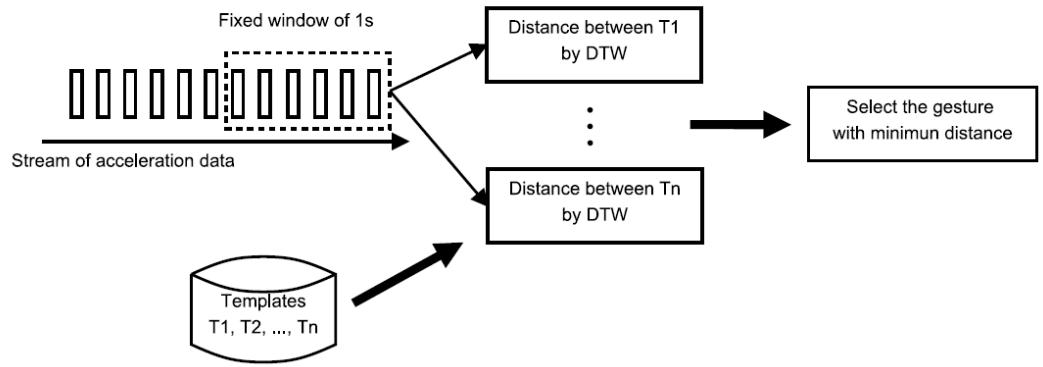
- Extend sequences by repeating elements
- Euclidean distance between extended sequences

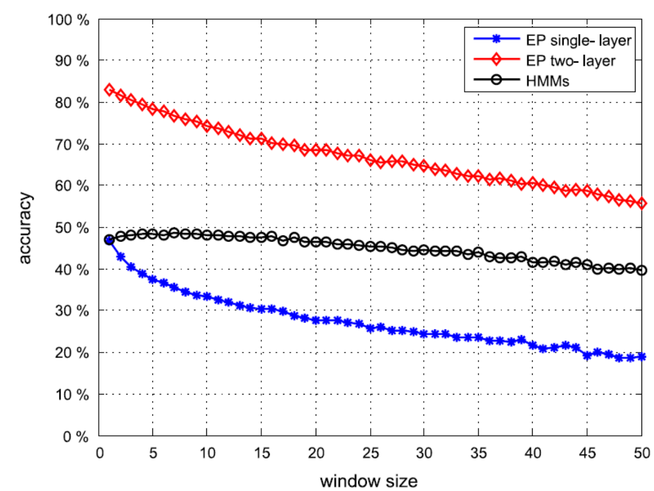
DTW example



DTW based activity recognition

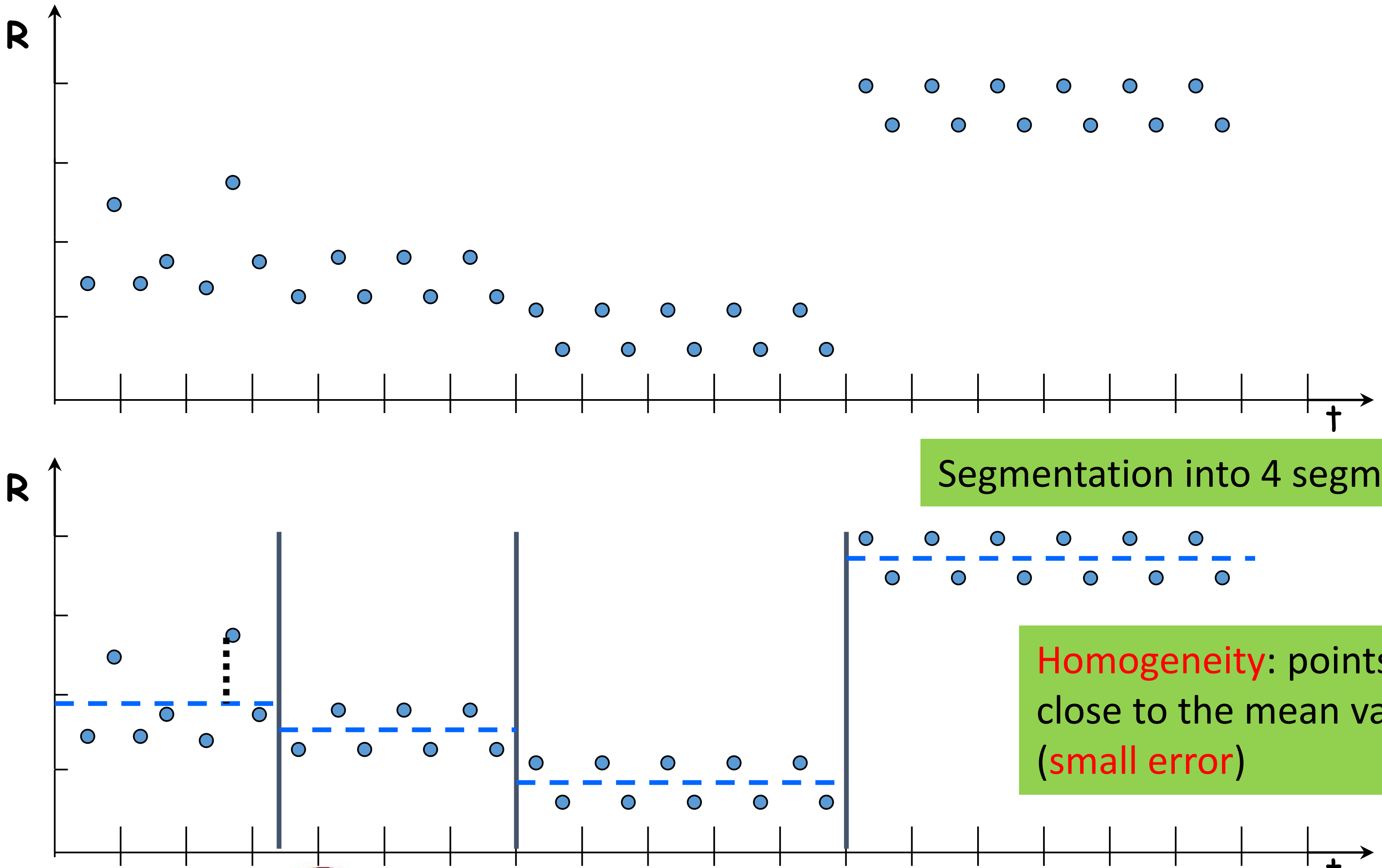






Wang, Liang, et al. "A hierarchical approach to real-time activity recognition in body sensor networks." *Pervasive and Mobile Computing* 8.1 (2012): 115-130.

Stream Data Processing



Segmentation into 4 segments

Homogeneity: points are close to the mean value (small error)

The K-segmentation problem

- A K-segmentation S: a partition of T into K contiguous segments $\{s_1, s_2, \ldots, s_K\}$.
- Similar to K-means clustering, but now we need the points in the clusters to respect the order of the sequence

Given a sequence T of length N and a value K, find a K-segmentation $S = \{s_1, s_2, \ldots, s_K\}$ of T such that the SSE error E is minimized.

Solve via Dynamic Programming:

- Construct the solution of the problem by using solutions to problems of smaller size
- Build the solution bottom up from smaller to larger instances

Outlier detection

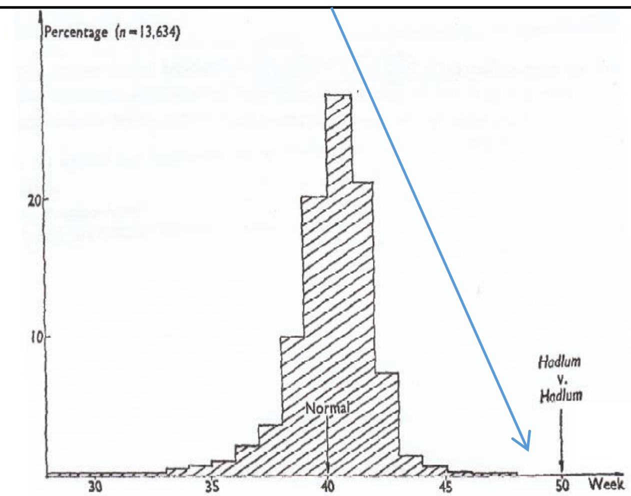
Definition (anomaly/novelty detection)

“those measurements that significantly deviate from the normal pattern of the sensed data”

Types: Noise, Errors, Events & Attacks

Outlier Detection	Event Detection
No prior knowledge	semantics
comparative	Threshold based
False alarms	Detection

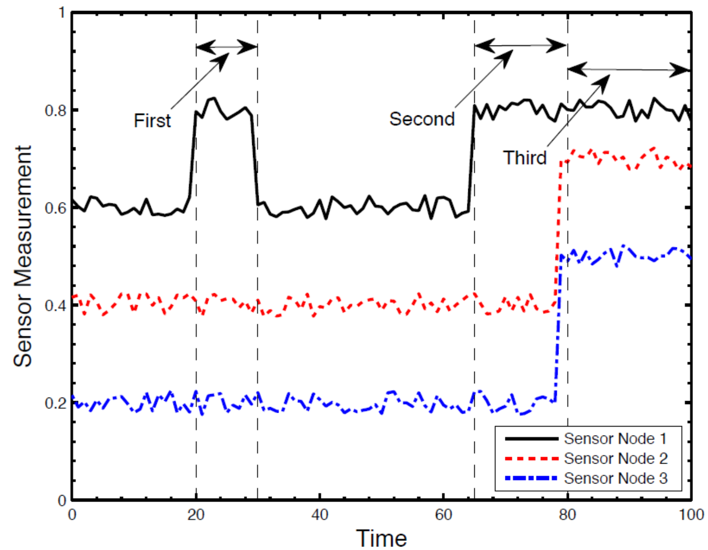
The birth of a child to Mrs. Hadlum happened 349 days (11,5 months) after Mr. Hadlum left for military service.



Distribution of human gestation periods.

Types of outliers

- First Order Anomalies:
  - Partial data measurements are anomalous at a sensor node
- Second Order Anomalies:
  - All data measurements at a sensor node are anomalous
- Third Order Anomalies:
  - Data from a set of sensor nodes are anomalous

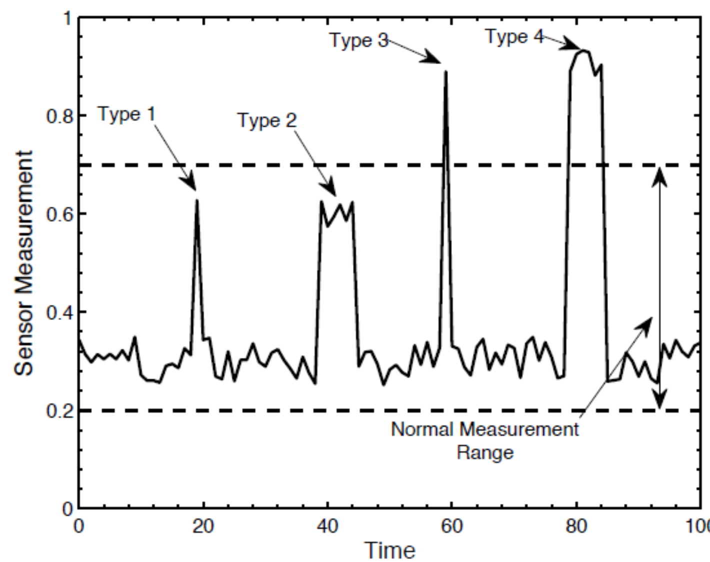


Type 1: Incidental absolute errors:
- A short-term extremely high anomalous

Type 2: Clustered absolute errors:
- A continuous sequence of *type 1* errors

Type 3: Random errors:
- Short-term observations outside normal range

Type 4: Long term errors:
- A continuous sequence of *type 3* errors



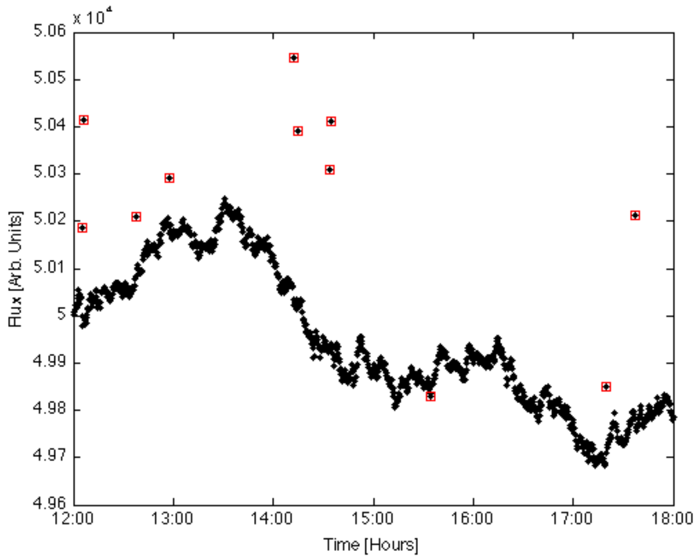
Outlier detection in WSNs

Objectives

- ➢ Data reliability
- ➢ Quality of Service
- ➢ Communications overhead
- ➢ Adaptive sampling rates
- ➢ Security alert

Applications

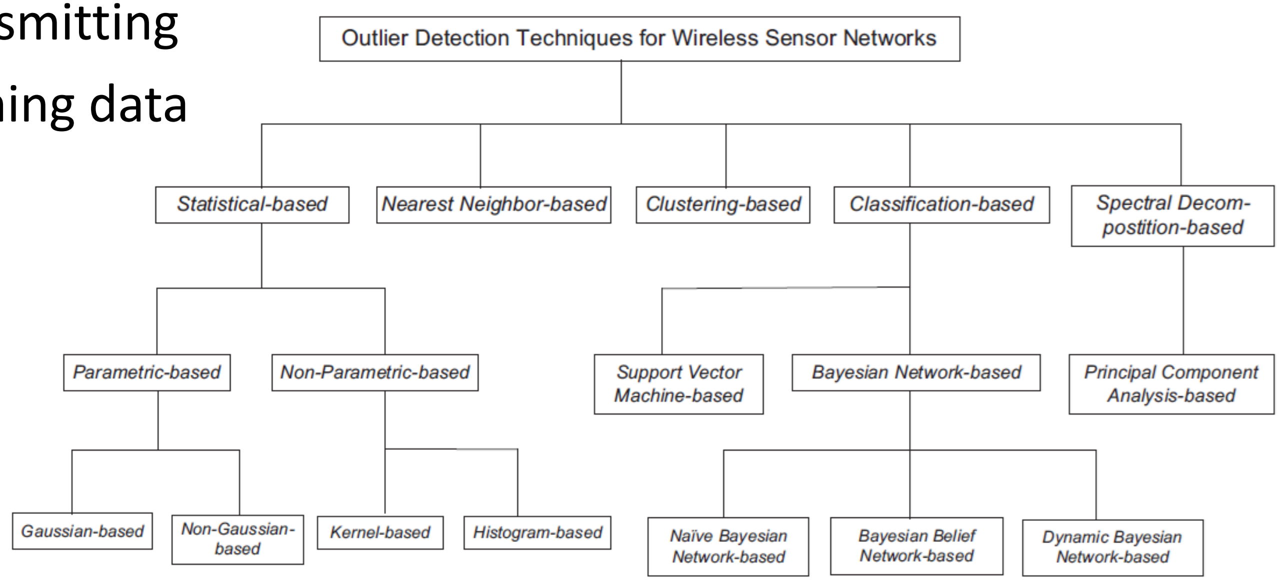
- Environmental monitoring (e.g. fire)
- Health monitoring (e.g. heart attack)
- Industrial monitoring (e.g. malfunctions)



Outlier detection in WSNs

Challenges

- Low cost & quality
- Processing vs Transmitting
- Distributed streaming data
- Network topology
  - Failures,
  - Disconnections,
  - Mobility
- Deployment scale
- Type detection



Statistical

Gaussian-based models

- Send measurements -> model
- Build model -> send parameters

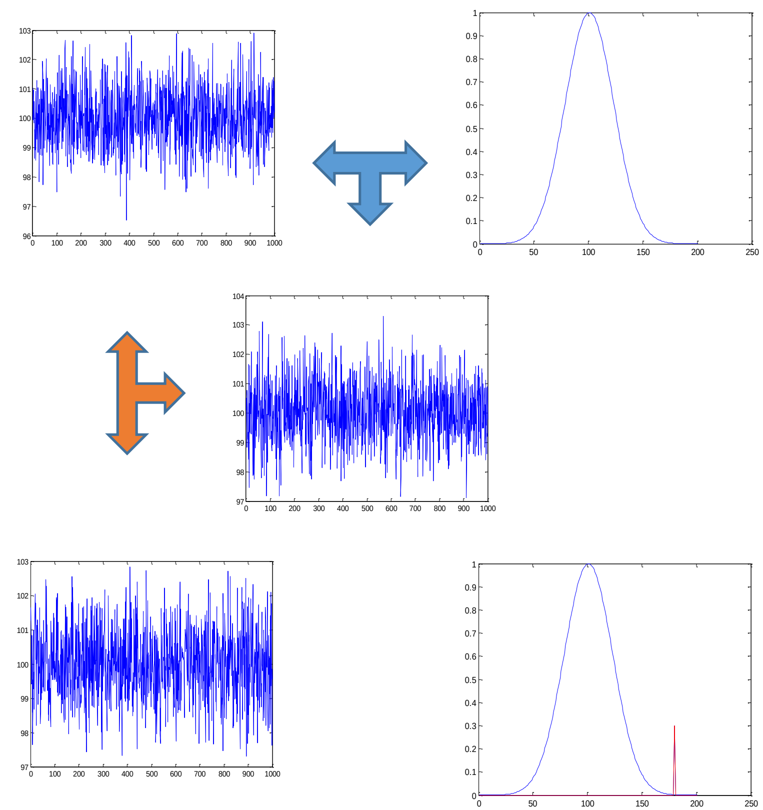
Non-Gaussian

- Symmetric α-stable distributions

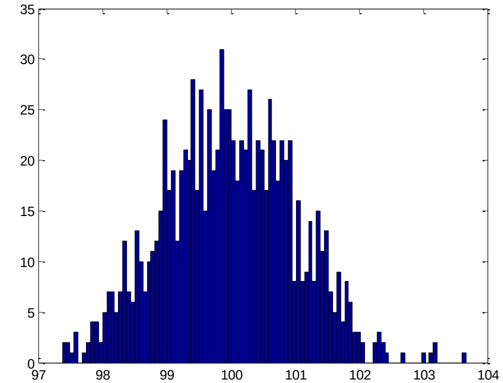
Mixtures

Clusters

Detection Thresholds



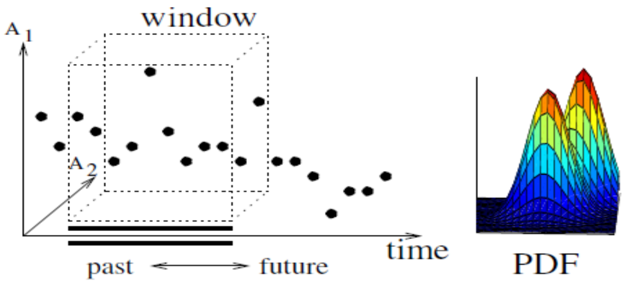
Non-parametric modeling



Histogram based

1. Obtain v_{min} and v_{max} information
2. Collect histogram
3. Collect outliers and potential outliers
4. Diffuse potential outliers and count the number of neighbors within d

- Number of bins
- Thresholds



Kernel Density Estimation

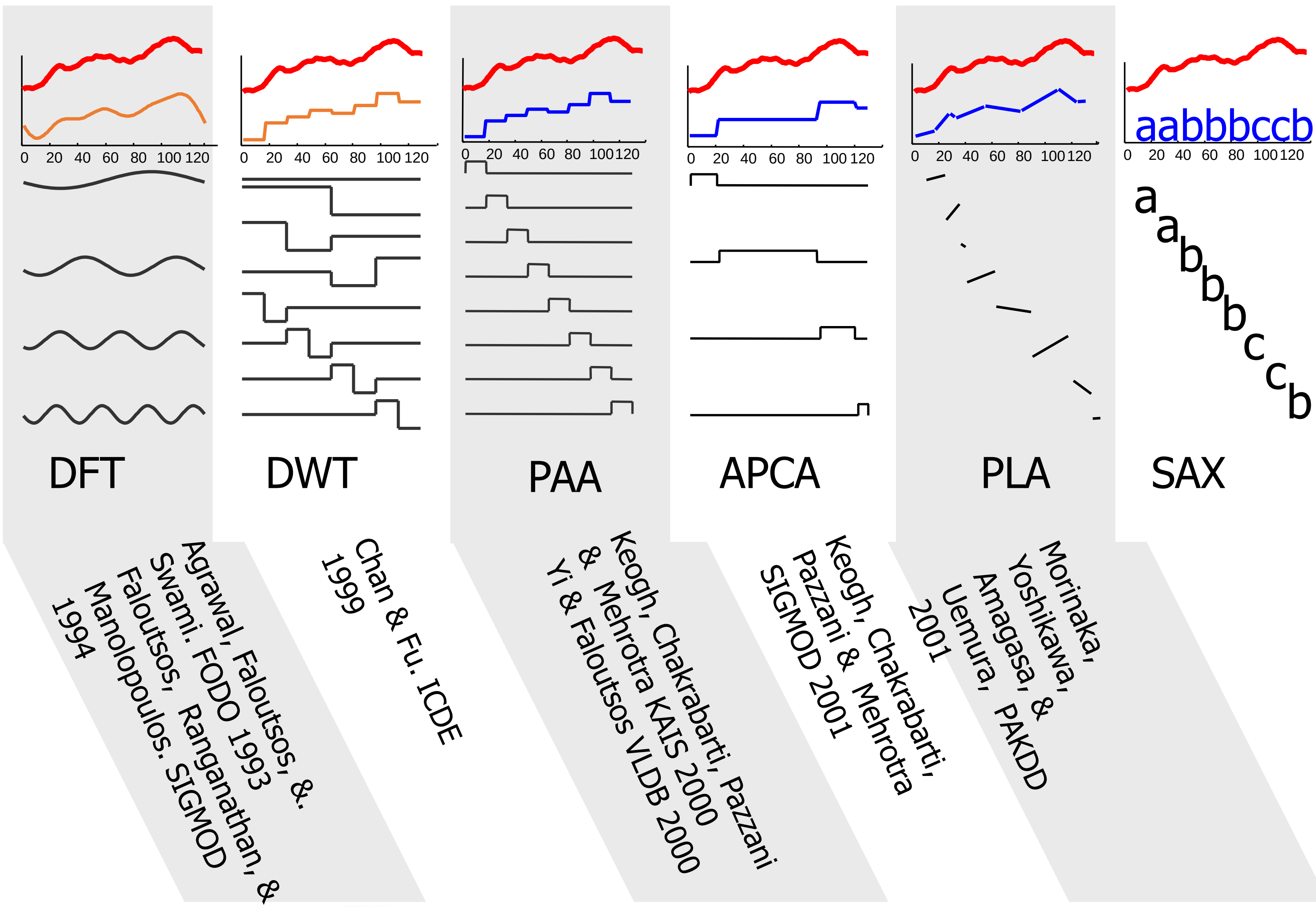
$$f(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_i}K\left(\frac{x-x_i}{h_i}\right)$$

Kernel

Bandwidth

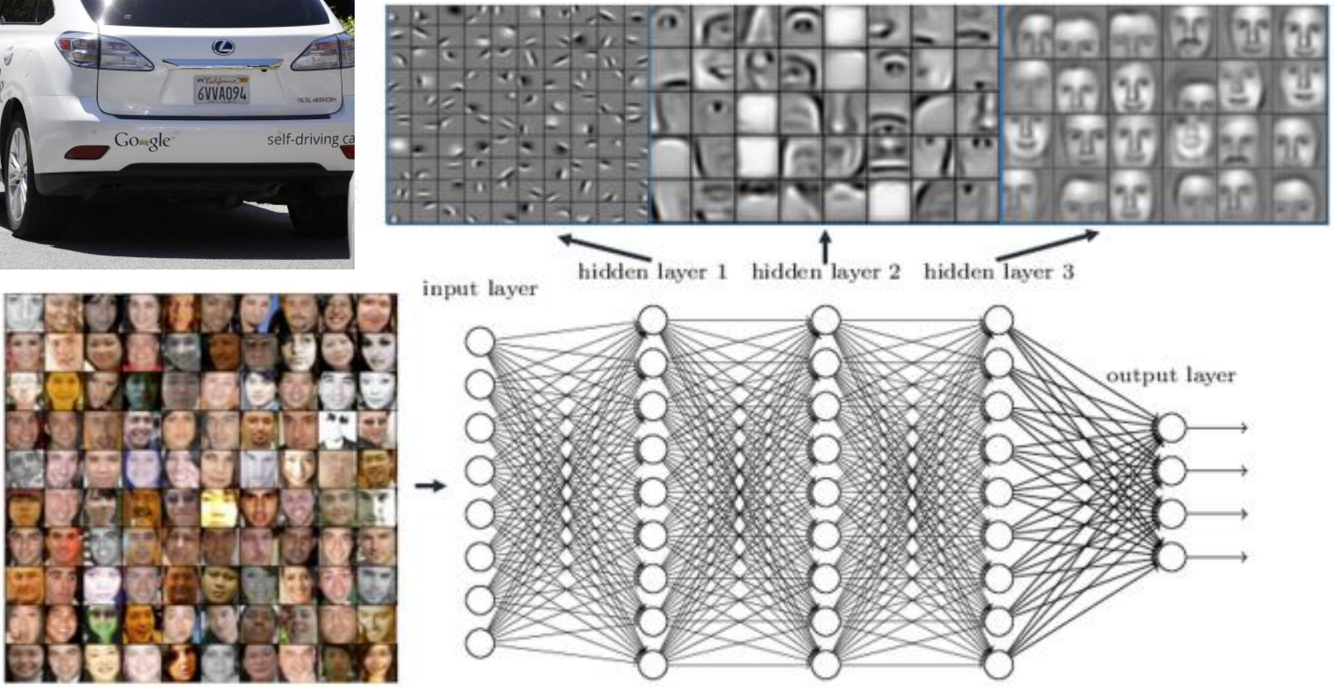
Gaussian $K(u) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}u^2}$

Time series



DFT
DWT
PAA
APCA
PLA
SAX
aabbbccb
0 20 40 60 80 100 120
Agrawal, Faloutsos, &. Swami. FODO 1993
Faloutsos, Ranganathan, & Manolopoulos. SIGMOD 1994
Chan & Fu. ICDE 1999
Keogh, Chakrabarti, Pazzani & Mehrotra KAIS 2000
Yi & Faloutsos VLDB 2000
Keogh, Chakrabarti, Pazzani & Mehrotra SIGMOD 2001
Morinaka, Yoshikawa, Amagasa, & Uemura, PAKDD 2001

Machine Learning



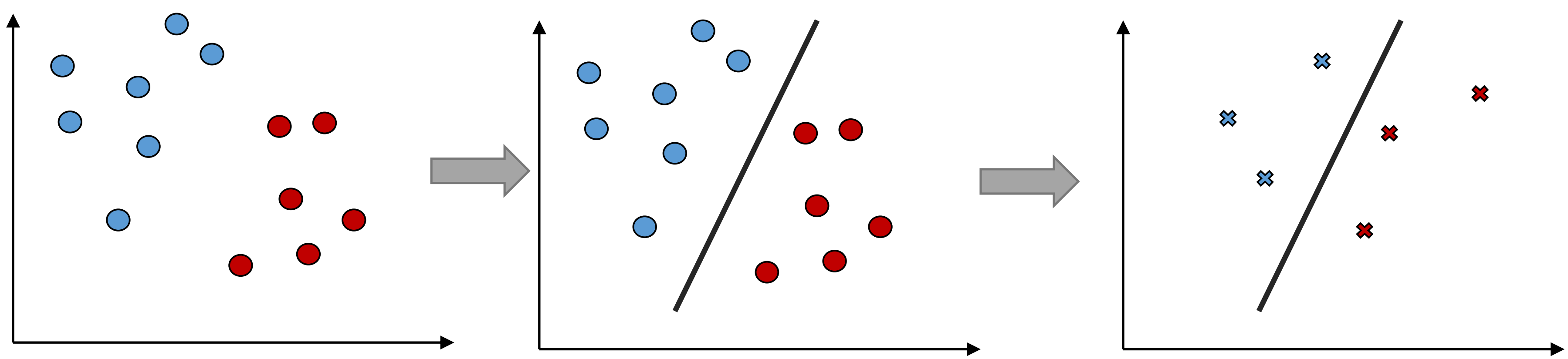
Machine Learning

Machine learning: construction and study of algorithms that can learn from data

- Models of example inputs (training data) → make predictions or decisions on new inputs (testing data)
- Data: characteristics
- Prior assumptions: a priori knowledge
- Representation: How do we represent the data
- Model / Hypothesis space: Hypotheses to explain the data
- Feedback / learning signal: Learning signal (delayed, labels)
- Learning algorithm: Model update
- Evaluation: Check quality

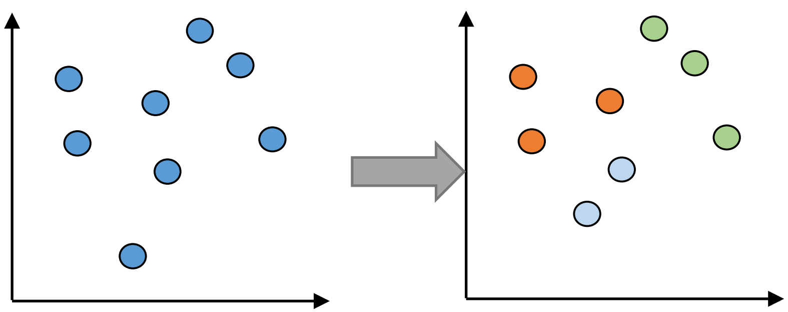
Types of ML

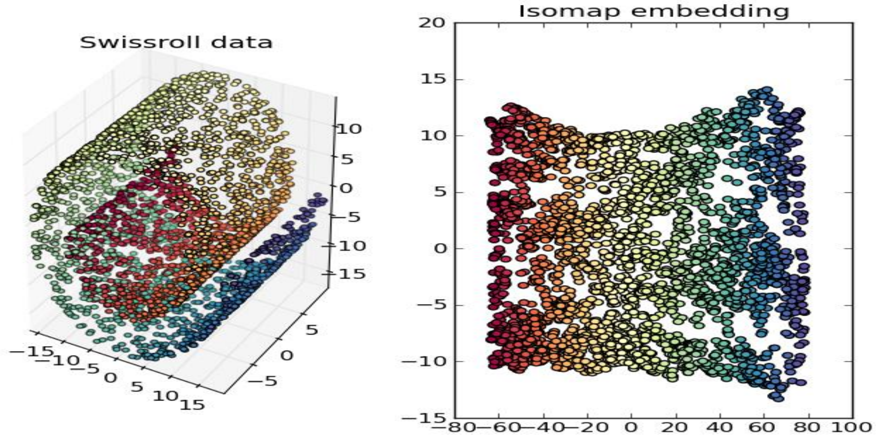
Supervised learning: present example inputs and their desired outputs (**labels**) → learn a general rule that maps inputs to outputs.



Types of ML

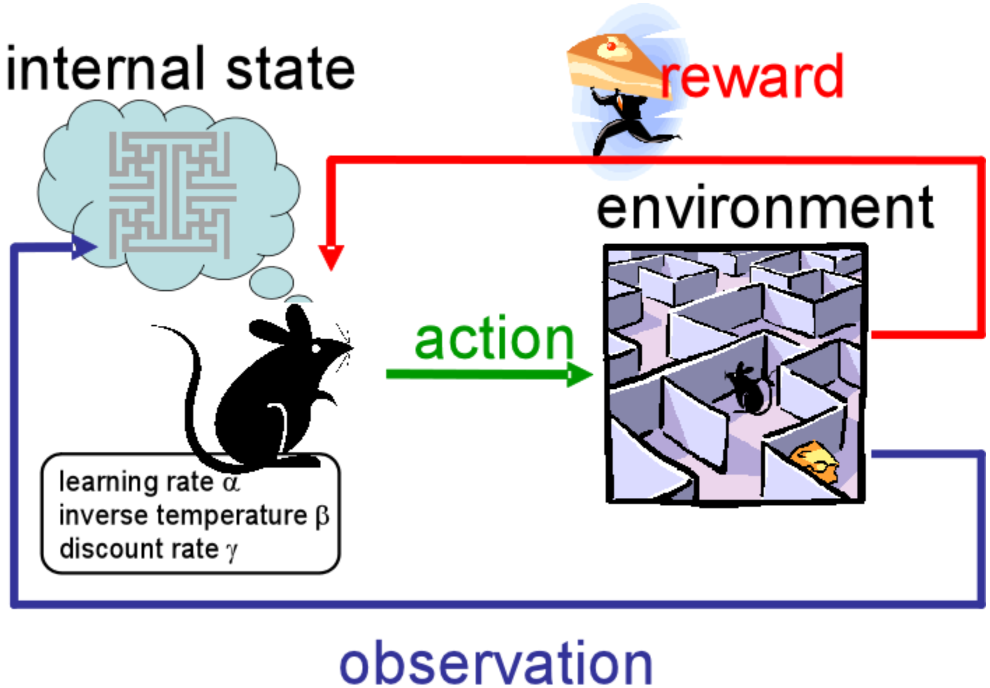
Unsupervised learning: no labels are given → find structure in input.





Types of ML

Reinforcement learning: system interacts with environment and must perform a certain goal without explicitly telling it whether it has come close to its goal or not.



Applications in WSNs

Network performance optimization

- Routing
- Distributed regression framework
- Data Aggregation
- Localization and Objects Targeting
- Medium Access Control

Data Mining

- Activity recognition
- Event Detection and Query Processing

Unsupervised learning - Clustering

What is a cluster?

groups of data instances that are similar to each other in one cluster and data instances that are very different from each other into different clusters

Hard vs. Soft

- *Hard*: belong to single cluster
- *Soft*: belong to multiple clusters

Flat vs. Hierarchical

- *Flat*: clusters are flat
- *Hierarchical*: clusters form a tree

K-means clustering

- K-means is a partitional clustering algorithm
- Let the set of data points (or instances) *D* be

  $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$,

  where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in a real-valued space $X \subseteq R^r$, and r is the number of attributes (dimensions) in the data.
- The *k*-means algorithm partitions the given data into *k* clusters.
  - Each cluster has a cluster **center**, called **centroid**.
  - *k* is specified by the user

K-means algorithm

Given *k*, the *k-means* algorithm works as follows:

1) Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers
2) Assign each data point to the closest centroid
3) Re-compute the centroids using the current cluster memberships.
4) If a convergence criterion is not met, go to 2).

Stopping criteria

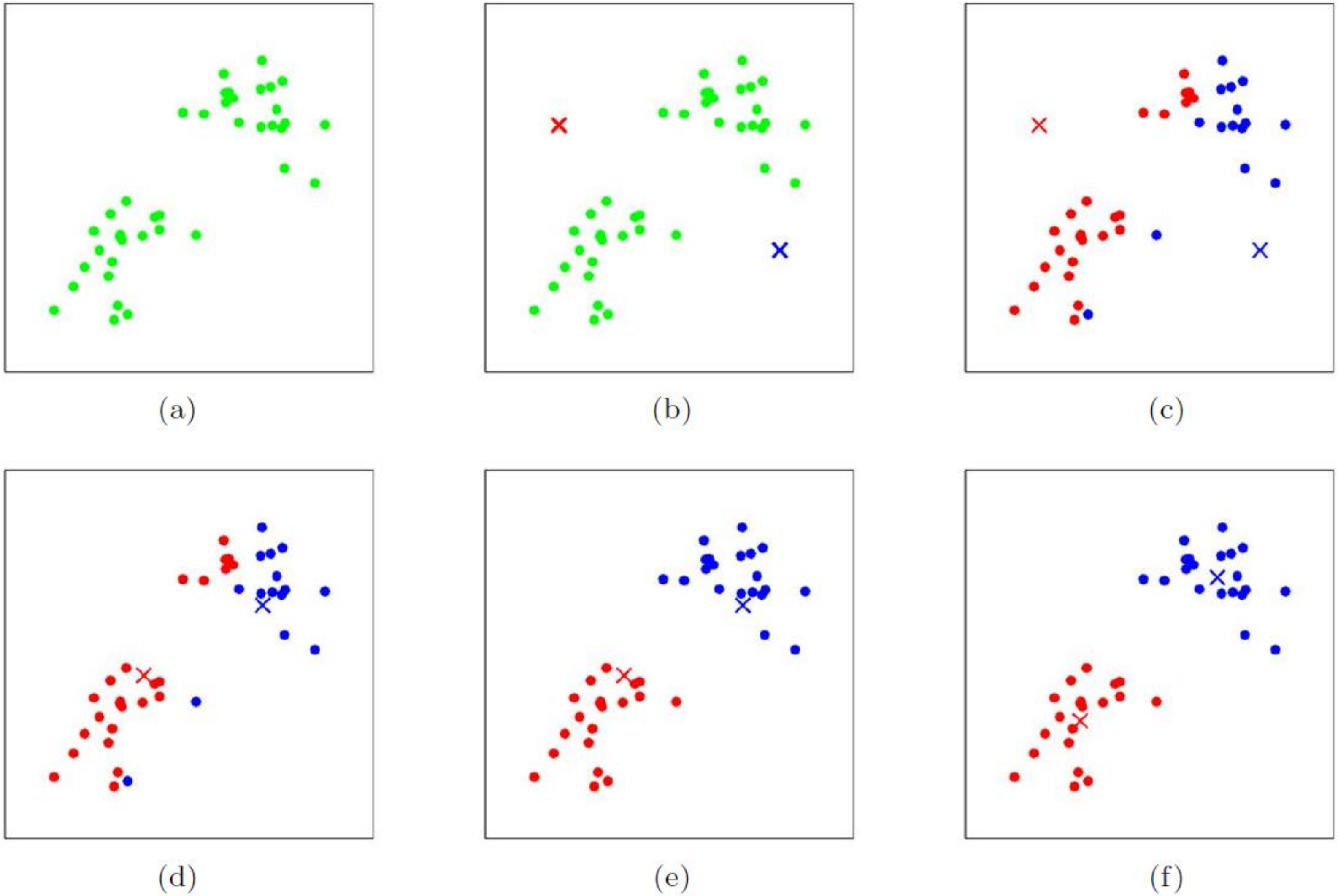
- no re-assignments of data points to different clusters
- no change of centroids
- minimum decrease in the $SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$

K-means example

Complexity is O(n * K * I * d)

n = number of points, K = number of clusters,

I = number of iterations, d = dimensionality

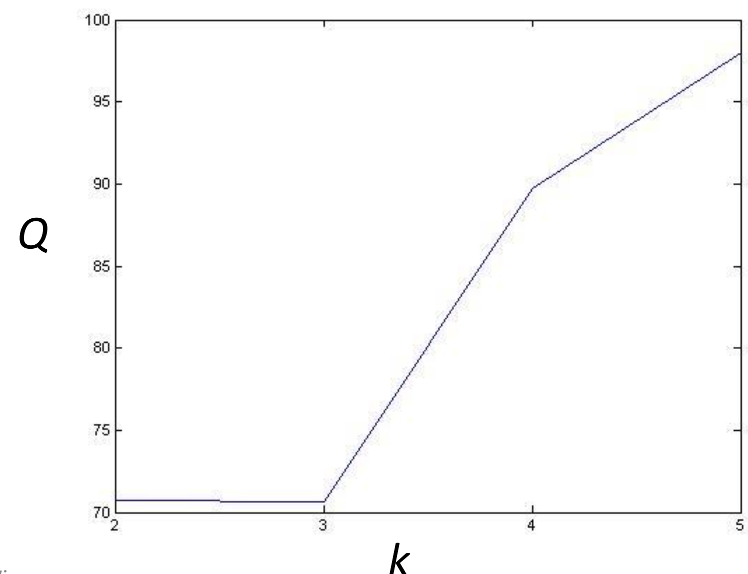


Issues with K-means

- Random initialization -> different clusters each time
- Data points are assigned to only one cluster
- Implicit assumptions about the “shapes” of clusters
- You have to pick the number of clusters…

Cluster tightness

$$Q = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in C_i} d(\boldsymbol{x}, \mu_i)$$

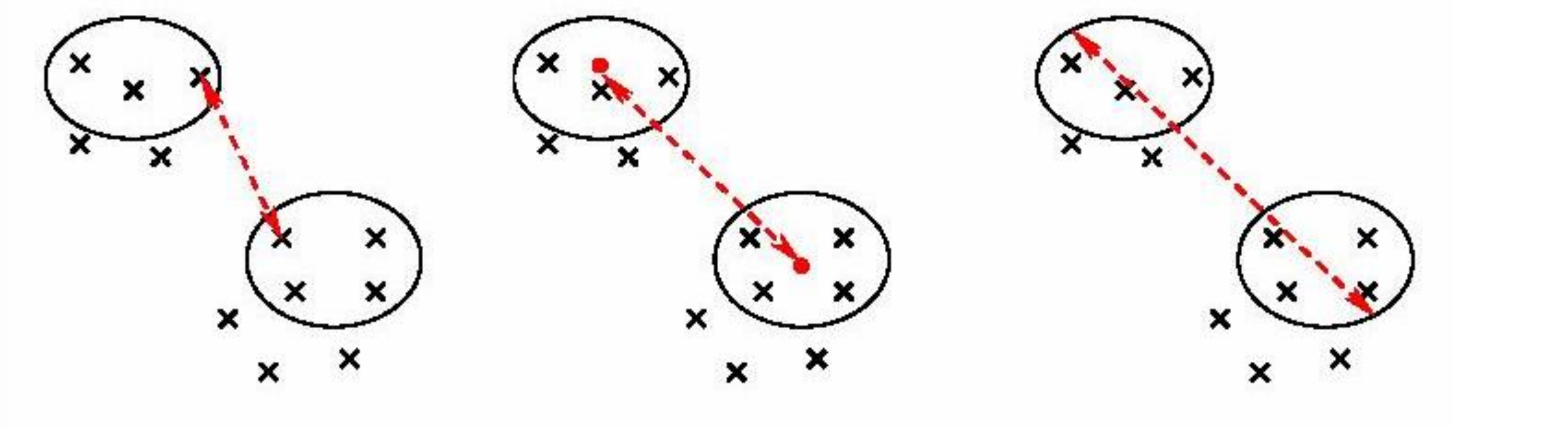


Distance Between Two Clusters

single-link clustering: distance between clusters -> shortest distance between any two members.

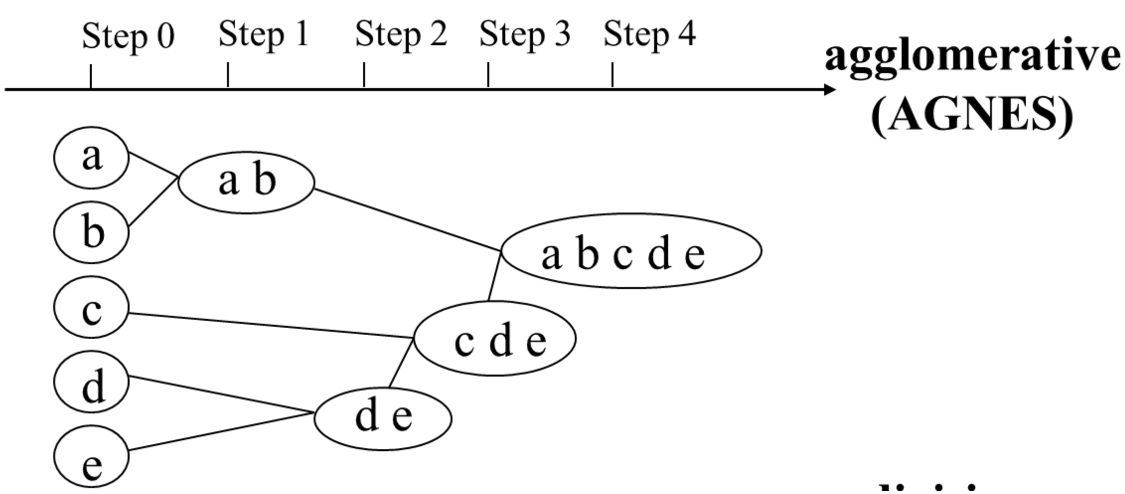
complete-link clustering: distance between clusters -> longest distance between any two members.

average-link clustering: distance between clusters -> average distance between any two members

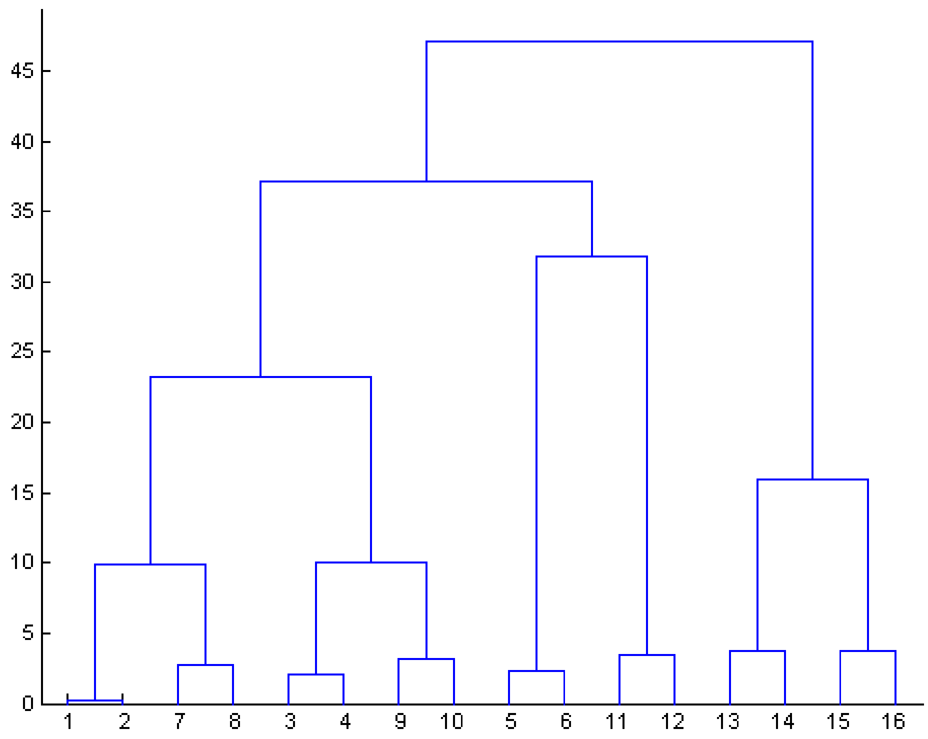


Hierarchical Agglomerative Clustering

- We start with every data point in a separate cluster
- We keep merging the most similar pairs of data points/clusters until we have one big cluster left
- This is called a bottom-up or agglomerative method



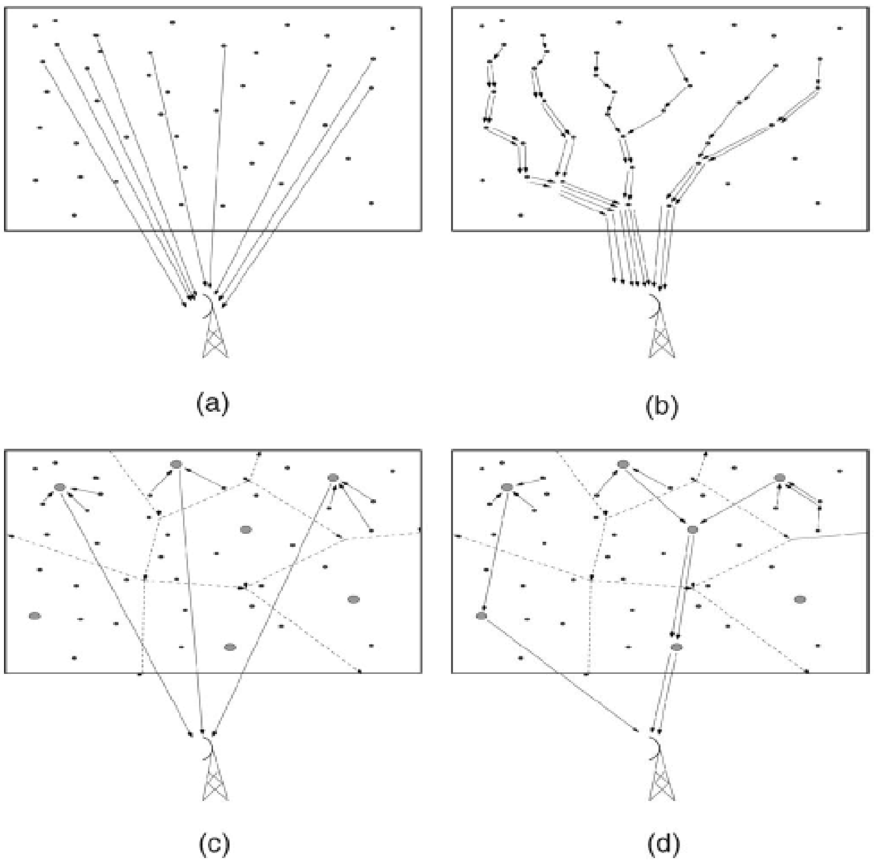
Hierarchical Clustering (cont.)



- This produces a binary tree or ***dendrogram***
- The final cluster is the root and each data item is a leaf
- The height of the bars indicate how close the items are

Clustering in WSN

- Scalability:
  - Reduce routing tables to within cluster
- Data Aggregation
  - Energy reduction vs. full data transmission
  - CH based data fusion
  - multi-hop tree structure aggregation
- Load Balancing
  - Eliminate redundant data transmissions
  - Communications between CHs
- Energy reduction
  - Selective sampling within cluster
  - Short-range communications with CH
- Robustness & Fault tolerance
  - Support node failure/recovery
  - mobility of sensors
  - noisy measurements etc.
- Efficiency
  - Collision avoidance (intra vs. inter cluster communications)
  - Latency reduction by reducing hops
  - Network life-time maximization
  - Quality-of-service



Reading List

- Esling, Philippe, and Carlos Agon. "Time-series data mining." *ACM Computing Surveys (CSUR)* 45.1 (2012): 12.