



Medical Time Series Mining

23 Feb 2023 - 24 Feb 2023



Agenda

23 February

9h30-12h30:

Part I. Time series classification and counterfactual explanations, mainly using time series extracted from Electronic Health Records

12h30 - 13h30: Lunch

13h30 - 16h30:

Part II. Missingness in time series, inducing predictors and predictive patterns, mainly on time series of mHealth recordings

24 February

9h30-12h30:

Part III. Time series forecasting methods using time series obtained from the Intensive Care Unit

12h30 - 14h: Walk to Health House & Lunch

14h00-17h00:

Part IV. Health House

Sharing moment: short presentations by young researchers



Medical Time Series Mining

Part I. Time series classification and counterfactual explanations

Based on original slides by Prof. George Kollios, Prof. Dimitrios Gunopulos and Prof. Christos Faloutsos with some slides from tutorials by Prof. Eamonn Keogh, Prof. Panos Karras, and Dr. Michalis Vlachos.

Excellent tutorials (and not only) about Time Series can be found there:

<http://www.cs.ucr.edu/~eamonn/tutorials.html>

A nice tutorial on Matlab and Time series is also there:

<http://www.cs.ucr.edu/~mvlachos/ICDM06/>

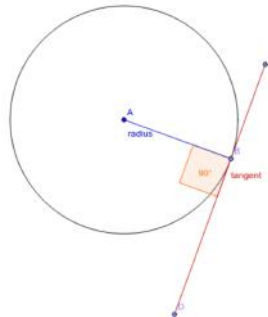
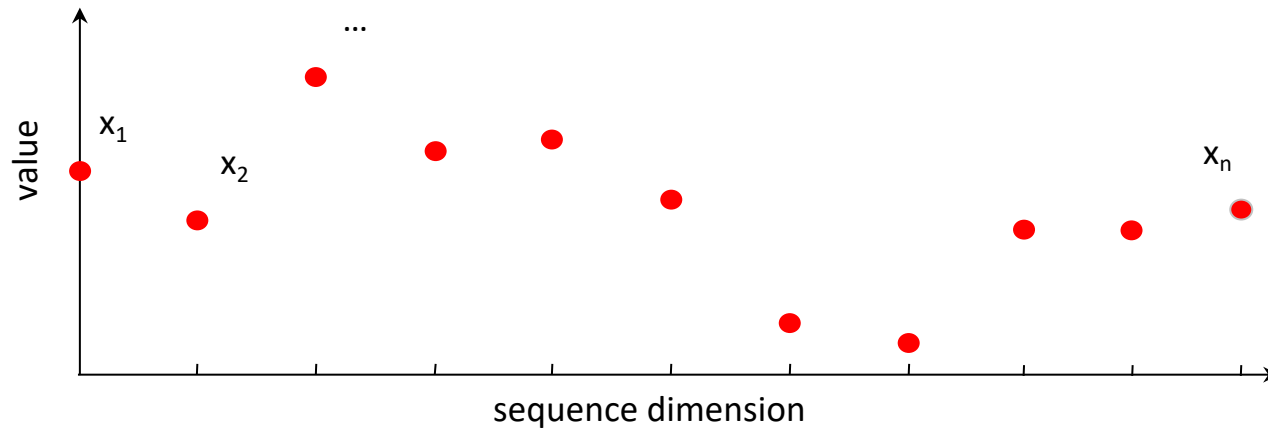
Panagiotis Papapetrou
Professor, Stockholm University

Outline

- Data series
- Electronic Health Records
- Time series summarization
- Time series classification
- Time series counterfactuals
- Challenges and future directions

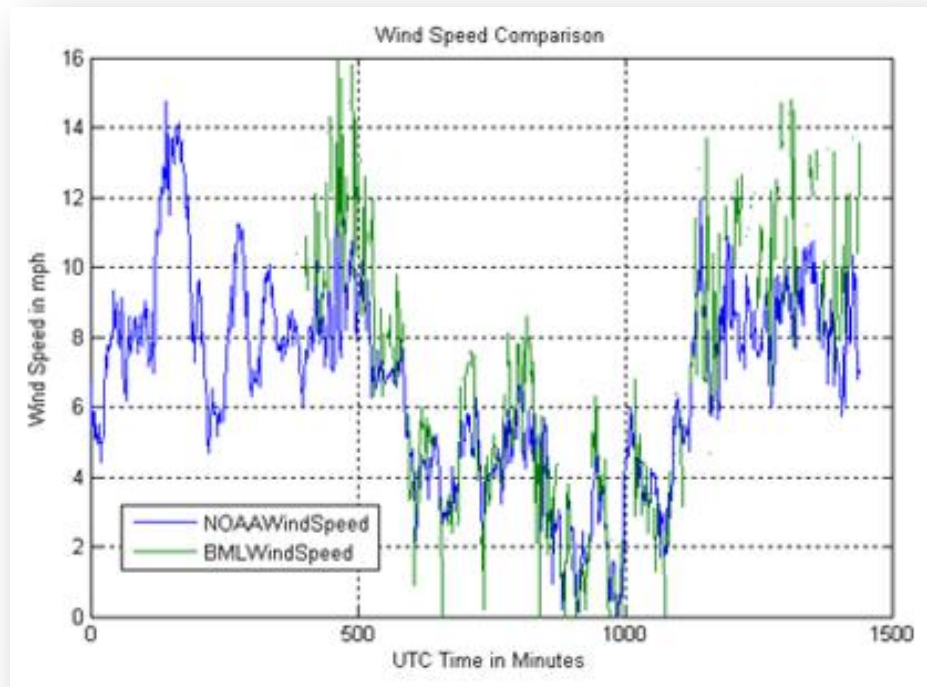
Data series

- Sequence of points ordered along some dimension



Data series

- Sequence of points ordered along some dimension



Wind speed

From ocean observing node project, <http://bml.ucdavis.edu/boon/wind.html>

Data series

- Sequence of points ordered along some dimension

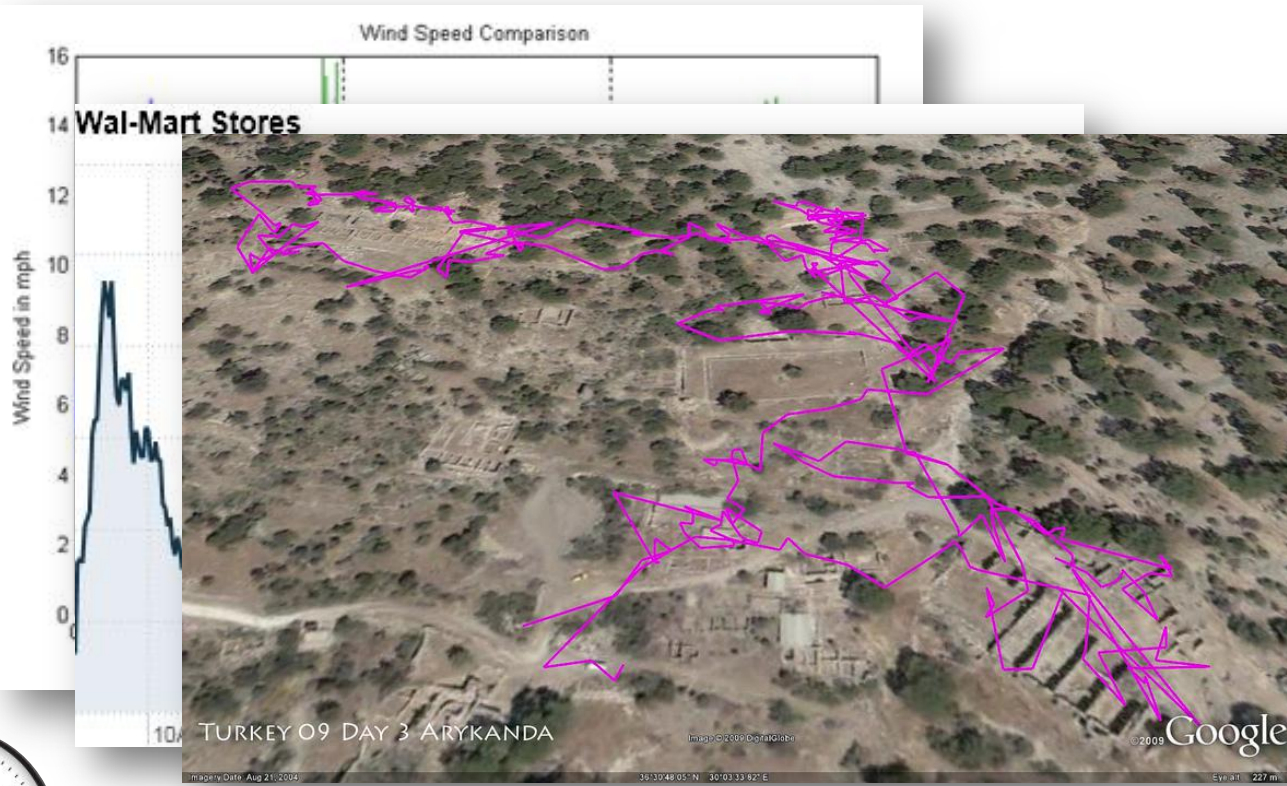


Historical stock quotes

From http://money.cnn.com/2012/04/23/markets/walmart_stock/index.htm

Data series

- Sequence of points ordered along some dimension

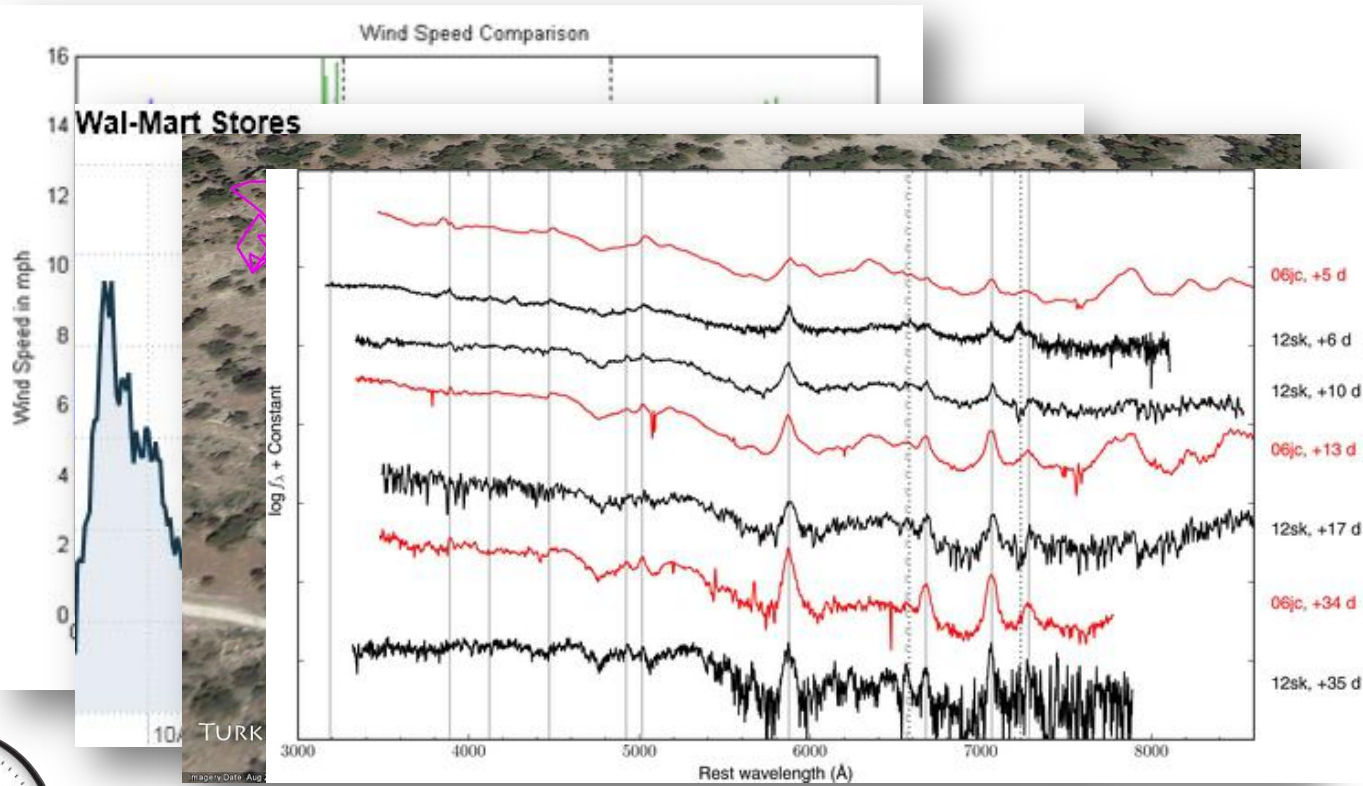


Trajectories from GPS logs

From <http://www.flickr.com/photos/kitepuppet/3604115258>

Data series

- Sequence of points ordered along some dimension

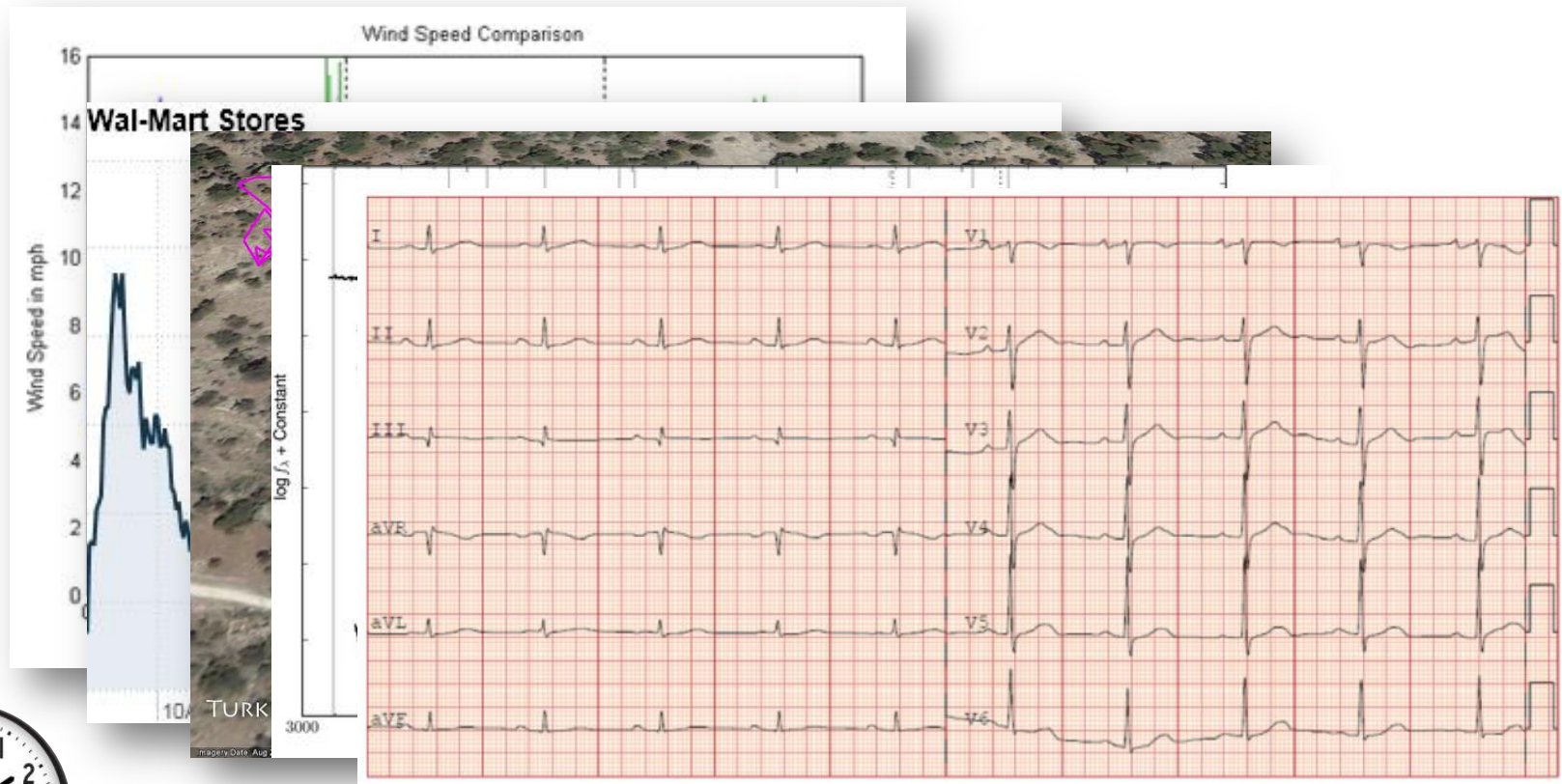


Spectroscopic sequence data (astronomy)

From Sanders et al., <http://dx.doi.org/10.1088/0004-637X/769/1/39>

Data series

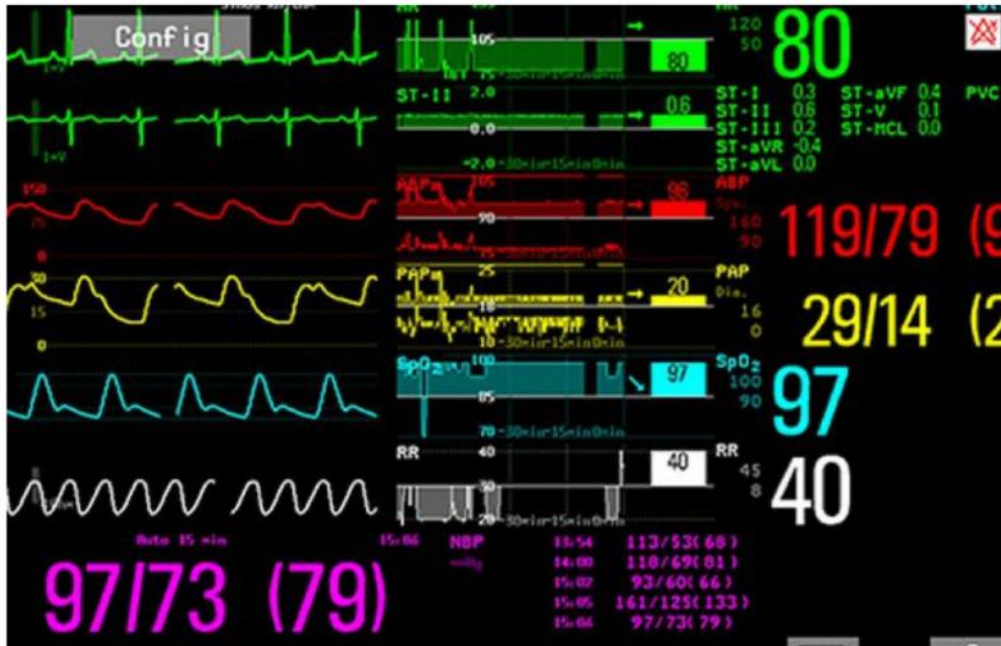
- Sequence of points ordered along some dimension



Electocardiograms (cardiology)

<https://archive.physionet.org/physiobank/>

Medical time series - in the ICU



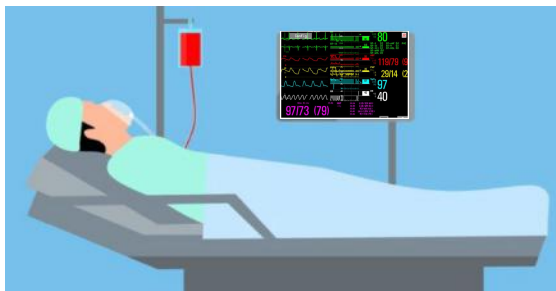
heart rate

systolic/diastolic blood pressure

pulmonary artery pressure

blood oxygen supply

respiration rate



Over 100 variables are measured over time

Outline

- Data series
- Electronic Health Records
- Time series summarization
- Time series classification
- Time series counterfactuals
- Challenges and future directions

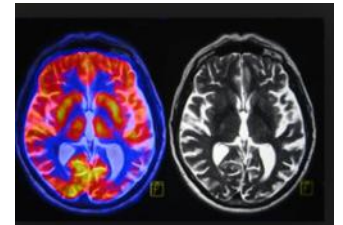
Electronic Health Records (EHRs)

Longitudinal collection of **electronic health information** about individual **patients** and **populations**

- **Diagnoses**
- **Drug prescriptions**
- **Laboratory measurements**
- **More complex structures**
 - clinical notes
 - medical images
 - MRIs
 - ECGs
 - ...

I25.110

A01AD05



ICD10* codes

- 10th revision of the **International Classification of Diseases and Related Health Problems**
- A classification system that is used to record medical activity
- The system contains 22 chapters and enables classification and quantification of diseases and other health-related issues

3 x-7
characters
long

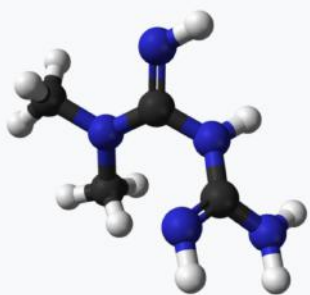
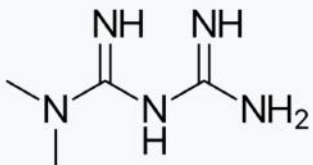


Chapter	Code Range	Estimated # of Codes	Description
1	A00-B99	1,056	Certain infectious and parasitic diseases
2	C00-D49	1,620	Neoplasms
3	D50-D89	238	Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism

* <http://www.ahima.org/icd10>

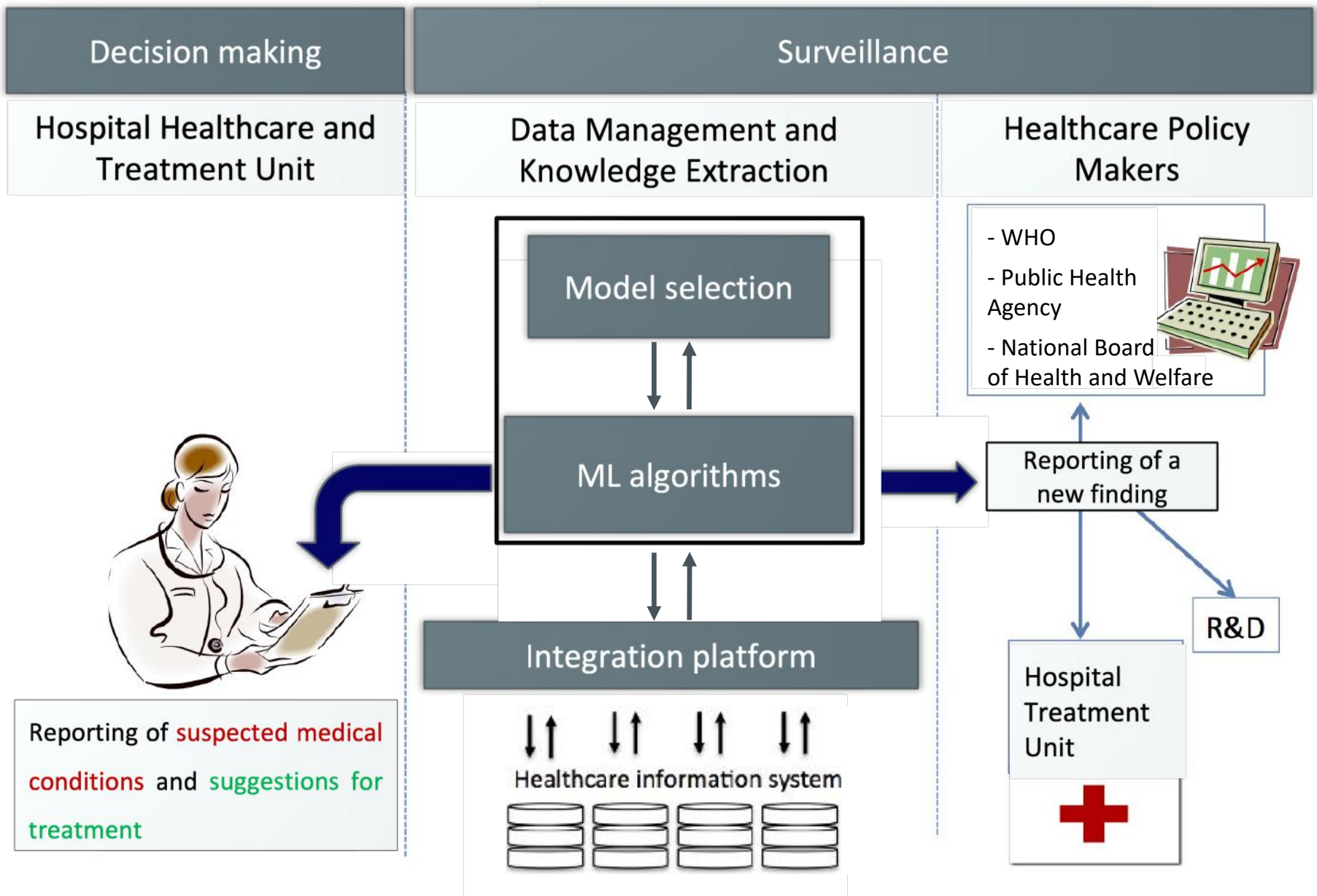
ATC* codes

- Anatomical Therapeutic Chemical codes, first published in 1976
- Used for classification of **active ingredients** of drugs
- Based on the organ/system on which they act: **therapeutic, pharmacological, and chemical** properties
- Controlled by the World Health Organization Collaborating Centre (WHOCC) for drug statistics methodology



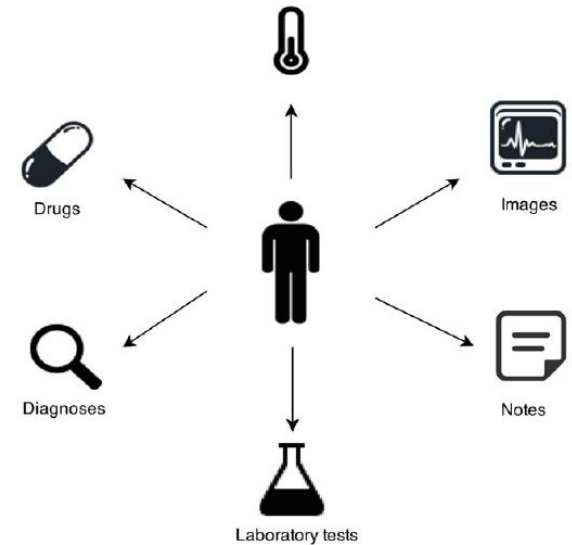
Level	Content	Type	Example
I	anatomical main group	1 letter	A : alimentary tract and metabolism
II	therapeutic subgroup	2 digits	A10 : diabetes drugs
III	pharmacological subgroup	1 letter	A10 B : blood glucose lowering drugs, excl. insulins
IV	chemical subgroup	1 letter	A10B A : <u>biguanides</u>
V	chemical substance	2 digits	A10BA 02 : metformin

* http://www.whocc.no/atc_ddd_index/



EHR features

- Mainly two feature types:
 - static features
 - temporal features

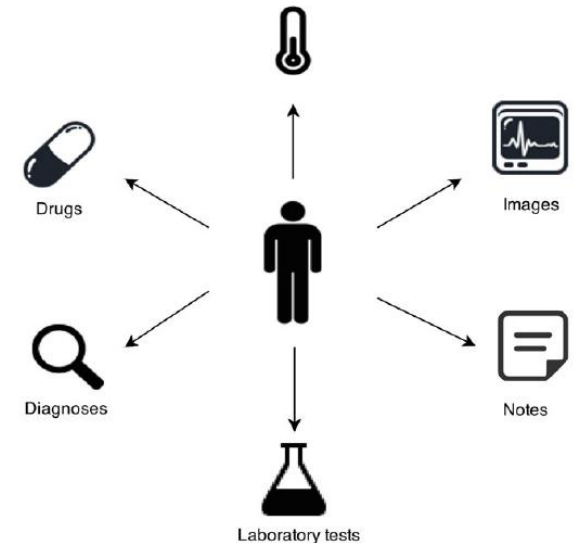


EHR features

- Mainly two feature types:

- static features
- temporal features

	ADE				...				...				...
	YES	✓			...		✓	✓	...		✓		...
	NO			✓	...				...			✓	...
	NO				...	✓			...		✓		...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮



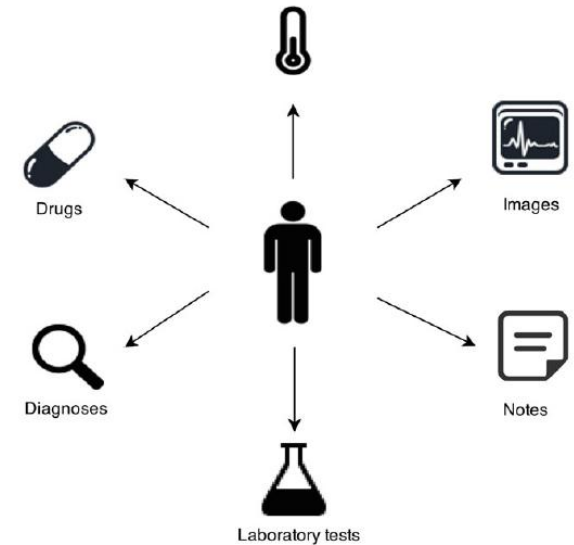
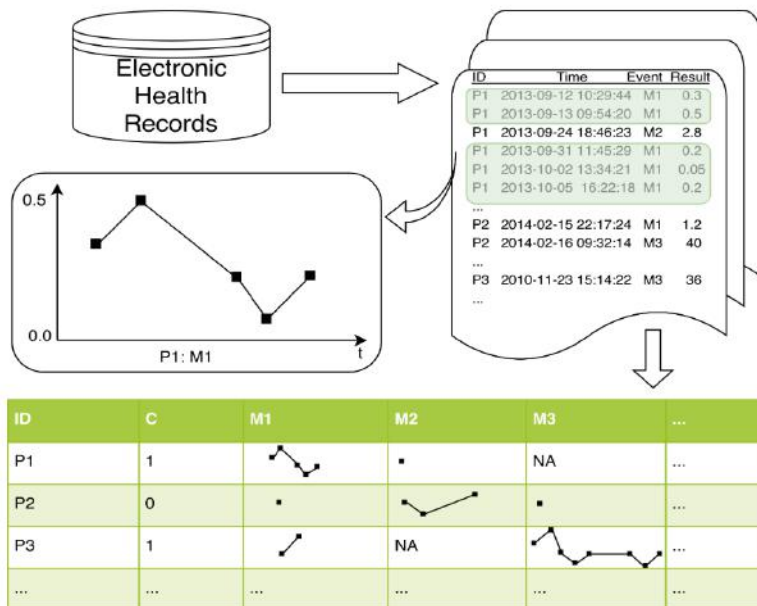
The **class labels** assigned depending on task at hand, e.g., ADE detection

- Existing out-of-the-box classifiers are used

- Decision trees, SVMs, XGBoost, deep learning architectures

EHR features

- Mainly two feature types:
 - static features
 - temporal features



Challenges:

- different measurement units
- varying times of measurement
- sparsity and missingness

More challenges

- **Fluctuations** of medical variables over time are highly relevant
- The **actual values** may be less relevant compared to **trends**
- Consider, for example, **fever**
- Segments of **consecutive value fluctuations** between [37.4, 38.2] can be summarized as *“low grade fever”*
- **More challenging:** multiple variables over time



C	F	RANGE
40.6	105.1	HIGH FEVER
40.4	104.7	
40.2	104.3	
40.0	104.0	
39.8	103.7	
39.6	103.3	
39.4	102.9	FEVER
39.2	102.6	
39.0	102.2	
38.8	101.8	
38.6	101.5	
38.4	101.2	LOW GRADE FEVER
38.2	100.8	
38.0	100.4	
37.8	100.1	
37.6	99.7	
37.4	99.3	NORMAL
37.2	99.0	
37.0	98.6	
36.0	96.8	
35.0	95.0	

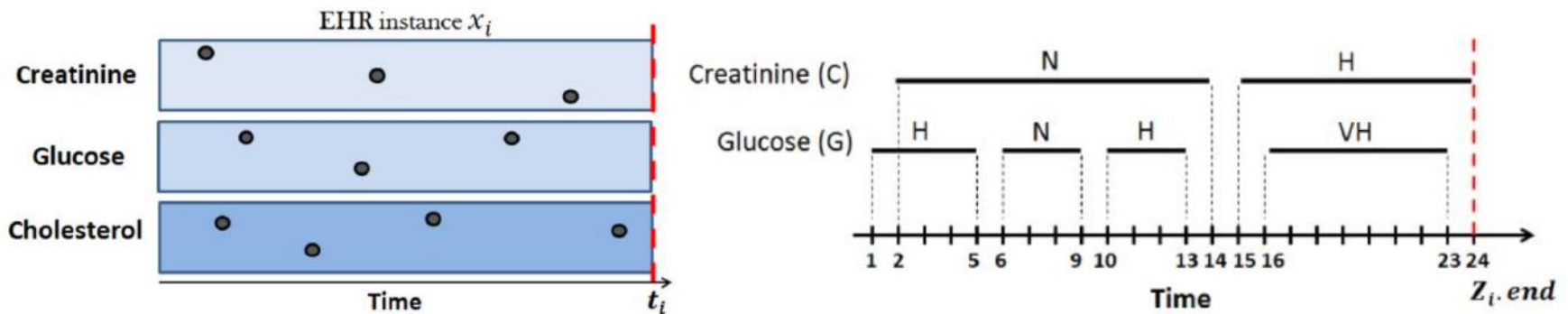
How to summarize time series?

Outline

- Data series
- Electronic Health Records
- Time series summarization
- Time series classification
- Time series counterfactuals
- Challenges and future directions

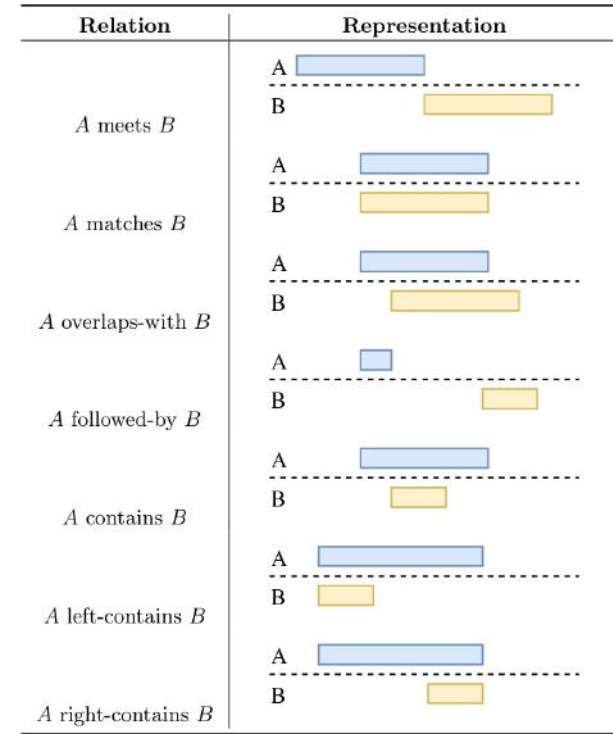
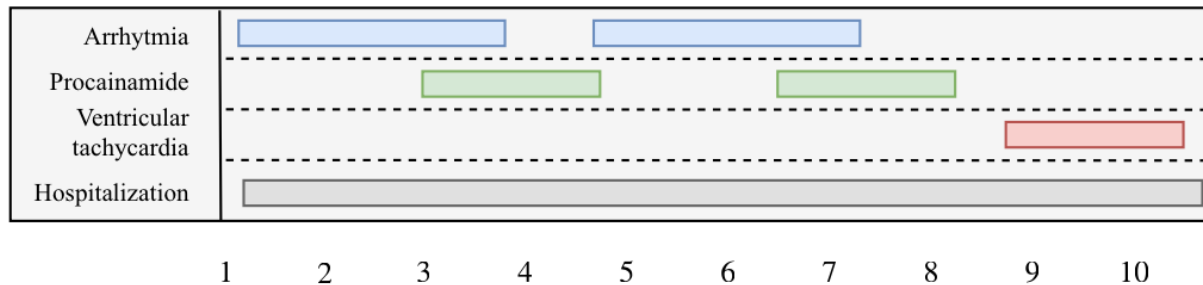
Temporal abstractions

- **Multiple temporal variables** registered and evolving concurrently
- Each variable with **multiple readings** until a **critical time point t_i** , e.g., glucose, creatinine, cholesterol
- **Class label**: diagnosis/symptom detected at time t_i (**event of interest**)
- **Main question**: are all values over time really relevant?



Temporal abstractions

- Trend abstraction:
 - e.g., decreasing, steady, increasing
- Value abstraction:
 - e.g., very low, low, normal, high, very high

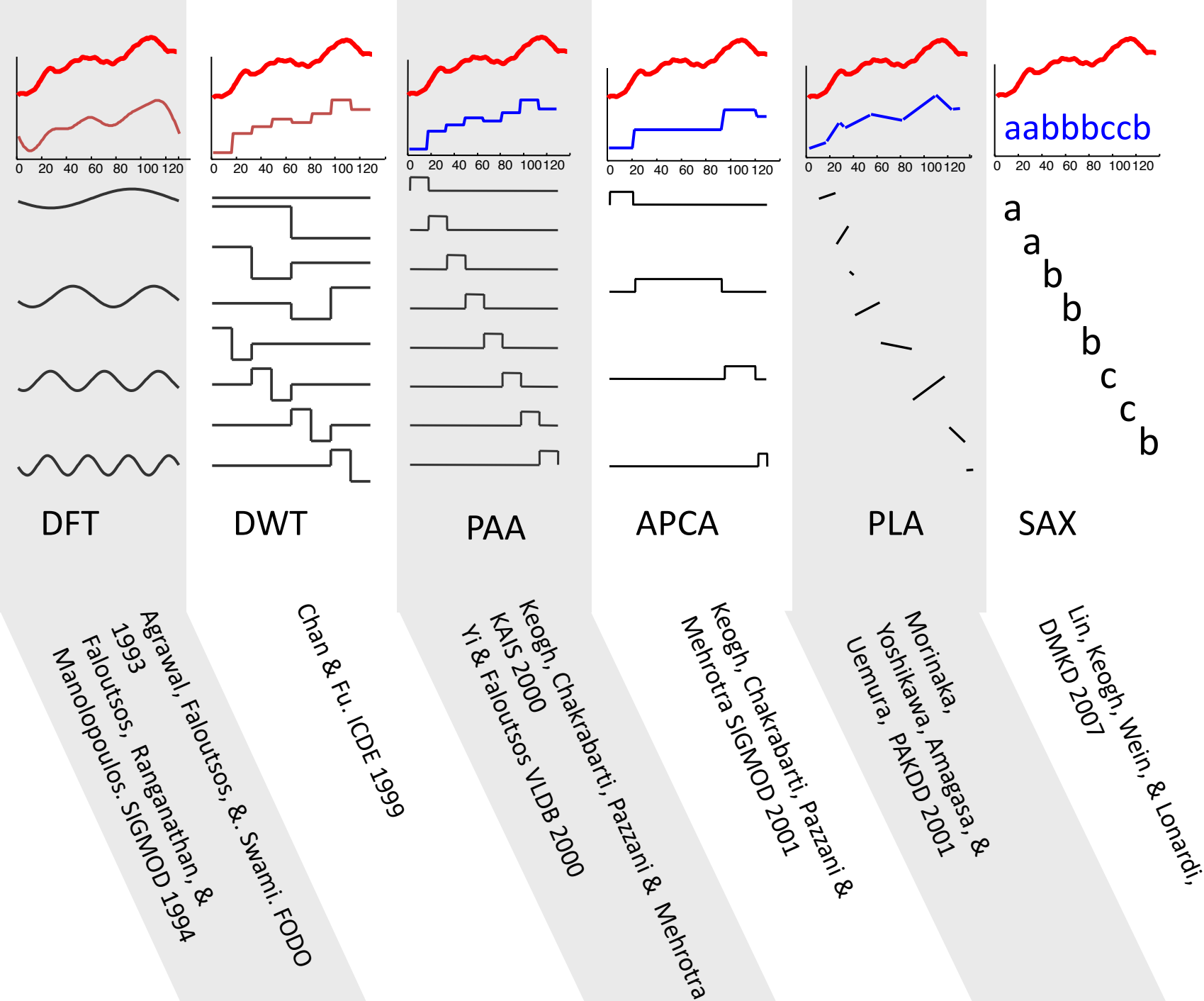


Allen's temporal logic

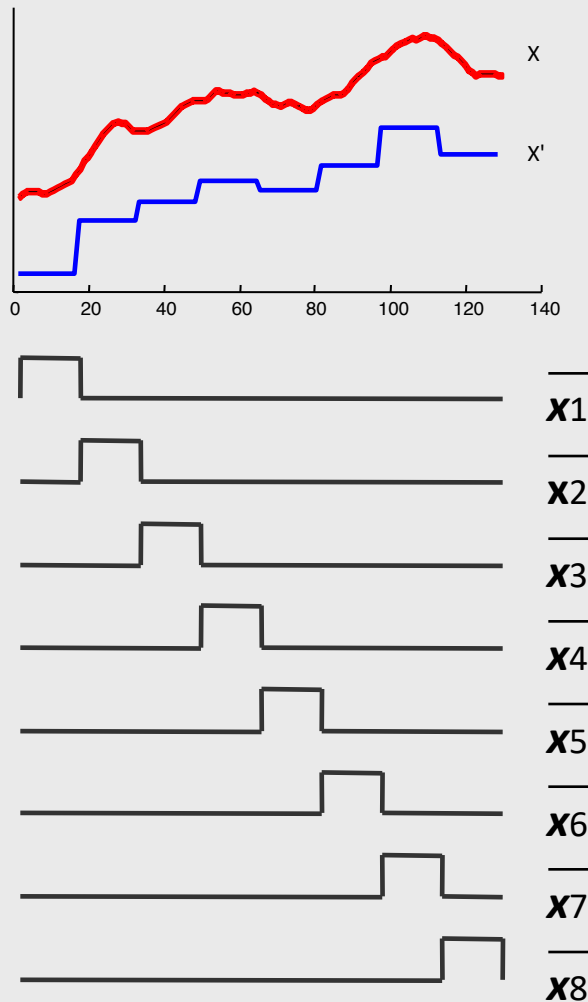
What is a temporal feature?

a sequence of “**temporal relations**” between two or more event intervals

What are the types of “temporal relations”?



Piecewise Aggregate Approximation (PAA)



Basic Idea: Represent the original time series of length n as a sequence of box basis functions, each box being of the same length resulting in a new time series of length $w \ll n$

Computation:

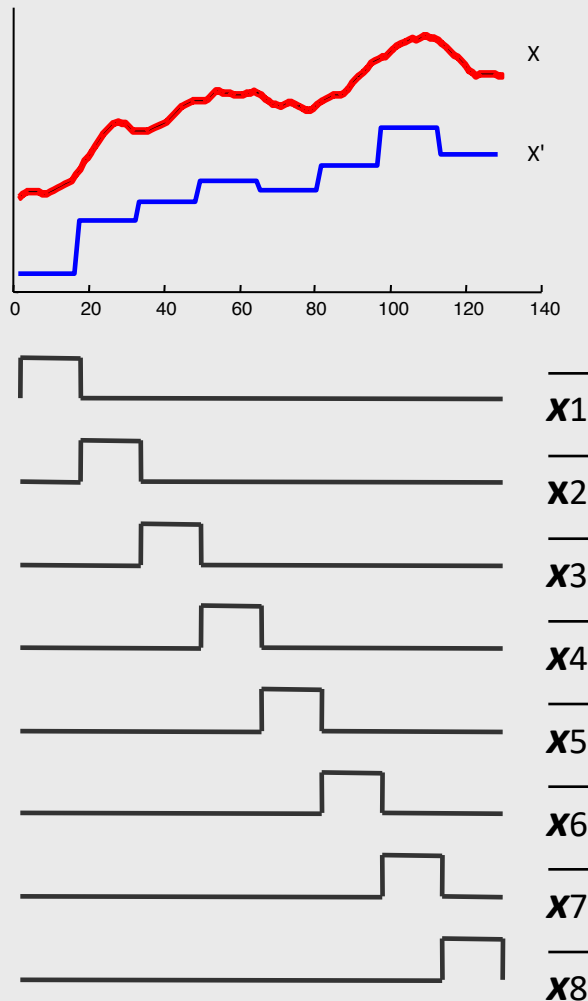
- X : original time series of length n
- It can be represented in the w -dimensional space as:

$$\bar{x}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} x_j$$

Keogh, Chakrabarti, Pazzani & Mehrotra, KAIS (2000)

Byoung-Kee Yi, Christos Faloutsos, VLDB (2000)

Piecewise Aggregate Approximation (PAA)



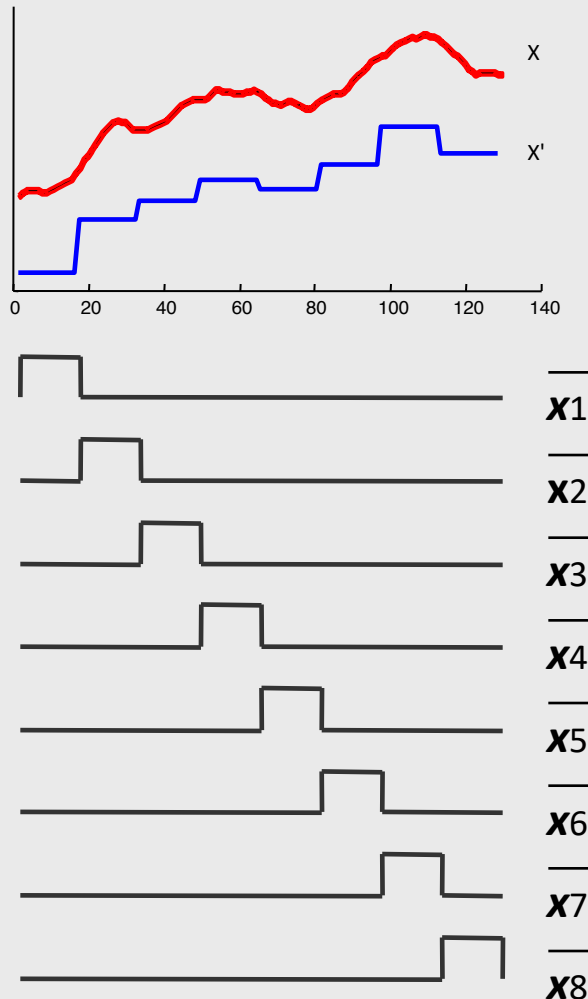
Example

Let $X = [1 \ 3 \ -1 \ 4 \ 4 \ 4 \ 5 \ 3 \ 7]$

- X can be mapped from its original dimension $n = 9$ to a lower dimension, e.g., $w = 3$, as follows:

$$\begin{array}{ccccccc} [1 & 3 & -1 & 4 & 4 & 4 & 5 & 3 & 7] \\ \hline \downarrow & & \downarrow & & \downarrow & & & & \\ [& 1 & & 4 & & 5 & & &] \end{array}$$

Piecewise Aggregate Approximation (PAA)



Pros and Cons of PAA as a time series representation

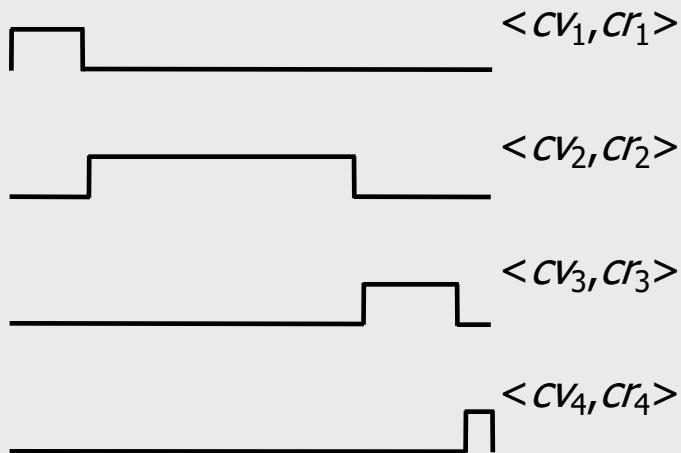
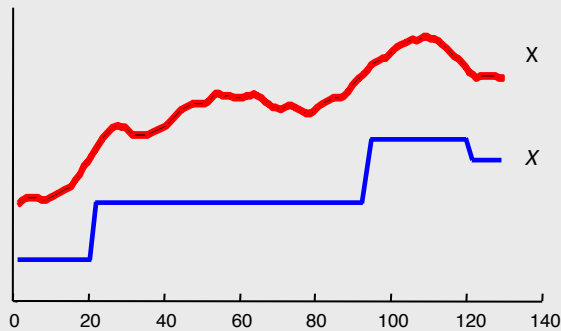
Pros:

- Fast to calculate
- Can support any Minkowski metric
- Supports non-Euclidean measures
- Simple and Intuitive!

Cons:

- It assumes a fixed-length non-overlapping summary window over the time series

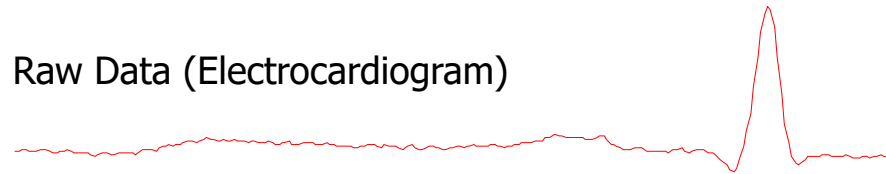
Adaptive Piecewise Constant Approximation (APCA)



Generalize PAA to allow the piecewise constant segments to have arbitrary lengths

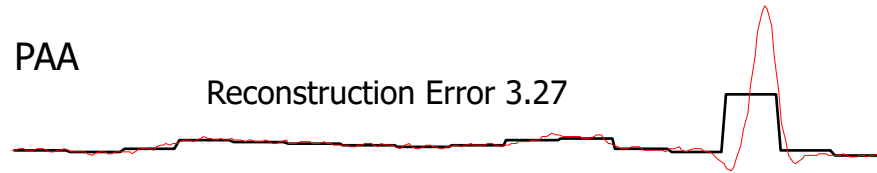
We need *two coefficients* to represent each segment: its value (cv_i) and its length (cr_i)

Raw Data (Electrocardiogram)



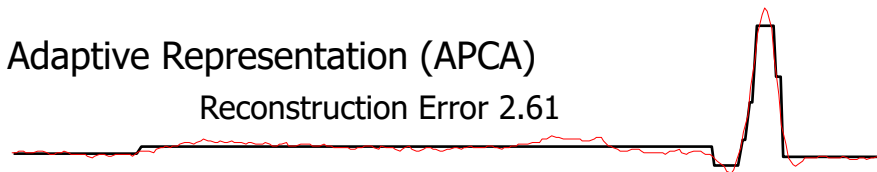
PAA

Reconstruction Error 3.27



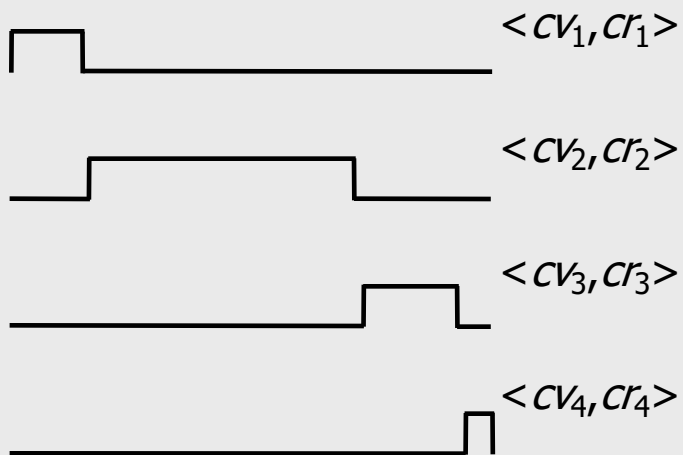
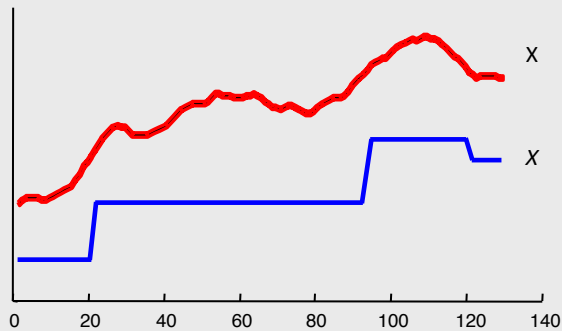
Adaptive Representation (APCA)

Reconstruction Error 2.61



Intuition: many signals have little detail in some places and high detail in other places; APCA can *adaptively* fit itself to the data

Adaptive Piecewise Constant Approximation (APCA)



Pros and Cons of APCA as a time series representation

Pros:

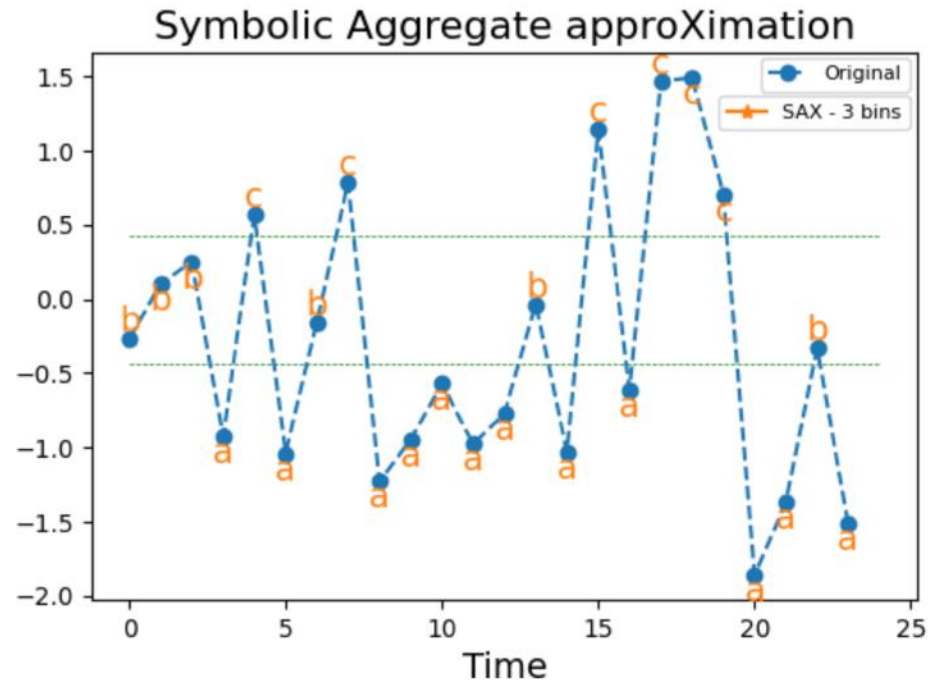
- Fast to calculate $O(n)$
- Supports non Euclidean measures
- Supports fast exact queries, and even faster approximate queries on the same data structure

Cons:

- Performance depends highly on the segmentation technique used

Symbolic ApproXimation (SAX)

- Similar in principle to PAA
 - uses **segments** to represent each time series
- Represents segments with **symbols** (rather than real numbers)
 - **smaller** memory footprint



Symbolic ApproXimation (SAX)

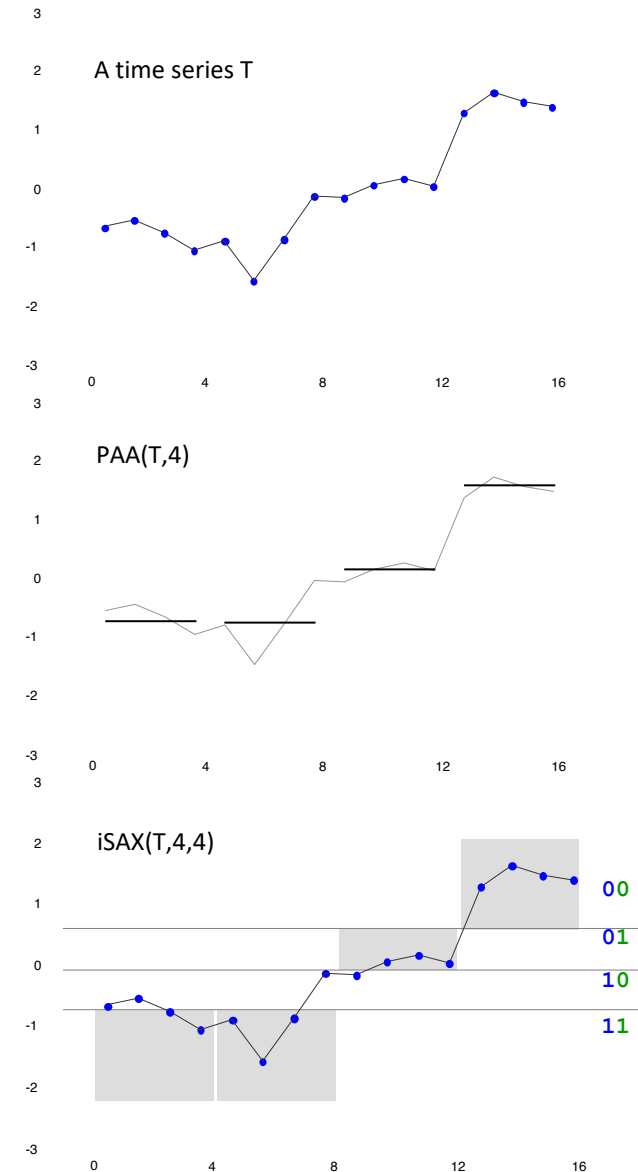
Represent the **time series** T of length n with w **segments** using PAA

- T typically normalized to $\mu = 0, \sigma = 1$
- PAA(T, w): $\bar{T} = \bar{t}_1, \dots, \bar{t}_w$

$$\text{with } \bar{t}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} T_j$$

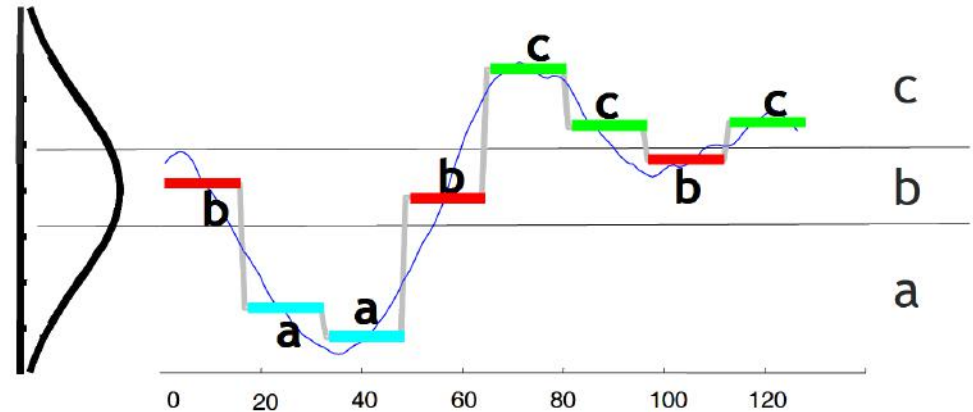
Discretize into a vector of **symbols**

- **breakpoints** map to small **alphabet** α of symbols



Symbolic ApproXimation (SAX)

- **Assumption:** normalized time series follow a Gaussian distribution
- Determine the **breakpoints** that will produce k equal-sized areas under the **Gaussian curve**
- Use a **lookup table** with **breakpoints** that divide a Gaussian distribution to an arbitrary number of equi-probable regions



$\beta_i \backslash a$	3	4	5	6	7	8	9	10
β_1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
β_2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
β_3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
β_4			0.84	0.43	0.18	0	-0.14	-0.25
β_5				0.97	0.57	0.32	0.14	0
β_6					1.07	0.67	0.43	0.25
β_7						1.15	0.76	0.52
β_8							1.22	0.84
β_9								1.28

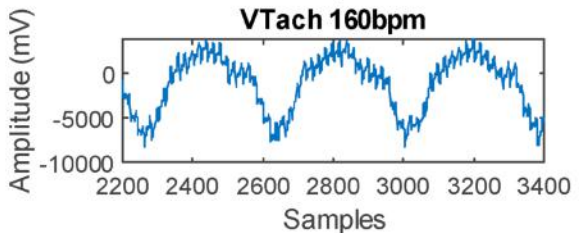
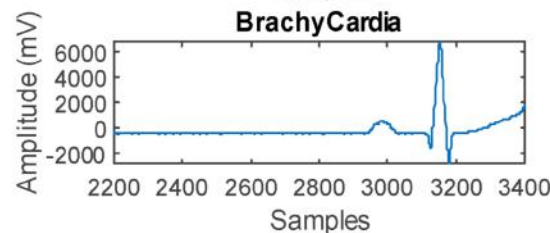
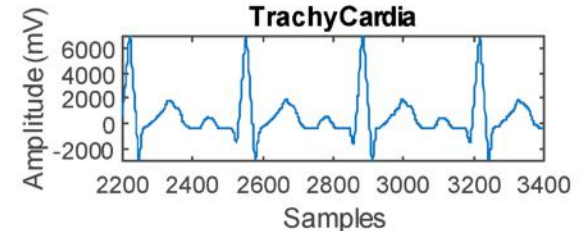
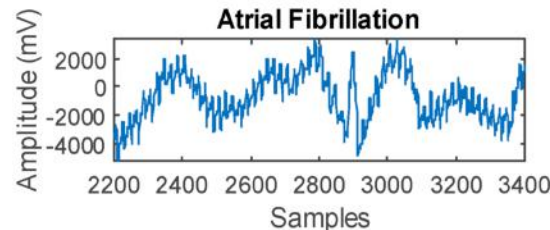
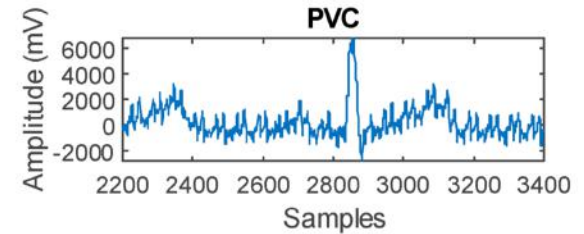
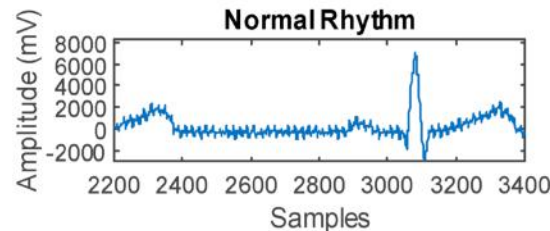
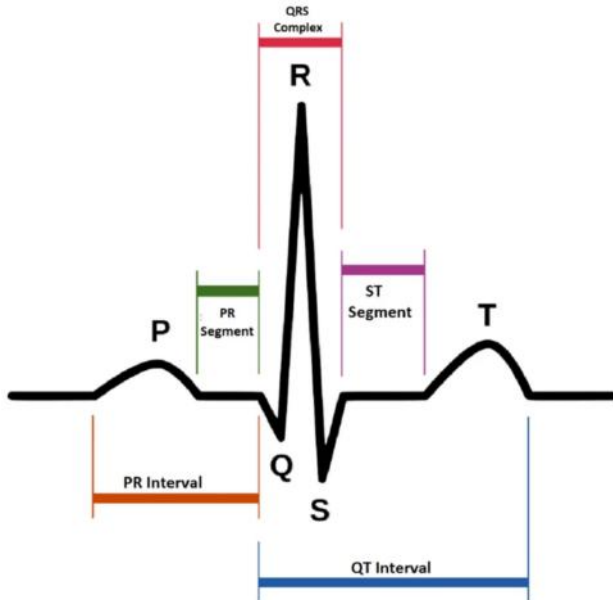
Outline

- Data series
- Electronic Health Records
- Time series summarization
- Time series classification
- Time series counterfactuals
- Challenges and future directions

Time series classification



abnormal,
atrial fibrillation

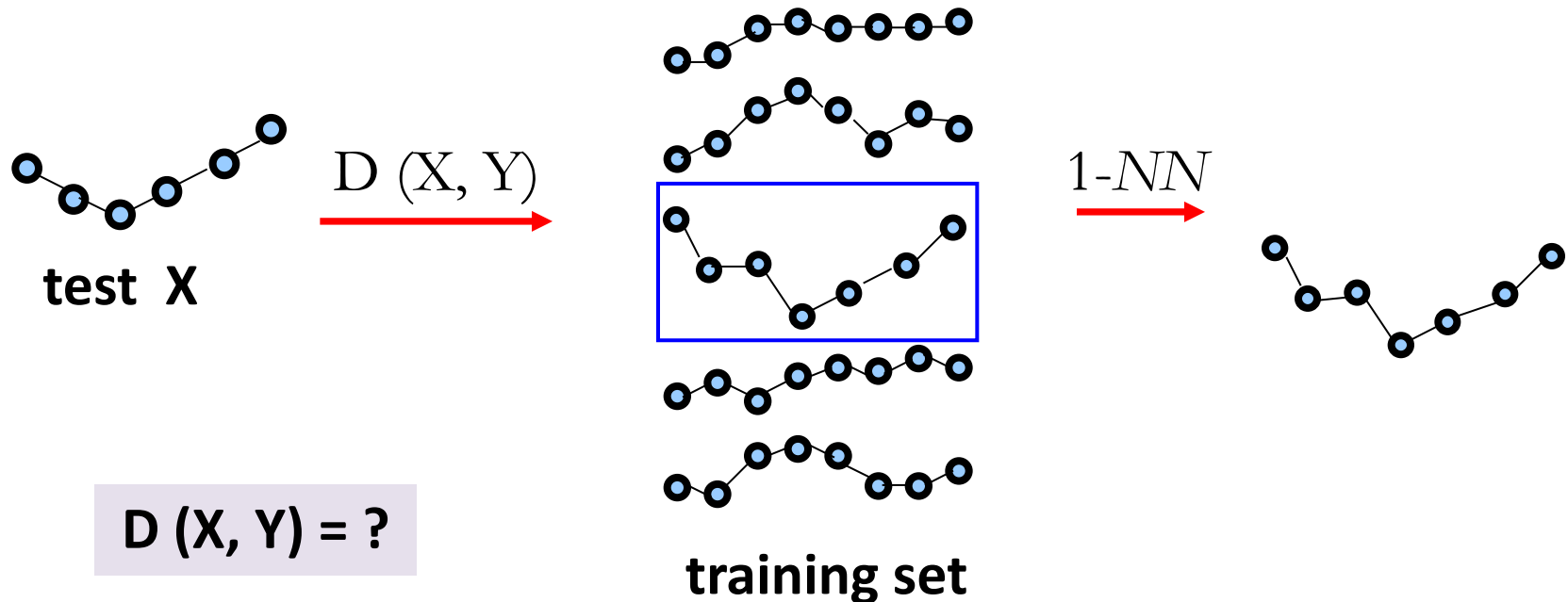


Many time series classifiers

- Distance-based
- Feature-based
- Deep learning-based

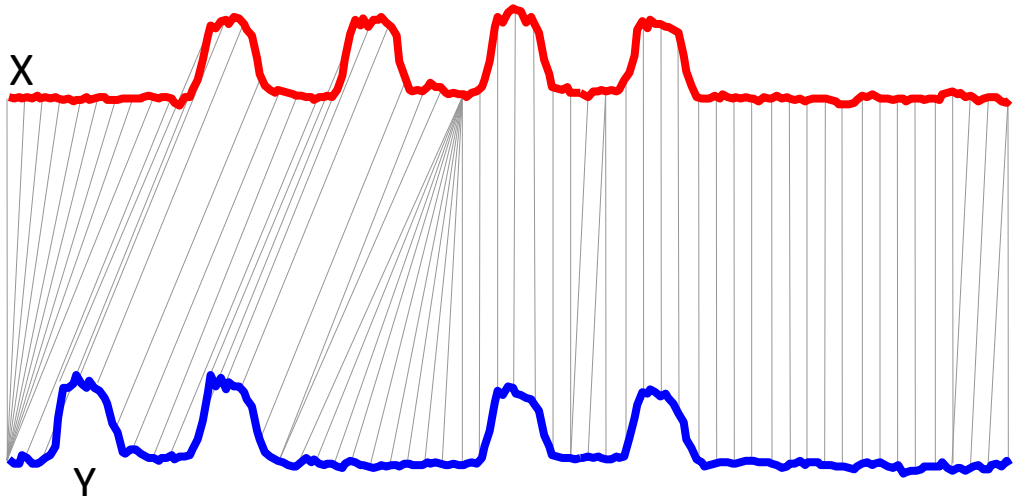
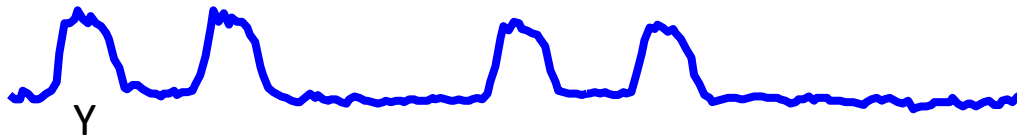
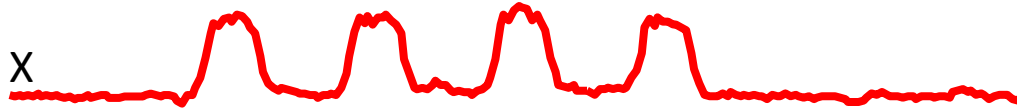
k-NN time series classification

- Given a **time series training set** and a **test time series X**
- Find the **best match** of X in the database
- Assign the **class** of the **1-NN** to X



Limitations of Euclidean Distance

figures taken from Eamonn Keogh, University of California, Riverside



Euclidean Distance

Sequences are aligned “one to one”.

$$D(X, Y) \equiv \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

“Warped” Time Axis

Nonlinear alignments are possible.

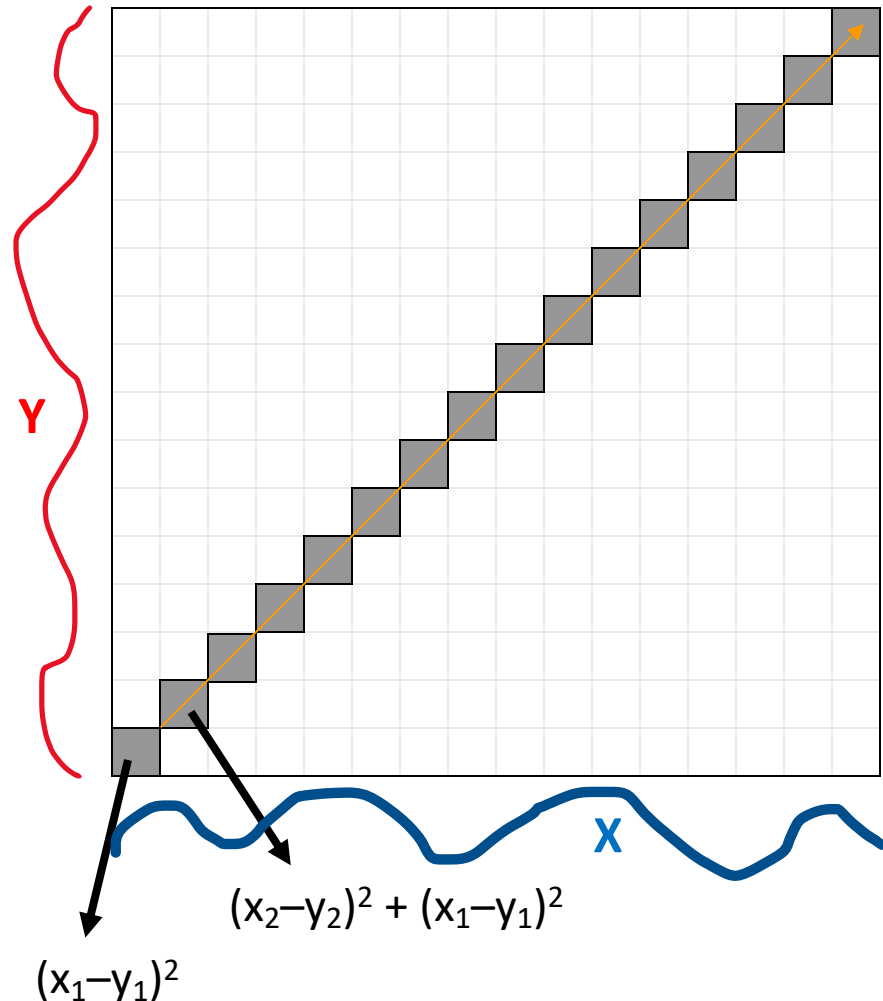
Computing DTW

- DTW is computed by **dynamic programming** (DP)
- Given two sequences
 - $X = \{x_1, x_2, \dots, x_n\}$
 - $Y = \{y_1, y_2, \dots, y_m\}$
- Use an $n \times m$ **DP matrix** f to store the scores

$$D_{dtw}(X, Y) = f(n, m)$$

$$f(i, j) = \|x_i - y_j\| + \min \begin{cases} f(i, j-1) \\ f(i-1, j) \\ f(i-1, j-1) \end{cases}$$

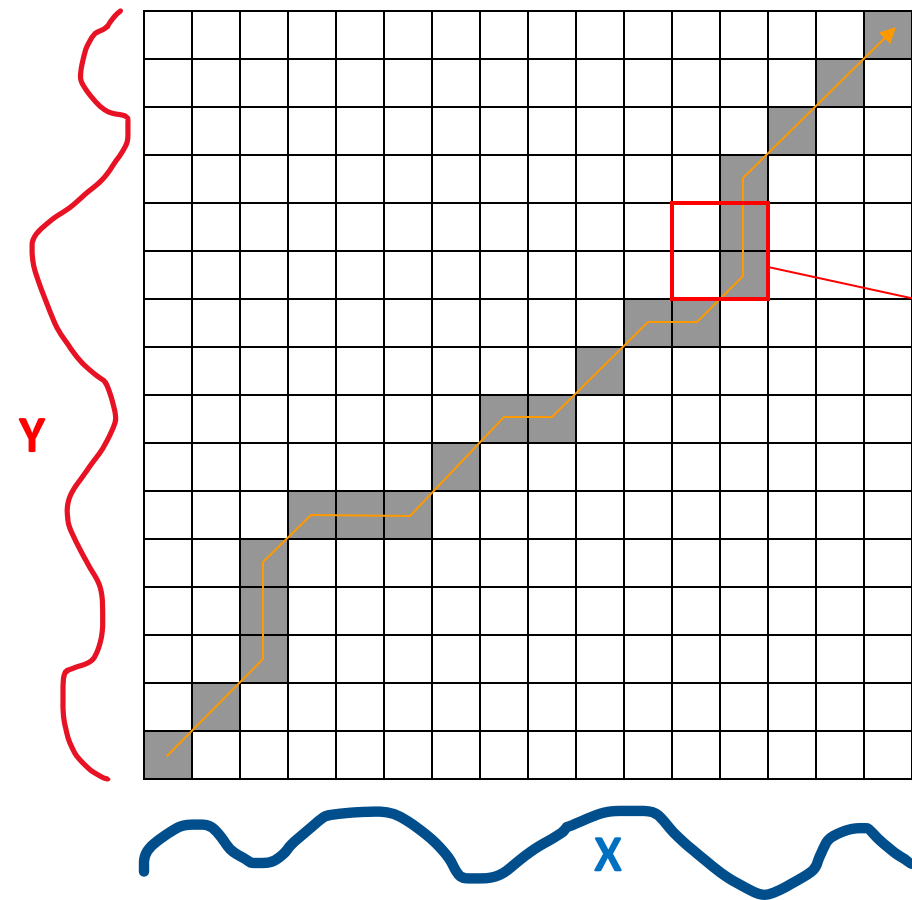
Computing DTW



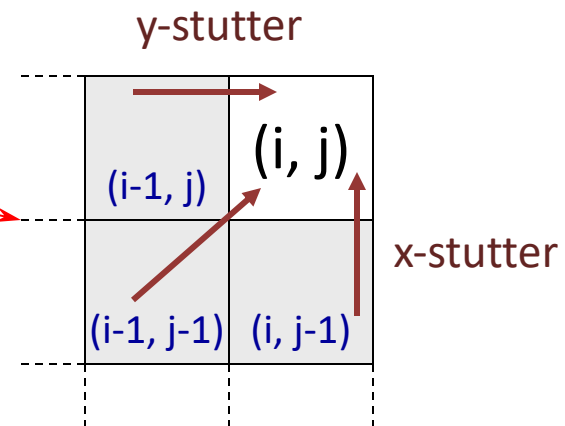
- Each cell (i, j) is a pair of indices
- Cell values are computed using some norm, e.g., $p=2$
- **Euclidean path:**
 - $i = j$
 - Ignores off-diagonal cells

$$f(i, j) = \|x_i - y_j\| + \min \begin{cases} f(i, j-1) \\ f(i-1, j) \\ \boxed{f(i-1, j-1)} \end{cases}$$

Computing DTW



- DTW allows any path
- Examine all paths:

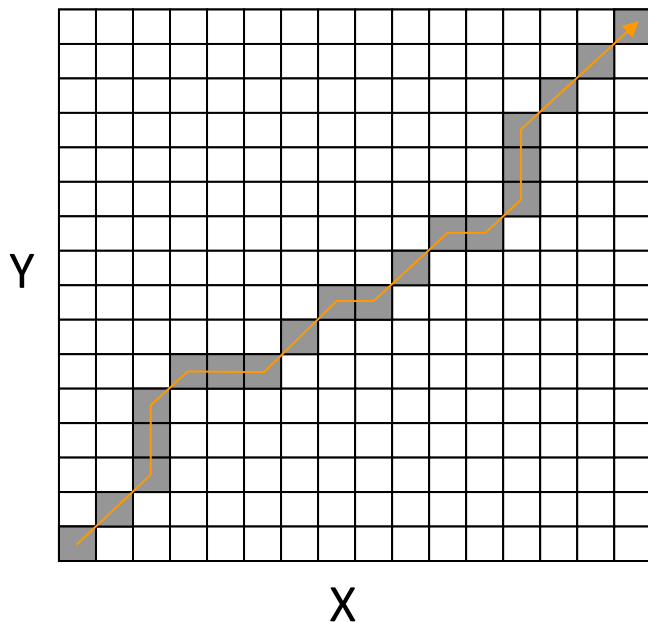


- Standard dynamic programming to fill in the DP table
- The top-right cell contains final result

$$f(i, j) = \|x_i - y_j\| + \min \begin{cases} f(i, j-1) \\ f(i-1, j) \\ f(i-1, j-1) \end{cases}$$

Properties of DTW

- **Warping path W :**
 - set of grid cells in the time warping matrix
- **DTW finds an optimum warping path W :**
 - the path with the smallest matching score



Properties of a DTW legal path

I. Boundary conditions

$$W_1=(1,1) \text{ and } W_K=(n,m)$$

II. Continuity

Given $W_k = (a, b)$, then

$$W_{k-1} = (c, d), \text{ where } a-c \leq 1, b-d \leq 1$$

III. Monotonicity

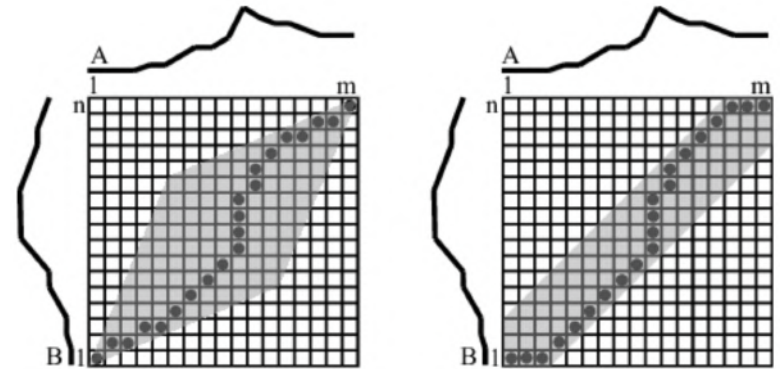
Given $W_k = (a, b)$, then

$$W_{k-1} = (c, d), \text{ where } a-c \geq 0, b-d \geq 0$$

Pros and cons

- **Advantages**

- Query and target lengths **may not** be of equal length
- Can be constrained
- Can tolerate:
 - time shifts
 - sequences out of phase
 - scaling in the y-axis (how?)



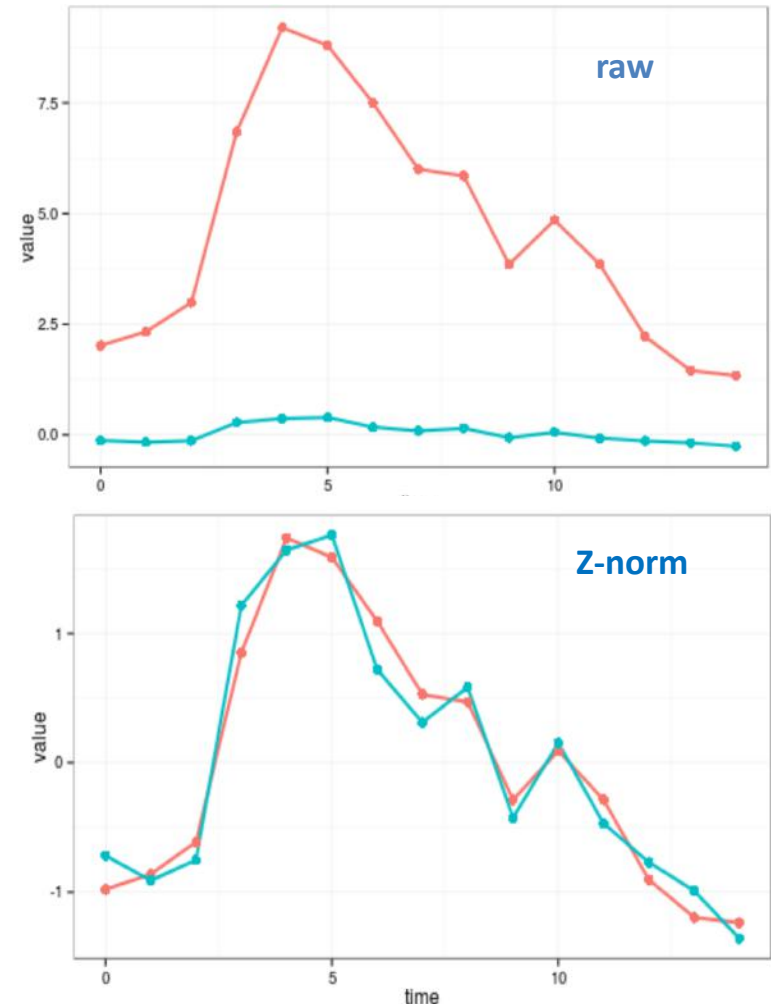
- **Disadvantages**

- Computational complexity: $O(n \cdot m)$ for unconstrained and $O(r \cdot n)$ for constrained
- May not be able to handle missing or noisy values

Z-normalization

- Needed when interested in *detecting trends* and not absolute values
- For streaming data:
 - each subsequence of interest should be z-normalized before being compared to a z-normalized query
 - otherwise the trends are lost
- Z-normalization guarantees:
 - offset invariance
 - scale/amplitude invariance

$$x'_i = \frac{x_i - \mu}{\sigma}, \text{ where } i \in \mathbb{N}$$

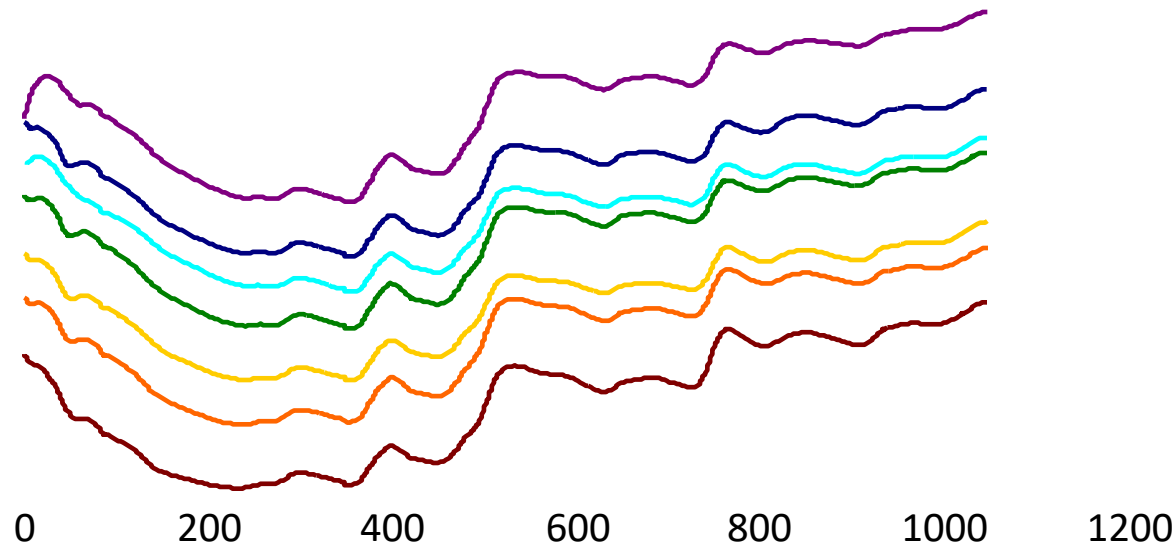


Other time series distance measures

- **DDTW**: Derivative DTW
- **WDTW**: Weighted DTW
- **LCSS**: Longest Common Subsequence
- **MSM**: Move-Split-Merge
- **ERP**: Edit Distance with Real Penalty
- **TWE**: Time Warp Edit

The k -NN classifier for time series

- Given **seven** time series **classes**



- k-NN
 - Use any of the **existing distance measures**, e.g., Euclidean, DTW, or variants
 - **Cons:** unable to identify smaller patterns or shapes that are class distinctive

Many time series classifiers

- Distance-based
- Feature-based
- Deep learning-based

How about feature-based classification?

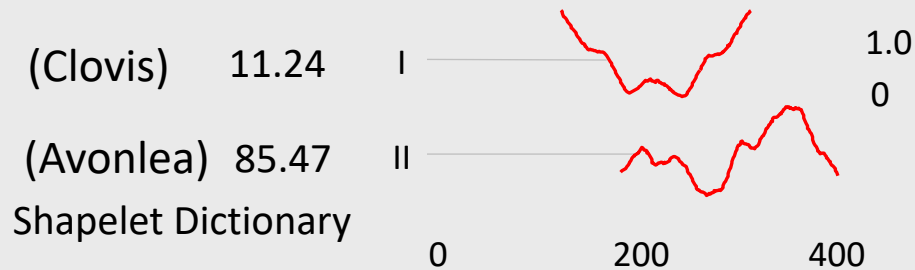
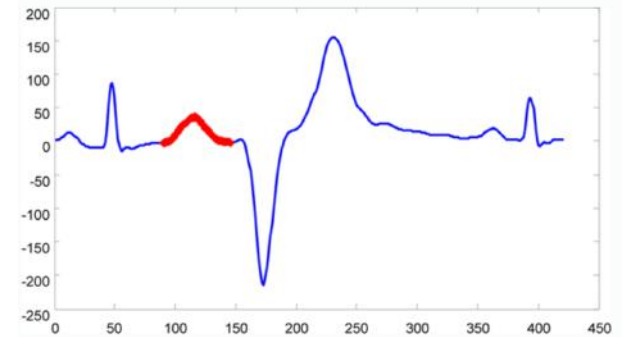
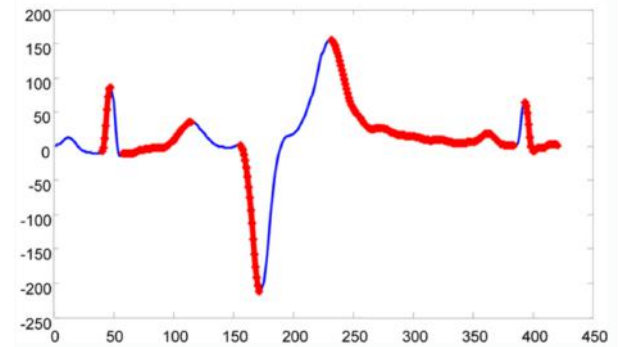
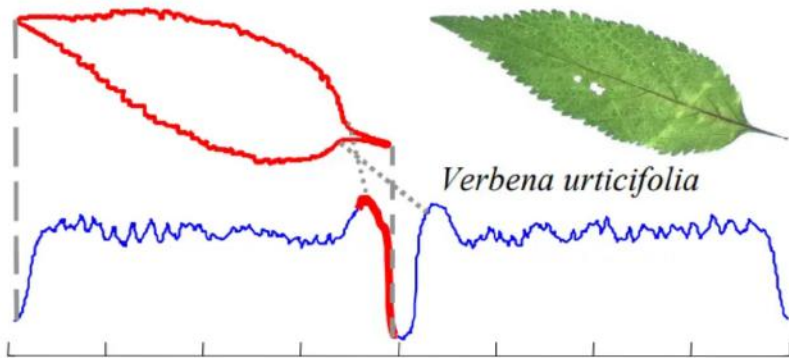
- Use **shapelets** as “attributes” or “features” for splitting a node in the decision tree

shapelet

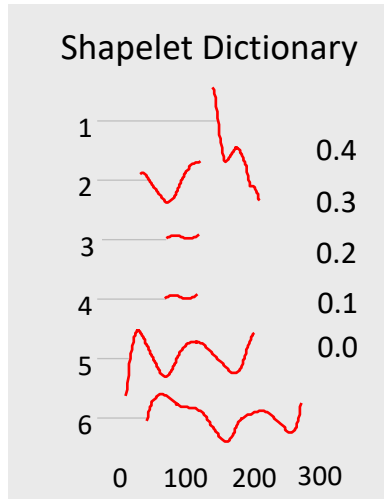


- **Shapelets:**
 - time series subsequence
 - *maximally representative* of a class
 - *discriminative* from other classes

Examples

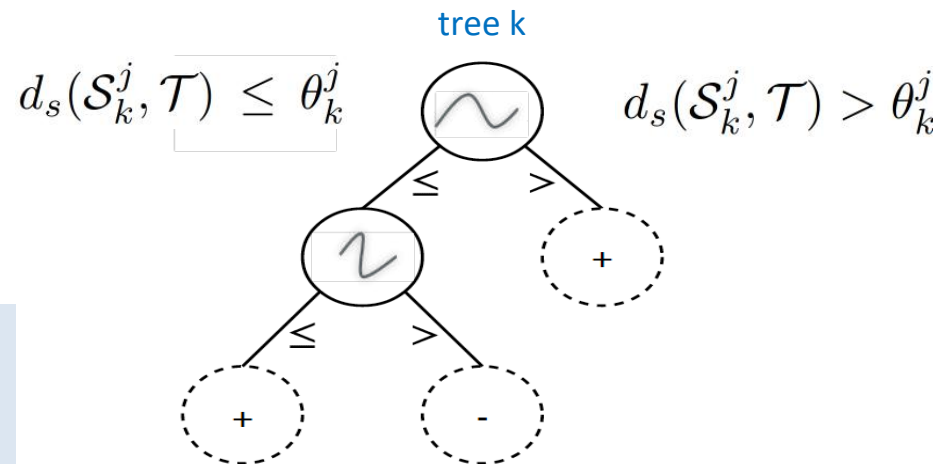


The Shapelet Tree classifier

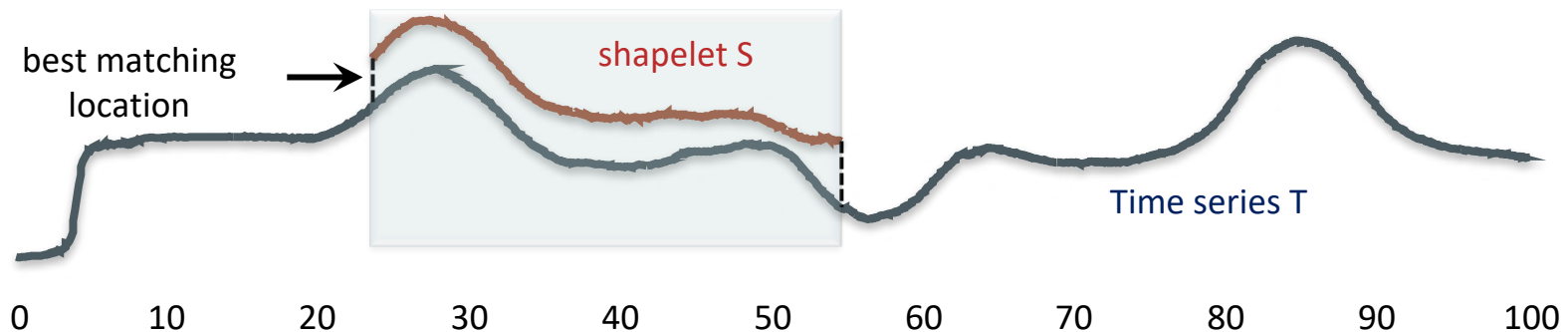


Each tree defines several root-leaf paths

$$p_{k,j} = \{(x_1 \underset{\geq}{\leq} \theta_1), (x_2 \underset{\geq}{\leq} \theta_2), \dots, (x_n \underset{\geq}{\leq} \theta_n)\}$$



non-leaf node condition: Euclidean distance, lowest scoring subsequence match of S in T



Alternative approaches

- Transformations + k-NN

- Improved subsequence searching and matching, using online normalization, early abandoning, and re-ordering
- Dimensionality reduction using SAX

- Shapelet-based features

- Select the top k most informative shapelets as features
- Learn any suitable classifier (e.g., SVM, Random Forest) using the transformed dataset

- Synthetic shapelet generation

- Initialize using K-means clustering
- Learn synthetic Shapelets

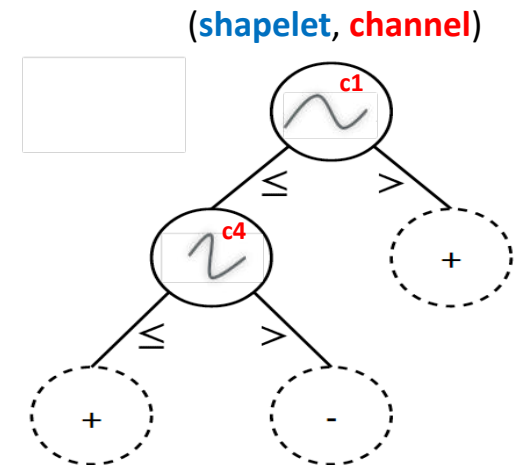
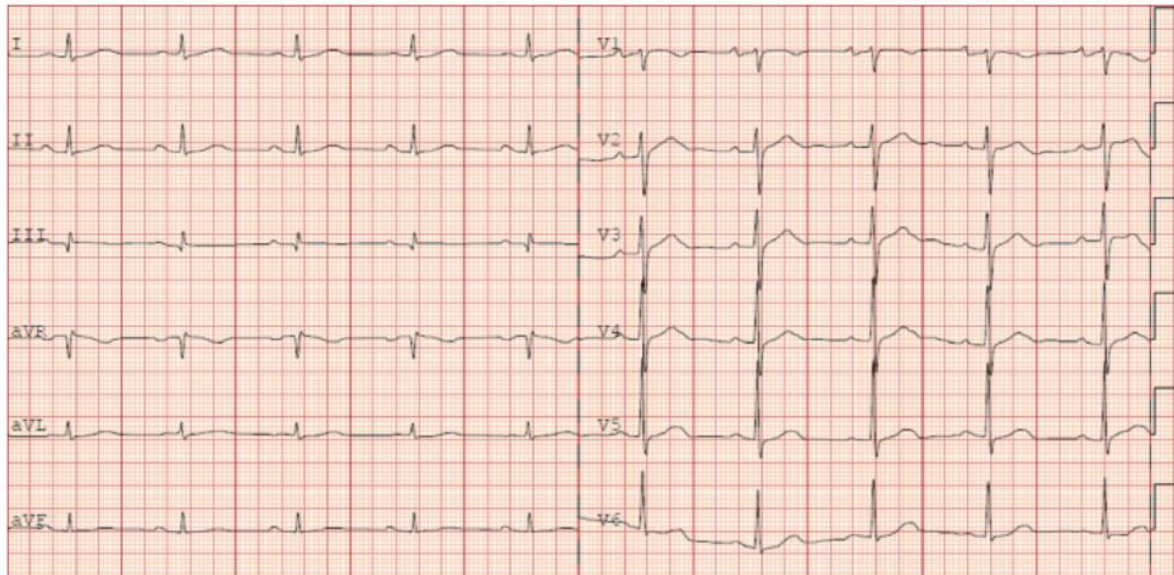
	s_1	s_2	$\dots$	s_k
d_1	0.3	3.3	$\dots$	0.1
d_2	0.2	3.2	$\dots$	3.8
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
d_n	3.1	0.9	$\dots$	9.6

Random Shapelet Forest (RSF)

- A **tree-structured ensemble** based on the classical **Random Forest** and the **Shapelet Tree classifier**
- **Learn:** build T random shapelet trees
 - each tree is built from a **random sample** (with replacement) of time series in the training set
 - Inspect r random shapelets at each node
- **Predict:**
 - Let each tree $t_1, \dots, t_T$ vote for a **class label**

Generalized Random Shapelet Forest (gRSF)

- A **generalization** of RSF for **multivariate** time series classification
- T random shapelet trees are built
 - each tree is built from a random sample (with replacement) of **time series channels** in the training set (channels are recorded in the decision nodes)
 - inspect r random shapelets at each node



Other feature-based classifiers

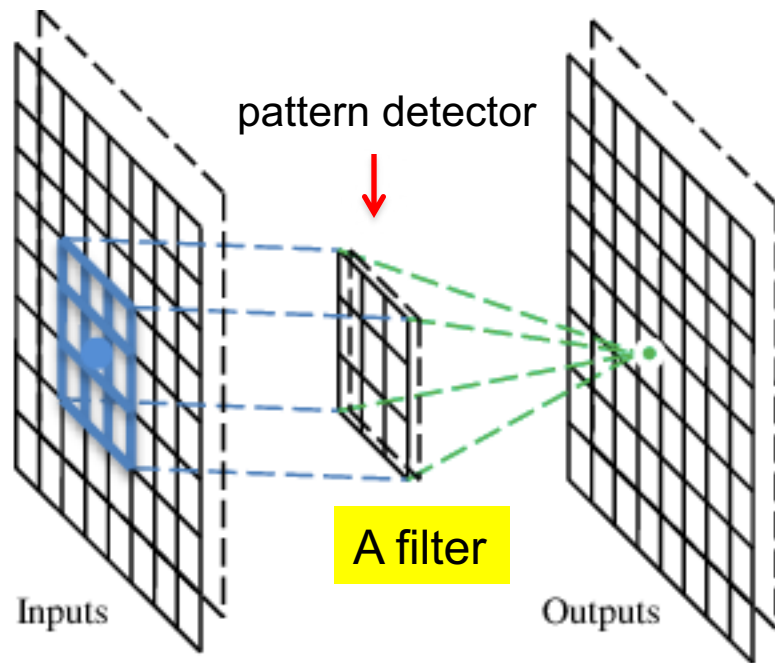
- **STC**: Shapelet Transform
- **BOSS**: Bag-of- SFA-Symbols
- **WEASEL**: Word eXtrAction for time SEries cLassification
- **MrSEQ**: Multiple Representation Sequence Learner

Many time series classifiers

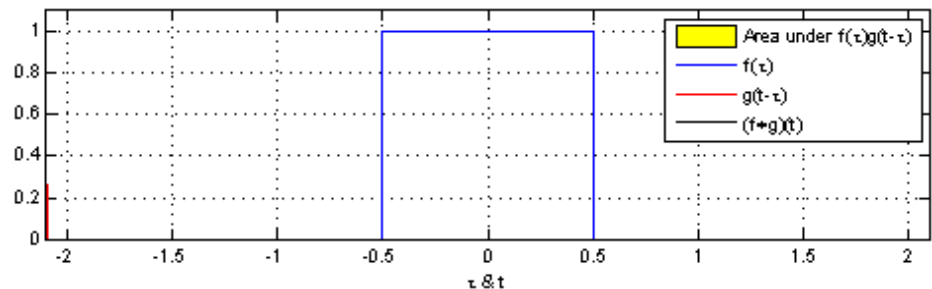
- Distance-based
- Feature-based
- Deep learning-based

Convolution-based classifiers

- They consist of **convolutional layers** (and some other layers)
- A convolutional layer has a **number of filters** that perform the operation of **convolution**



$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$$



The **sum** of an infinite number of copies of **$g()$** (**impulse response or input**) each **shifted** by a slightly different **time delay** (**τ**) and **scaled** according to the value of the input signal at the value of **t** that corresponds to delay **$f(\tau)$**

Example of a convolution

- Virus outbreak
- Treatment plan: 3 dosages in 1 day
- # of incoming patients per day: M:1, T:2, W:3, T:4, F:5

$$\begin{array}{rcl} \text{Plan} & * & \text{Patients} & = & \text{Daily Usage} \\ [3] & * & [1 \ 2 \ 3 \ 4 \ 5] & = & [3 \ 6 \ 9 \ 12 \ 15] \end{array}$$

- Suppose the disease mutates and requires a multi-day treatment, i.e., day 1: 3 dosages, day 2: 2 dosages, day 3: 1 dosage

Plan: [3 2 1]

Example of a convolution



M	T	W	T	F
1	2	3	4	5

reverse



$g(-\tau)$

5	4	3	2	1
---	---	---	---	---



D1	D2	D3
3	2	1

$f(\tau)$

3	2	1
---	---	---



5	4	3	2	1			
					3	2	1

Monday: $1 \times 3 = 3$

Daily total:

3

Example of a convolution



M	T	W	T	F
1	2	3	4	5

reverse



$g(-\tau)$

5	4	3	2	1
---	---	---	---	---



D1	D2	D3
3	2	1

$f(\tau)$

3	2	1
---	---	---



5	4	3	2	1	
			3	2	1



Monday: $1 \times 3 = 3$

Tuesday: $2 \times 3 + 1 \times 2 = 8$

Daily total:

3	8
---	---

Example of a convolution



M	T	W	T	F
1	2	3	4	5

reverse



$g(-\tau)$

5	4	3	2	1
---	---	---	---	---



D1	D2	D3
3	2	1

$f(\tau)$

3	2	1
---	---	---



5	4	3	2	1
		3	2	1
		3	2	1



Monday: $1 \times 3 = 3$

Tuesday: $2 \times 3 + 1 \times 2 = 8$

Wednesday: $3 \times 3 + 2 \times 2 + 1 \times 1 = 14$

Daily total:

3	8	14
---	---	----

Example of a convolution



M	T	W	T	F
1	2	3	4	5

reverse



$g(-\tau)$

5	4	3	2	1
---	---	---	---	---



D1	D2	D3
3	2	1

$f(\tau)$

3	2	1
---	---	---



5	4	3	2	1
	3	2	1	



Monday: $1 \times 3 = 3$

Tuesday: $2 \times 3 + 1 \times 2 = 8$

Wednesday: $3 \times 3 + 2 \times 2 + 1 \times 1 = 14$

Thursday: $4 \times 3 + 3 \times 2 + 2 \times 1 = 20$

Daily total:

3	8	14	20
---	---	----	----

Example of a convolution



M	T	W	T	F
1	2	3	4	5

reverse



$g(-\tau)$

5	4	3	2	1
---	---	---	---	---



D1	D2	D3
3	2	1

$f(\tau)$

3	2	1
---	---	---



5	4	3	2	1
3	2	1		



Daily total:

3	8	14	20	26
---	---	----	----	----

Monday: $1 \times 3 = 3$

Tuesday: $2 \times 3 + 1 \times 2 = 8$

Wednesday: $3 \times 3 + 2 \times 2 + 1 \times 1 = 14$

Thursday: $4 \times 3 + 3 \times 2 + 2 \times 1 = 20$

Friday: $5 \times 3 + 4 \times 2 + 3 \times 1 = 26$

Example of a convolution



M	T	W	T	F
1	2	3	4	5

reverse



$g(-\tau)$				
5	4	3	2	1



D1	D2	D3
3	2	1

$f(\tau)$

3	2	1
---	---	---



	5	4	3	2	1
3	2	1			



Daily total:

3	8	14	20	26	14
---	---	----	----	----	----

Monday: $1 \times 3 = 3$

Tuesday: $2 \times 3 + 1 \times 2 = 8$

Wednesday: $3 \times 3 + 2 \times 2 + 1 \times 1 = 14$

Thursday: $4 \times 3 + 3 \times 2 + 2 \times 1 = 20$

Friday: $5 \times 3 + 4 \times 2 + 3 \times 1 = 26$

Saturday: $5 \times 2 + 4 \times 1 = 14$

Example of a convolution



M	T	W	T	F
1	2	3	4	5

reverse



$g(-\tau)$

5	4	3	2	1
---	---	---	---	---



D1	D2	D3
3	2	1

$f(\tau)$

3	2	1
---	---	---



		5	4	3	2	1
3	2	1				



Daily total:

3	8	14	20	26	14	5
---	---	----	----	----	----	---

Monday: $1 \times 3 = 3$

Tuesday: $2 \times 3 + 1 \times 2 = 8$

Wednesday: $3 \times 3 + 2 \times 2 + 1 \times 1 = 14$

Thursday: $4 \times 3 + 3 \times 2 + 2 \times 1 = 20$

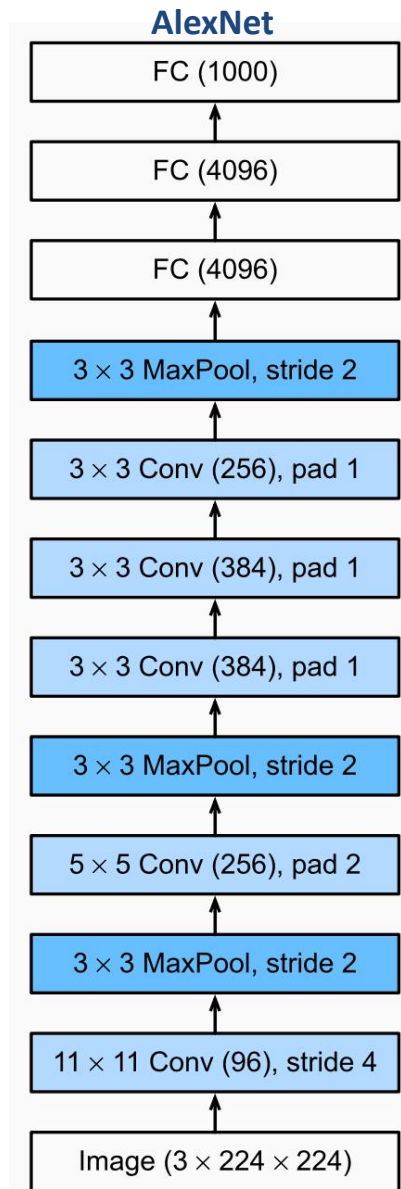
Friday: $5 \times 3 + 4 \times 2 + 3 \times 1 = 26$

Saturday: $5 \times 2 + 4 \times 1 = 14$

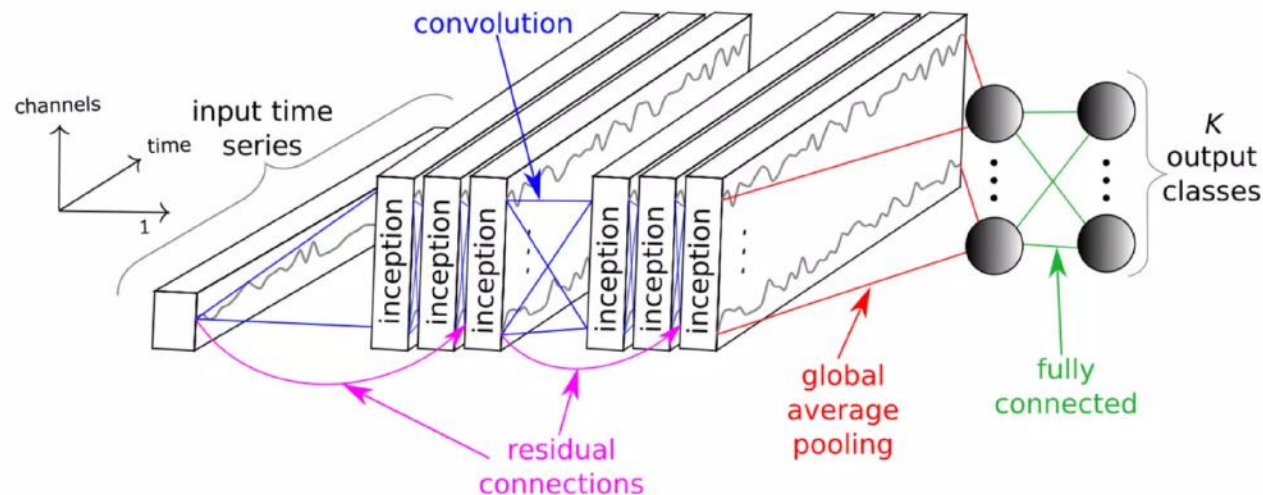
Sunday: $5 \times 1 = 5$

Inception Time [Fawaz 2020]

- The equivalent of **AlexNet** for time series
- An ensemble of **five deep learning models**
 - each created by **cascading** multiple **inception modules**
 - each having exactly the same architecture but with different **randomly** initialized weight values
- **Core idea of an inception module:**
 - apply **multiple filters** simultaneously to an input time series
 - includes filters of **varying lengths** allowing the network to automatically extract relevant features from both **long** and **short** time series

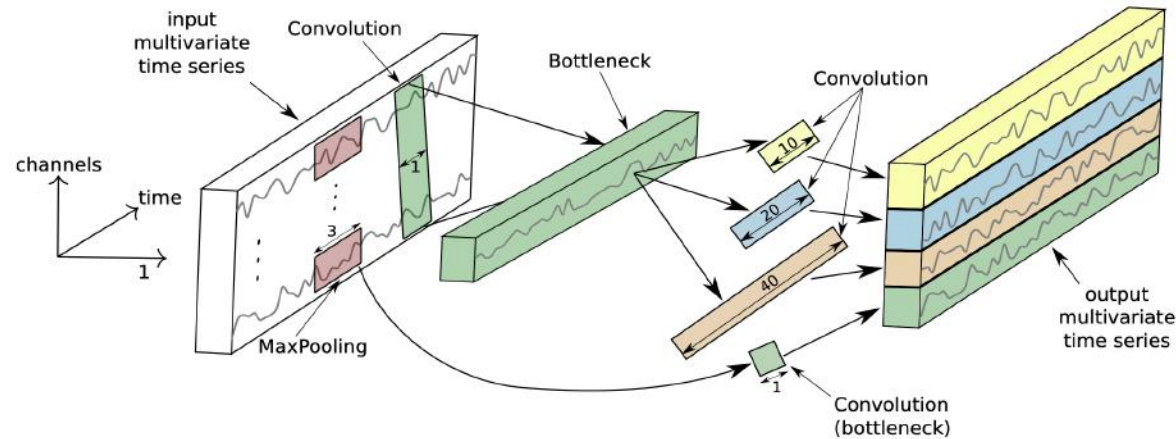


Inception Time [Fawaz 2020]



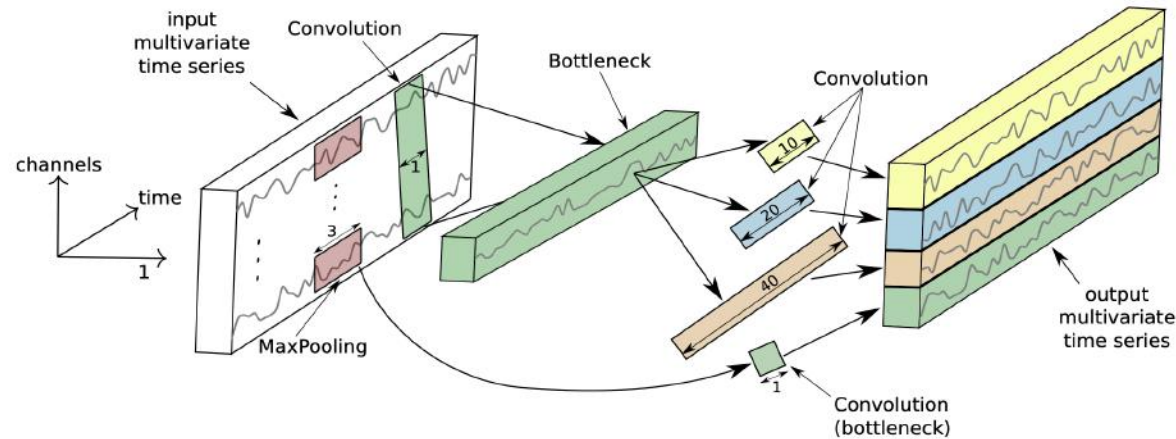
- Two residual blocks, each comprising three inception modules
- Each residual block's input is transferred via a linear connection to the next block
- **Global Average Pooling (GAP) layer:** averages the output multivariate time series over the *whole time dimension*
- **Fully-connected softmax layer:** number of neurons equals the number of classes

Inception Time [Fawaz 2020]



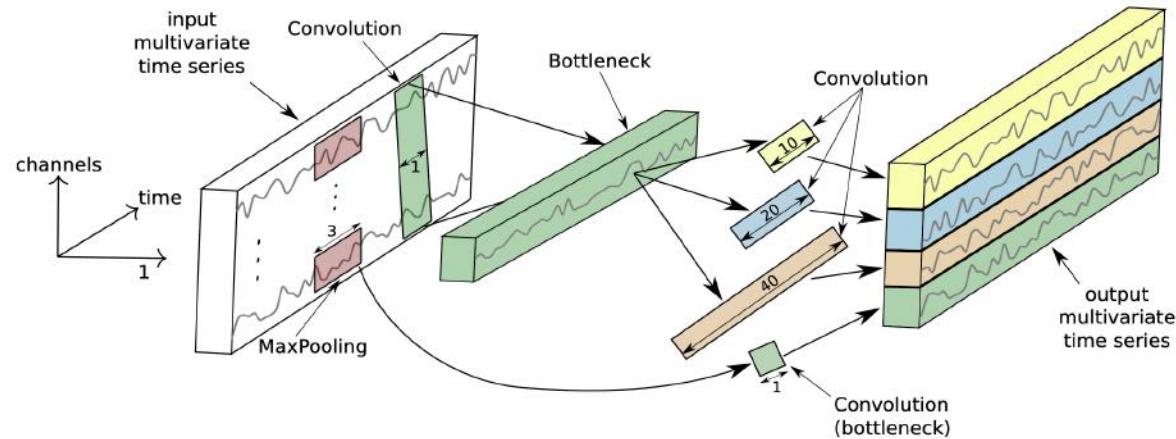
- The **Inception module** contains **four layers**:
 - Bottleneck Layer
 - A set of parallel Convolutional Layers with varying filter sizes
 - MaxPooling Layer
 - Depth Concatenation Layer
- The network can extract features of **multiple resolutions** using these filters

Inception Time [Fawaz 2020]



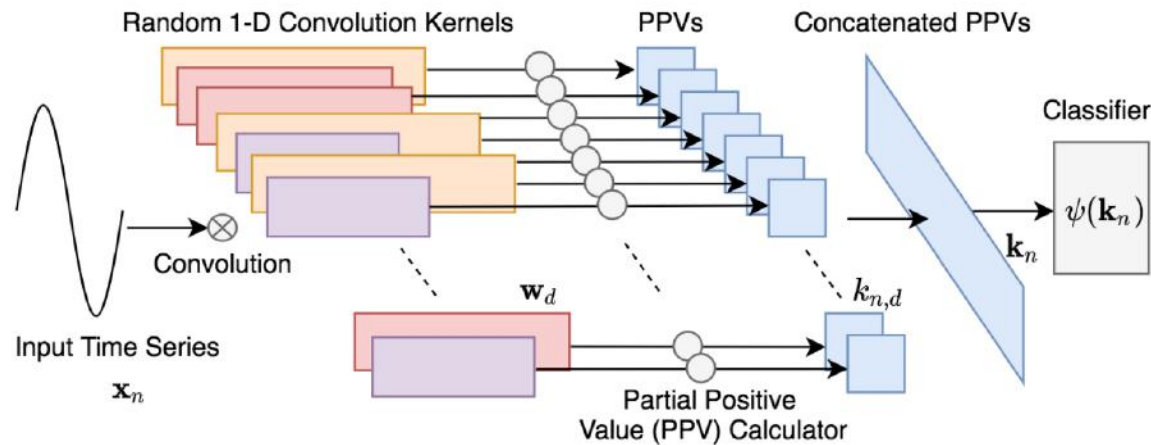
- **Bootleneck Layer:**
 - performs an operation of sliding m filters of length 1 with a stride of 1
 - reduces the dimension of the original time series from M to $m \ll M$
- **Multiple filters:**
 - **three** different convolutions with lengths $l \in \{10, 20, 40\}$ are applied to the input time series, which is technically the output of the bottleneck layer

Inception Time [Fawaz 2020]



- **MaxPooling Layer:**
 - parallel MaxPooling followed by a bottleneck layer to reduce dimensionality
 - the output of a MaxPooling window is computed by taking the maximum value in this given window of time series
- **Depth Concatenation Layer:**
 - the output of each independent parallel convolution/MaxPooling is concatenated to form the output multivariate time series
 - the latter operations are repeated for each individual Inception module

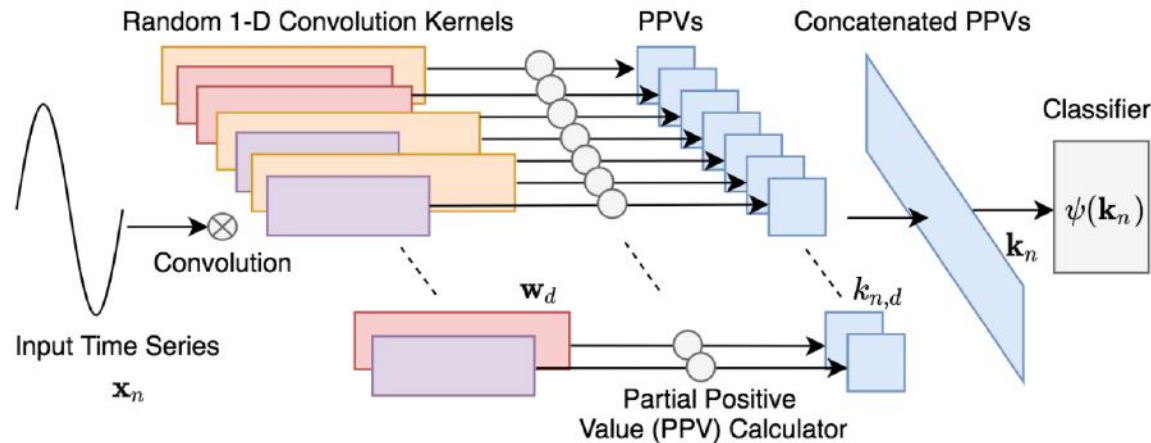
Random Convolutional Kernel Transform (ROCKET)



In short...

- **ROCKET** initializes a bank of random convolution kernels (e.g., 10 000)
- The convolution of each kernel with an input time series produces a **feature vector**
- Each feature vector is represented by the **proportion of (or partial) positive values (PPV)** and/or the maximum value (**max pooling**)
- The **concatenation** of PPV values from the kernels + the max pooling values is used as the input feature vector to train a **Ridge regression classifier**

Small random convolution kernels

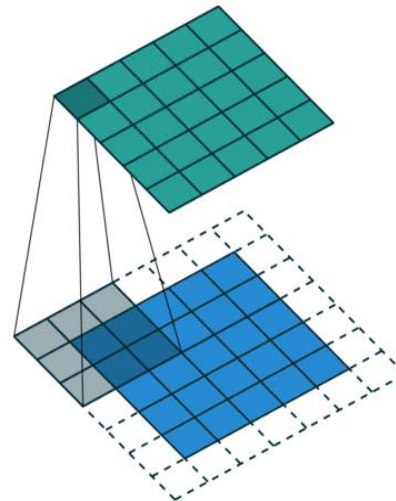
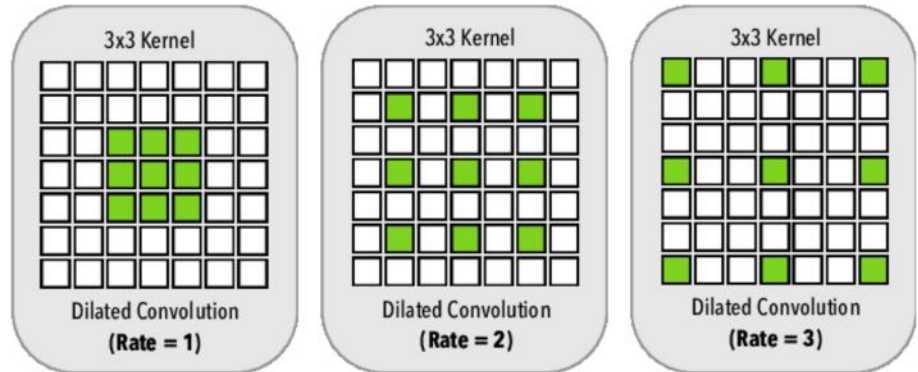


- Typical **kernel size**: $\{7, 9, 11\}$, selected **randomly** with **equal probability**
- **Kernel weights** are sampled from a **normal distribution**, $w \sim N(0,1)$, then mean centered: $w = W - \bar{W}$
- **Bias** $b \sim U(-1,1)$: the same kernel with different biases **highlights** different aspects of the resulting feature maps by **shifting** the values in a feature map above or below zero by a fixed amount

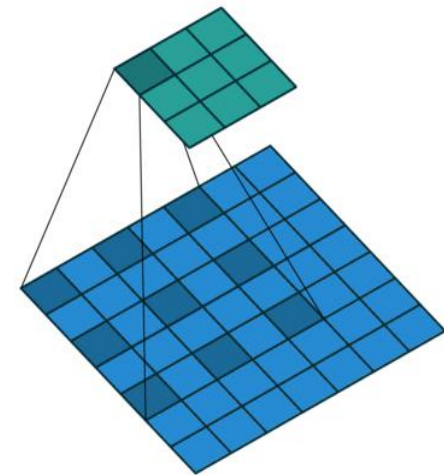
Dilated convolution

- A **3x3 kernel** with a **dilation rate of 2** will have the same field of view as a **5x5 kernel**, while only using **9 parameters**
- Equivalent to a **5x5 kernel** “ignoring” *every second column and row*
- Allows **kernels with similar weights** to capture **patterns at different scales**, despite rescaling

$$(F *_l k)(p) = \sum_{s+lt=p} F(s) k(t)$$



3x3 Kernel without dilation

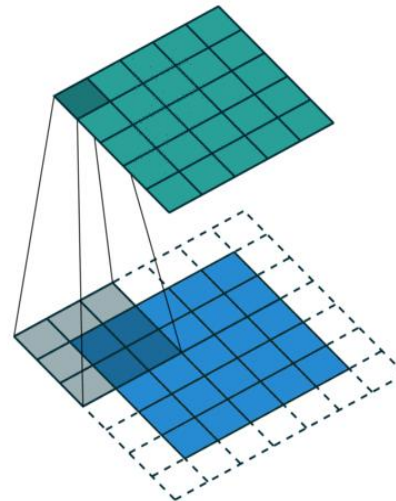


3x3 Kernel with dilation of 2

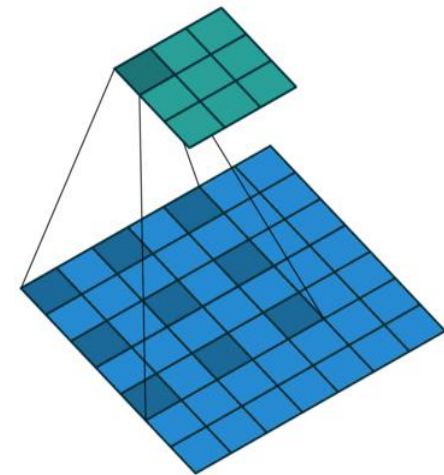
Dilated convolution – time series

- Frequency information is captured, since *large dilations* correspond to *lower frequencies* and *smaller dilations* to *higher frequencies*
- Kernels with similar weights capture patterns at different scales
- Multiple kernels with different dilations can capture discriminative patterns

- Sampled from an exponential scale, $d = \lfloor 2^x \rfloor$,
$$x \sim U(0, A), A = \log_2 \frac{l_{input}-1}{l_{kernel}-1}$$
- l_{input} : length of the input time series
- l_{kernel} : length of the Kernel



3x3 Kernel without dilation



3x3 Kernel with dilation of 2

Padding

- When each kernel is generated, a decision is made (at random, with equal probability) **whether or not padding will be used** when the kernel will be applied
- An amount of **zero padding** is appended to the **start** and **end** of each time series, so that the *middle* element of the kernel is centered on every time series point

$$padding = \frac{(l_{kernel}-1) \times d}{2}$$

- **Why is padding useful?**
 - without padding, kernels are not centered on the first and last $\lfloor l_{kernel}/2 \rfloor$ points of the time series; hence focusing only on **central regions**
 - with padding, kernels also *match patterns at the start or end of time series*

Final convolution features

- Each convolution

$$X_i * \omega = \sum_{j=0}^{l_{\text{kernel}}-1} X_{i+(j \times d)} \times \omega_j$$

results into **two feature values**:

- **max**: maximum value
- **PPV**: proportion of positive values

The final feature space is fed into a **linear classifier**, i.e, a Ridge or Logistic regressor

- stride = 1** for all convolutions
- ReLU is not applied** to the resulting feature maps

Final feature set: for **k kernels**, ROCKET produces **2k features per time series** (i.e., ppv and max); for example, for **10 000 kernels** (the default), ROCKET produces **20 000 features**

Variants of ROCKET

- **Mini-ROCKET***

	ROCKET	MINIROCKET
length	{7, 9, 11}	9
weights	$\mathcal{N}(0, 1)$	{-1, 2}
bias	$\mathcal{U}(-1, 1)$	from convolution output
dilation	random	fixed (rel. to input length)
padding	random	fixed
features	PPV + max	PPV
num. features	20K	10K

- **S-ROCKET**:**

- optimized kernels: minimizes the number of active kernels
- uses only **36-39%** (on average) of the kernels to achieve **similar classification accuracy** as **mini-ROCKET** and **ROCKET**

* <https://arxiv.org/pdf/2012.08791.pdf>

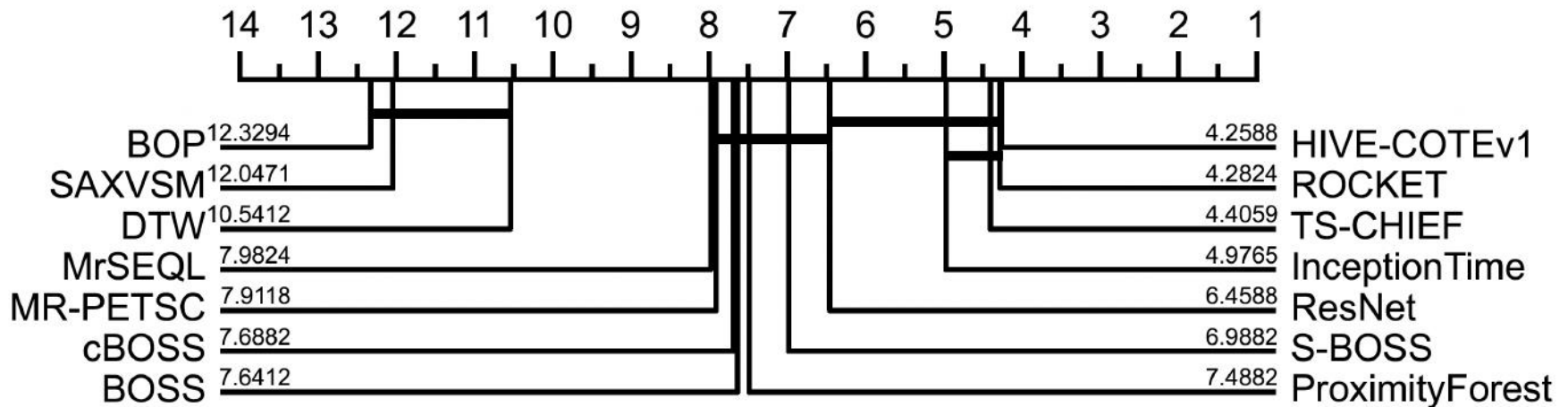
** <https://arxiv.org/pdf/2203.03445.pdf>

Other deep classifiers and ensembles

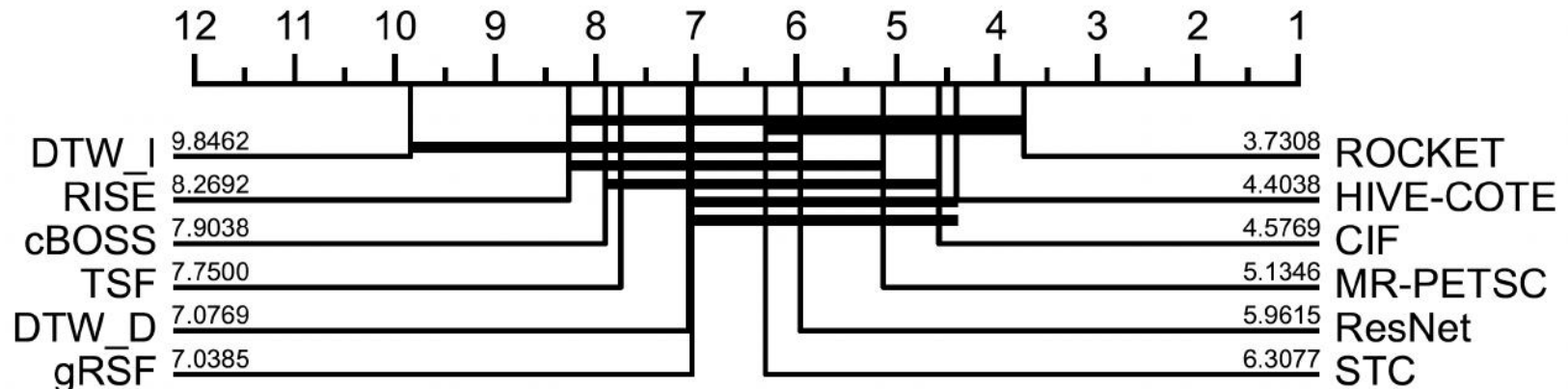
- **TapNet**: Time Series Attentional Prototype Network
- **ResNet** for time series classification
- **TS-CHIEF**: Time Series Combination of Heterogeneous and Integrated Embeddings Forest
- **HIVE-COTE**: Hierarchical Vote Collective of Transformation-based Ensembles
- **XEM**: An Explainable-by-Design Ensemble Method for Multivariate Time Series Classification
- **PETSC**: Pattern-Based Embedding for Time Series Classification

Overall winner?

Univariate time series classification



Multivariate time series classification



Outline

- Data series
- Electronic Health Records
- Time series summarization
- Time series classification
- Time series counterfactuals
- Challenges and future directions

Interpretable and actionable models

- Trade-offs between **interpretability** + **accuracy**
- Ability to **understand the prediction** **without compromising predictive** outcomes

I can tell you **what changes you need to make** to the patient record, so that I can **change my prediction** 😊



black box
classifier

The patient will die
from HF in 2 days!



Now what?
Please tell
me **why**?



What is a counterfactual (CF)?

- Given a **classifier** f , an **input instance** x with predicted **class label** c , defined over a set of variables
- A counterfactual explanation x' can provide an answer to the following question:

How should the configuration of the variables in x change to obtain class label c' instead of c ?

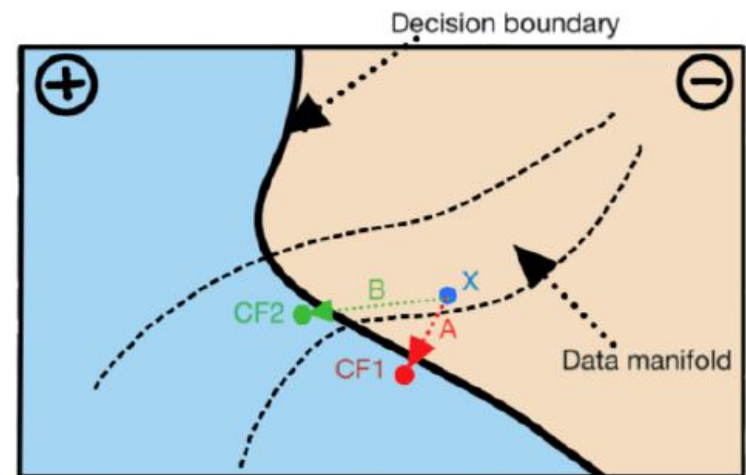


Computing counterfactuals

- Many possible solutions to the problem
- Two possible paths (A and B) for datapoint X (shown in blue)
- The red and green points are valid CFs by crossing the decision boundary
- Which one is preferable?

Several “goodness” metrics:

- proximity
- validity
- sparsity
- faithfulness
- fairness
- ...



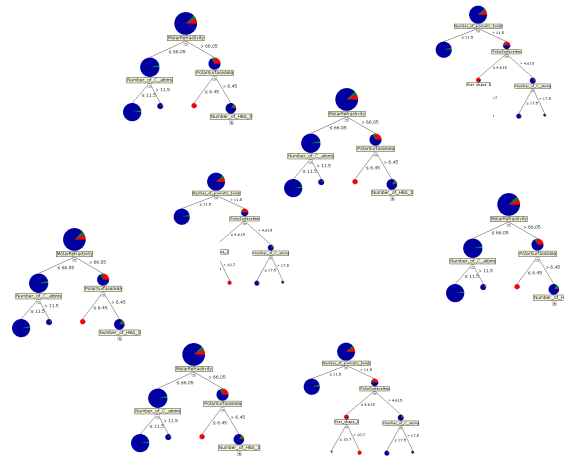
* Figure source: Verma, S., Dickerson, J., Hines, K.: Counterfactual Explanations for Machine Learning: A Review

Computing counterfactuals

- Several solutions both **model-specific** and **model-agnostic**
 - adjusting features until the **decision boundary is crossed**
 - adding **constraints**
- Many solutions for images based on **Autoencoders** and **Variational Autoencoders**
- Few solutions for **time series**

CFs for random forests (RFs)

ADE				...				...
 YES	✓	✗	✗	...	✓	✓	...	
 NO			✓	...			...	
 NO	✗	✗	✗	...	✓	✗	...	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮



$$\hat{f}(\mathbf{x}) = -1 \iff \left(\sum_{k=1}^K \hat{h}_k(\mathbf{x}) \right) \leq 0$$

Let $\mathbf{x}$ be a true-negative instance

Goal: minimum number of feature changes so that $\mathbf{x}$ becomes true-positive (TP), $\mathbf{x}'$

$$\mathbf{x}' = \arg \min_{\mathbf{x}^*} \left\{ \delta(\mathbf{x}, \mathbf{x}^*) \mid \hat{f}(\mathbf{x}) = -1 \wedge \hat{f}(\mathbf{x}^*) = +1 \right\}$$

Tolomei et al. Interpretable Predictions of Tree-based Ensembles via Actionable Feature Tweaking, KDD (and ARXIV), 2017

Solution for random forests (RFs)

- Let x be a true-negative (TN) instance
- **Goal:** minimum number of feature changes so that x becomes true-positive (TP), x'

Observe:

- If the prediction of the RF is -1, then **at least half of its trees** predict -1
- If the prediction of a tree is -1, then **the example is passed through a negative path** that predicts the class to be -1
- **Solution:** **revert these negative paths** and consequently the trees!

Note: if a single transformation results in changing another tree's decision, then ignore it!

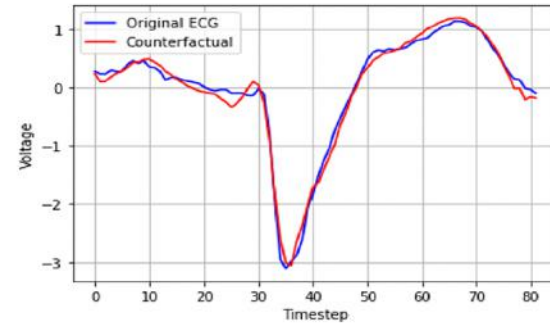
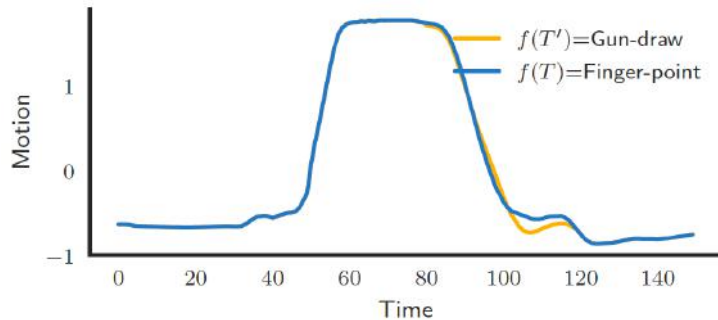
- Focus on the trees that **predict -1**
- For each tree: **explore the positive paths**, i.e., those that **predict +1**
- **Apply the transformations imposed by the positive path**

$$\mathbf{x}_{j(\epsilon)}^+[i] = \begin{cases} \theta_i - \epsilon & \text{if the } i\text{-th condition is } (x_i \leq \theta_i) \\ \theta_i + \epsilon & \text{if the } i\text{-th condition is } (x_i > \theta_i) \end{cases}$$

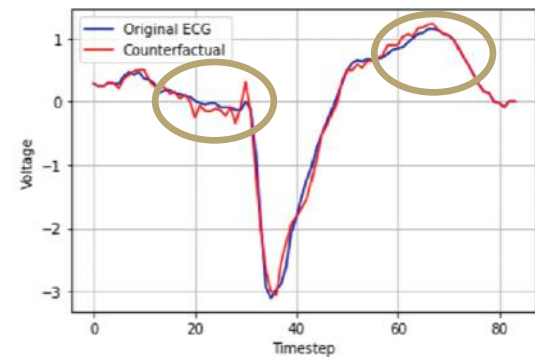
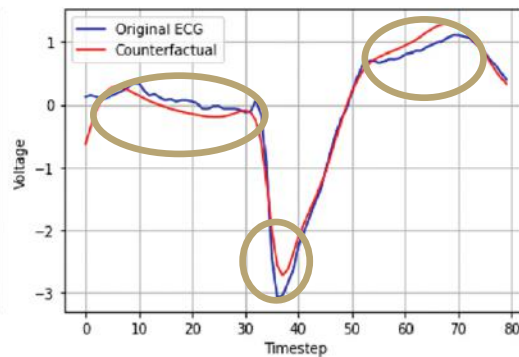
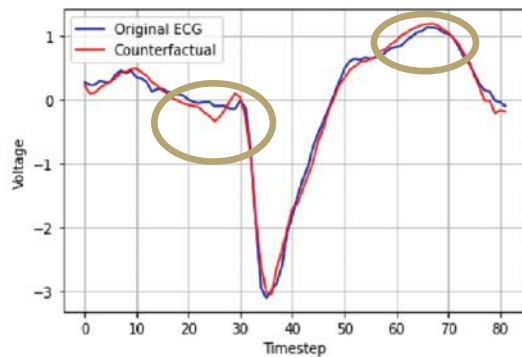
- Choose the transformation with the **minimum cost**

$$\mathbf{x}' = \arg \min_{\mathbf{x}_{j(\epsilon)}^+ \in \Gamma \mid \hat{f}(\mathbf{x}_{j(\epsilon)}^+) = +1} \left\{ \delta(\mathbf{x}, \mathbf{x}_{j(\epsilon)}^+) \right\}$$

Time series counterfactuals



Goal: What is the **minimum number of changes** to apply to a time series T so that a **given opaque classifier** changes its prediction?



Time series counterfactuals for RSF

- Focus on the trees that **predict -1**
- For each tree, explore the positive paths, i.e., those that **predict +1**
- Try to **force those trees to predict +1** by changing **shapelet features** of T

Given a non-leaf node (S_k^j, θ_k^j)

- **Increase distance:**
 - if S_k^j exists in T, that is $d_s(\mathcal{S}_k^j, \mathcal{T}) \leq \theta_k^j$
 - and the current node condition demands otherwise
 - ✓ increase the distance of **all matching instances** of S_k^j , so that they all fall **above the distance threshold θ_k^j**

Time series counterfactuals for RSF

- Focus on the trees that **predict -1**
- For each tree, explore the positive paths, i.e., those that **predict +1**
- Try to **force those trees to predict +1** by changing **shapelet features** of T

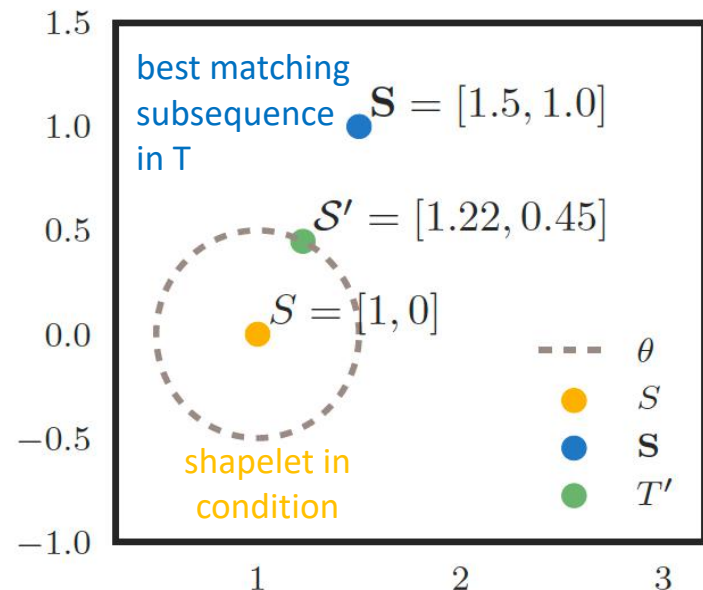
Given a non-leaf node (S_k^j, θ_k^j)

- **Decrease distance:**
 - if S_k^j does not exist in T, that is $d_s(\mathcal{S}_k^j, \mathcal{T}) > \theta_k^j$
 - and the current node condition demands otherwise
 - ✓ decrease the distance of **the best matching instance** of S_k^j , so that it falls **below the distance threshold** θ_k^j

How to transform the time series?

- Consider shapelet S as an **m-dimensional point**
- Define an **m-sphere** with S as its center and radius θ
- The **transformed time series counterpart of S** is given by the following equation:

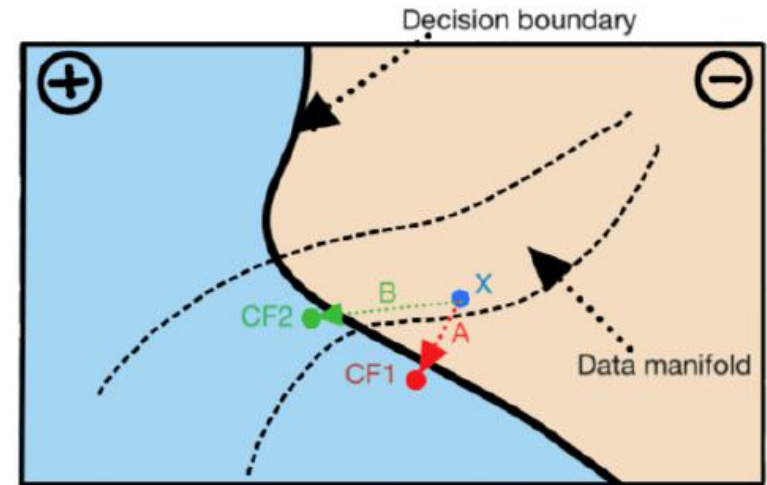
$$\tau_S(\mathbf{S}, p_{ik}^j, \epsilon) = \mathcal{S}_k^j + \frac{\mathcal{S}_k^j - \mathbf{S}}{\|\mathcal{S}_k^j - \mathbf{S}\|_2} (\theta_k^j + (\epsilon \delta_{ik}^j))$$



Karlsson et al. Explainable time series tweaking via irreversible and reversible temporal transformations, ICDM 2018

Counterfactual validity

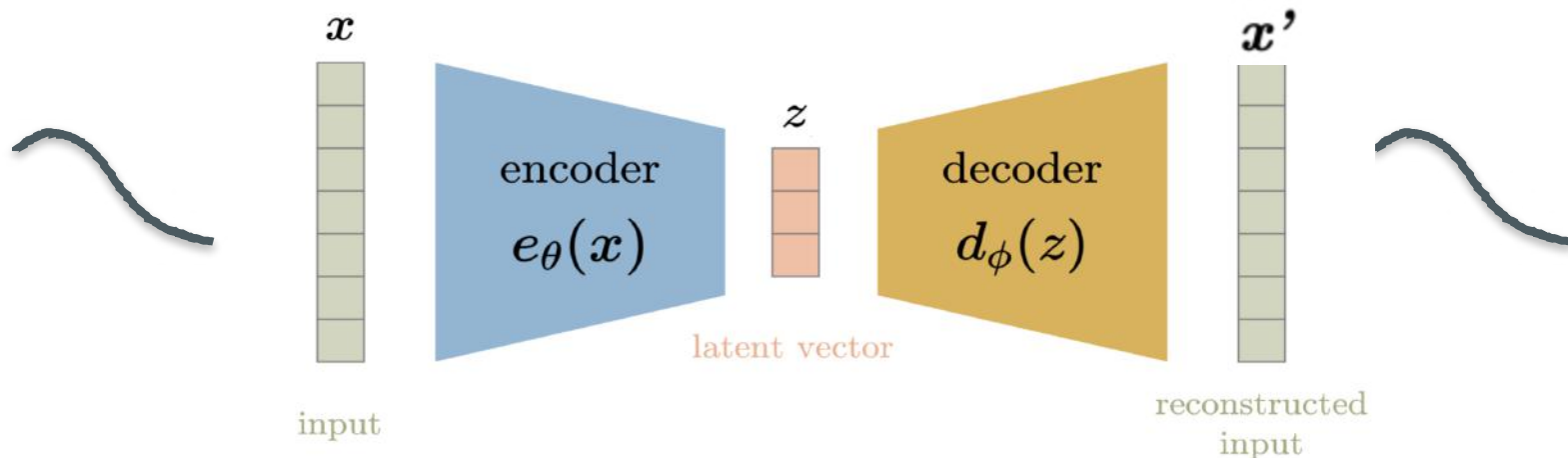
- It is not only **sparsity** and **proximity** that matter
- Counterfactuals should also be **valid**:
 - **compliant** with the original **data distribution**
 - should be **expected** to be observed



- **Solution:** find a way to learn the data manifold / distribution per class

* Figure source: Verma, S., Dickerson, J., Hines, K.: Counterfactual Explanations for Machine Learning: A Review

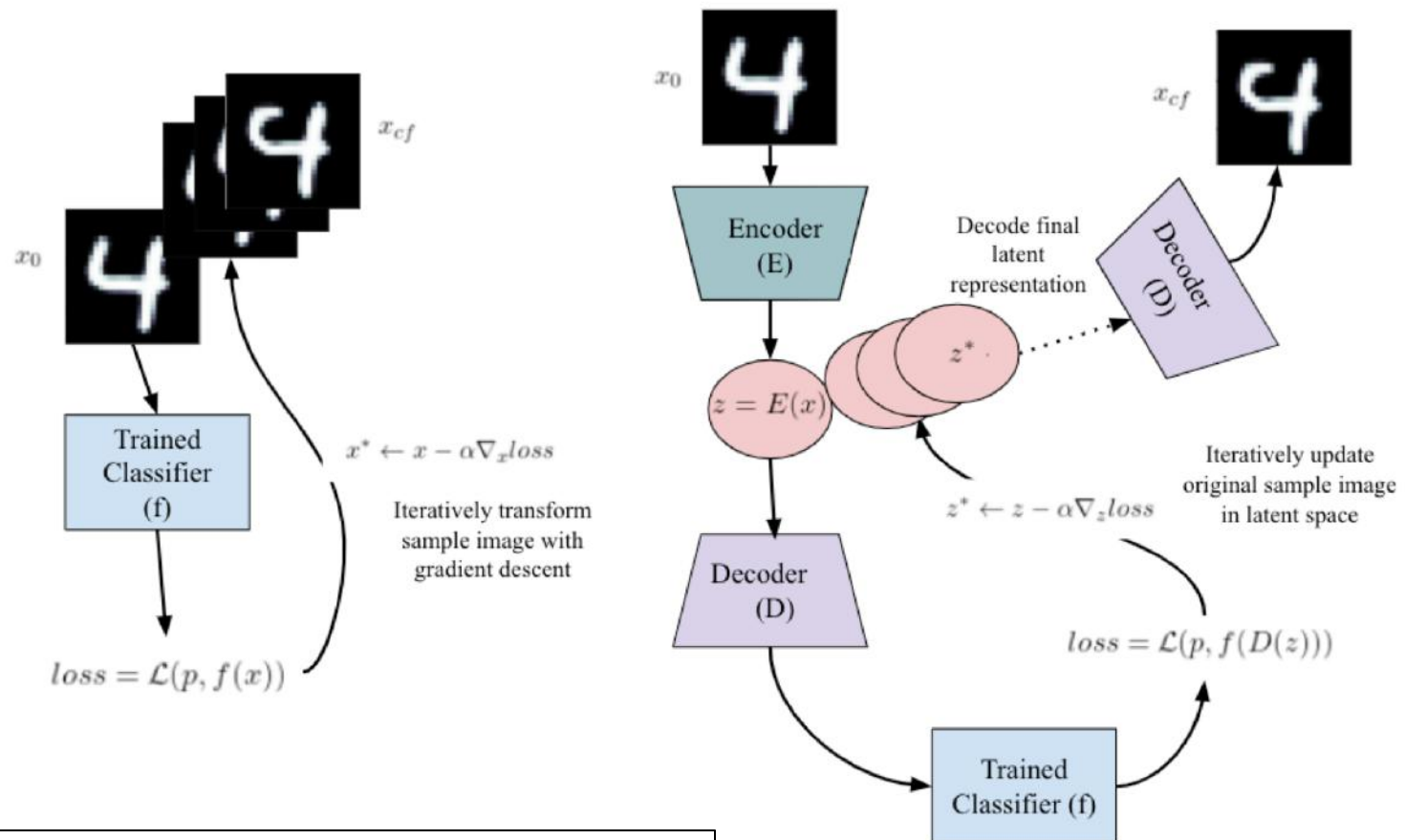
Autoencoders



$$loss = \|x - x'\|_2 = \|x - d_{\phi}(z)\|_2 = \|x - d_{\phi}(e_{\theta}(x))\|_2$$

- Use an **auto-encoder** to find the generated counterfactual with the desired class (e.g., positive) outcome
- **Perturb** the encoded latent representation $z = e(x)$ through a **gradient descent optimization approach** iteratively to generate a new time series sample $x' = d(z)$ such that the output target $f(x') = '+'$

Latent space CFs

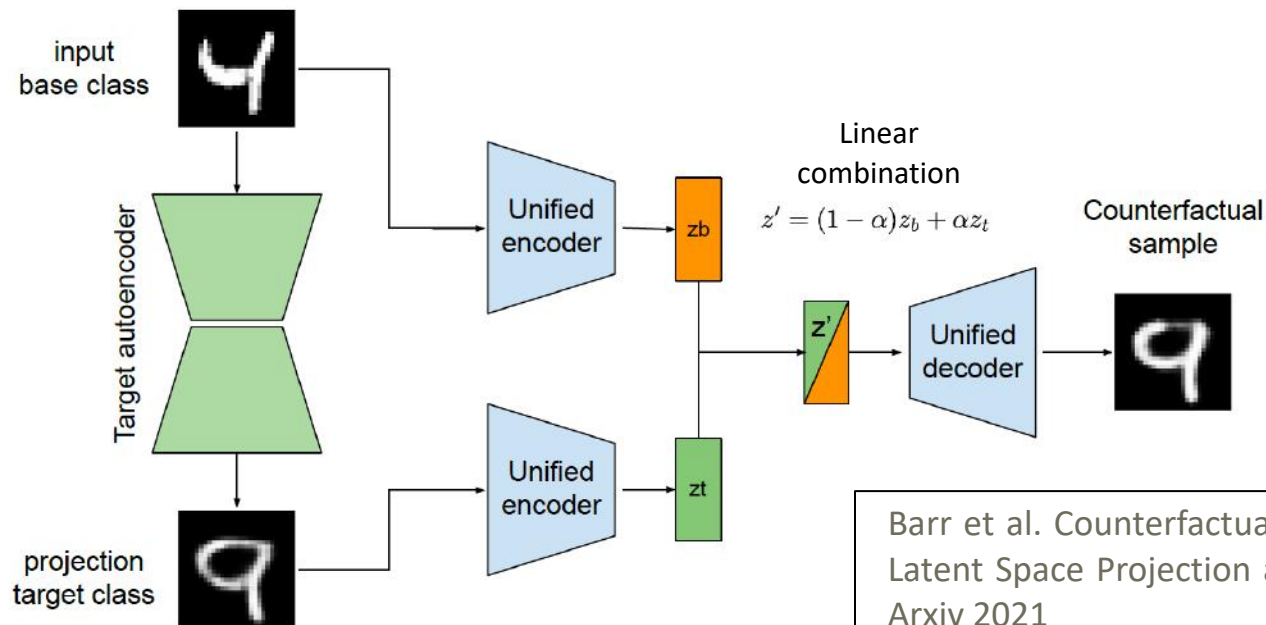


Balasubramanian et al. Latent-CF: A Simple Baseline for Reverse Counterfactual Explanations, Arxiv 2020

SharpShooter: Latent Space Projection and Interpolation

Two Autoencoders

- **Target AE:** for learning the class conversion
- **Unified AE:** for learning a common latent representation trained on both classes



Outline

- Data series
- Electronic Health Records
- Time series summarization
- Time series classification
- Time series counterfactuals
- Challenges and future directions

Multimodal learning

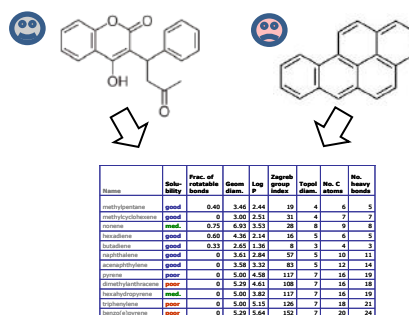
- Develop techniques and tools to concurrently learn from **multiple data types**
- Data **integration** and **fusion**
- Distributed data sources

Electronic patient records



Karolinska University Hospital

Chemical compound data



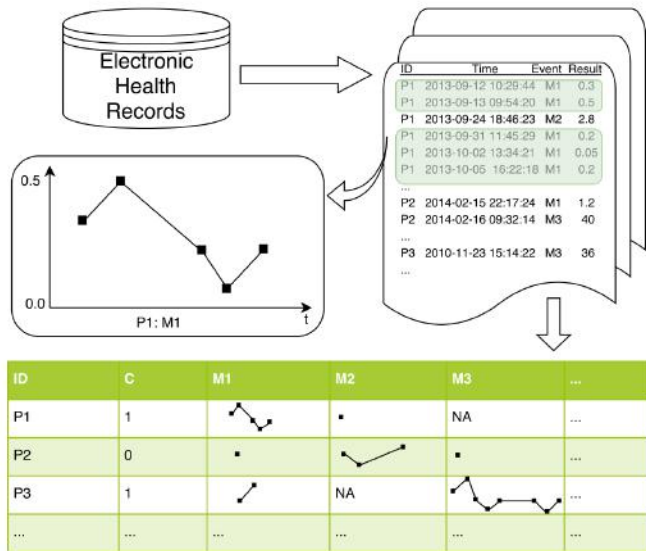
Pharmaceutical companies

Individual case safety reports



WHO Collaborating Centre for International Drug Monitoring

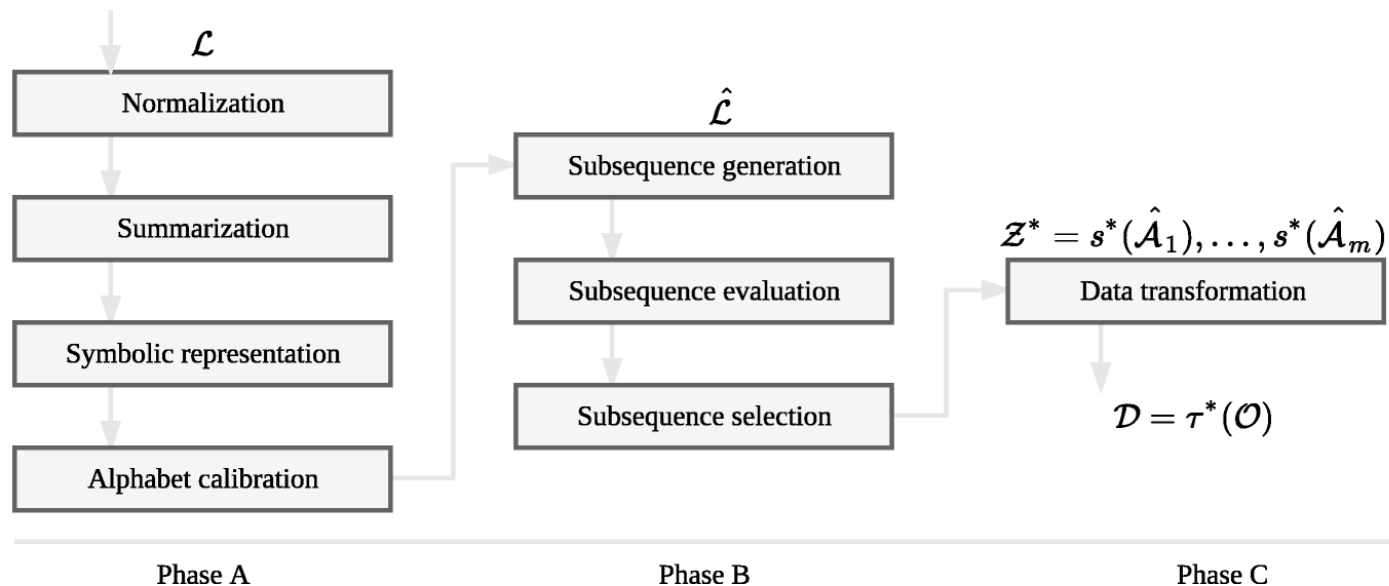
Dealing with data sparsity



Hielscher et al. **Mining Longitudinal Epidemiological**

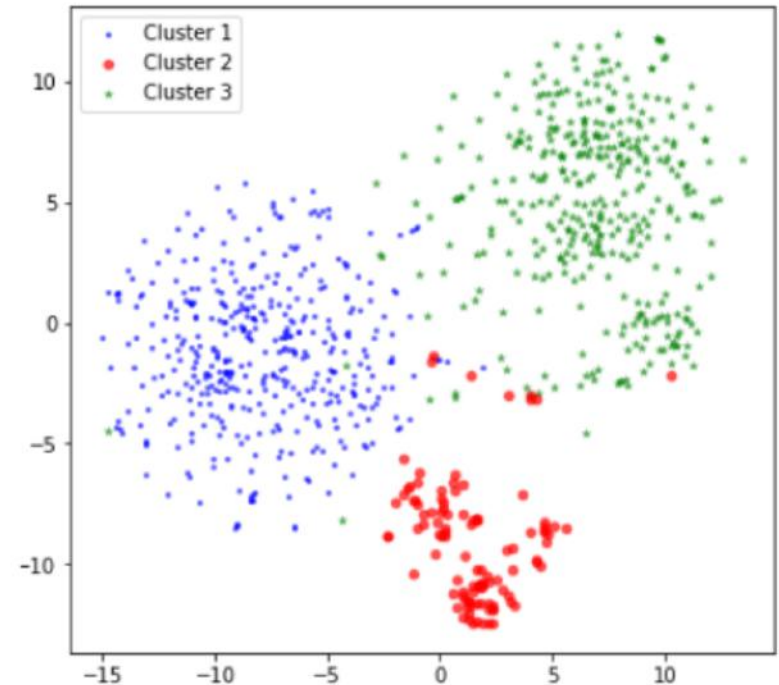
Bagattini et al. **A classification framework for exploiting**

Rebane et al. **SMILE: a feature-based temporal abstraction framework for event-interval sequence classification**, Data Mining and Knowledge Discovery, 2021



Patient profiling and risk assessment

- Identify patient subgroups and profiles
- Such subgroups can be used for better patient understanding, treatment recommendation, and risk assessment
- Machine learning techniques such as subgroup discovery and subspace clustering can be employed
- Define counterfactuals for clustering



Bampa et al. **A Clustering Framework for Patient Phenotyping with Application to Adverse Drug Events**, CBMS 2020

Towards actionable models

- Trade-offs between interpretability + accuracy
- Ability to understand the predictions + act to prevent undesirable outcomes without compromising predictive performance



black box
classifier



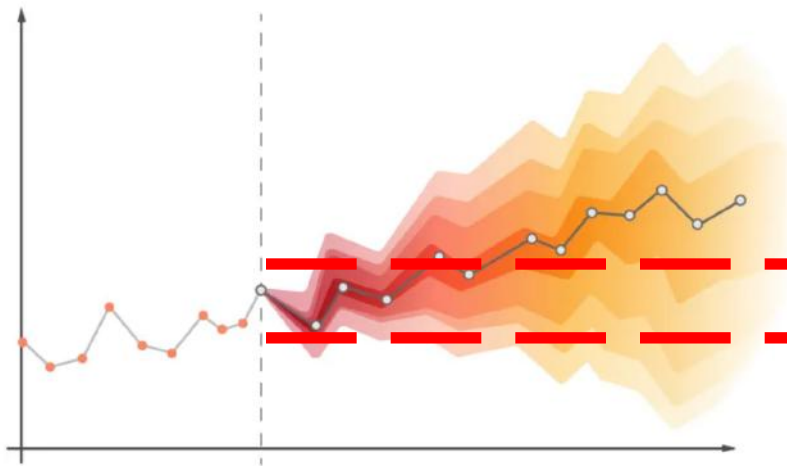
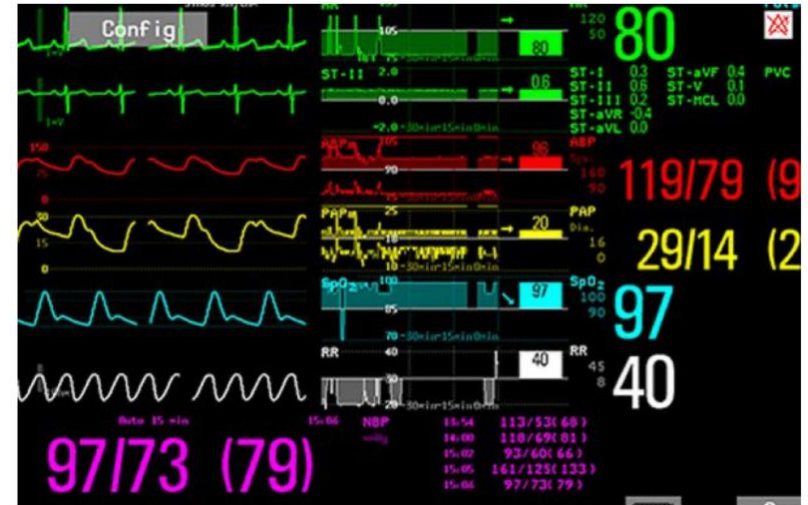
The patient will die
from HF in **2 days!**

Now what?
Please tell me
why?



Forecasting patient progression

- Monitor current patient **vitals**
- Forecast their **progression**
- Identify **timely interventions**
- Towards **forecasting counterfactuals**



Maintain the prediction within a **constant band**

Early interventions to prevent **“violating”** the band

Challenges

- Mixed data types
- Missing values
- Short time series
- Explainable and models
- Patient profiling

Thank you!



Stockholms
universitet



Panagiotis Papapetrou

Professor, Stockholm University

panagiotis@dsv.su.se

<https://papapetrou.blogs.dsv.su.se>

<https://datascience.dsv.su.se>