Time Series Analytics: Overview of Classification Methods

Danial Senejohnny, Data Scientist & Analytics Translator

Knowledge Sharing Lecture Notes, May 2023

Time Series (TS) Problems

- Classification:

Feature 1	Feature 2	Label
Series1	Series3	Class A
Series2	Series4	Class B
Series5	Series6	?

- Forecasting:

time	Value
2022-05-02	75
2022-05-03	25
2022-05-04	?

- Annotation:
    - Change point detection
    - Outlier detection
    - Time series segmentation

- Regression:

Feature 1	Feature 2	Target
Series1	Series3	10
Series2	Series4	25
Series5	Series6	?

- Clustering:

- Anomaly Detection:

Overview TS Python Packages

Classification

SKTIME

pyts

A Python Package for Time Series Classification

tslearn

tsai

state-of-the-art deep learning for time series and sequences

Forecasting & Regression

SKTIME

Darts

GLUONTS

tsai

state-of-the-art deep learning for time series and sequences

statsmodels

Clustering

SKTIME

tslearn

Anomaly Detection

GLUONTS

Feature Extraction & Engineering

tsfresh

Feature-engine

TS Classification Applications

- Transaction classification, fraudulent vs legitimate
- Supervised classification of share price trends
- Classify recession-and-recovery events emerging in the countries’ business cycles.
- Classifying churning clients with very dynamic time series feature [depo values]
- Some other examples that we will see soon.
- What other potential use cases?

TS Classification Approaches

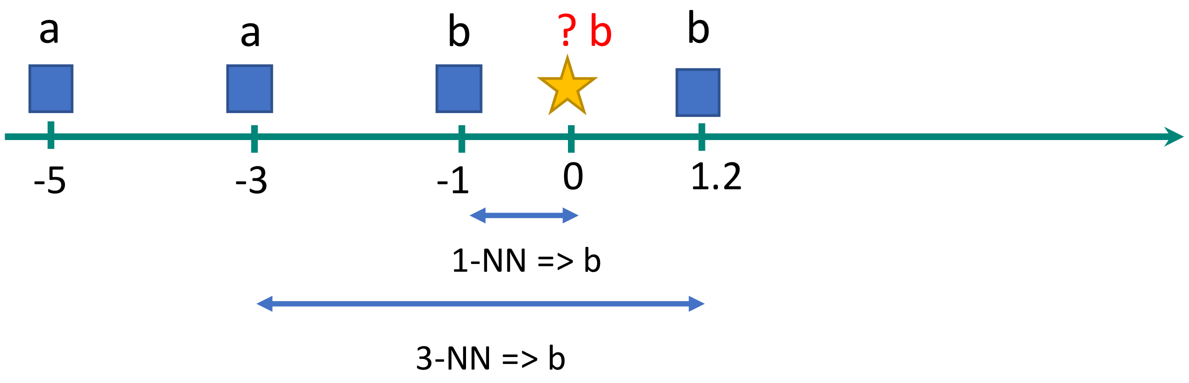
- **Distance-Based** 6
  - Gold standard for TS classification
  - Use distances, i.e., Euclidian,
- **Interval-Based** 5
  - Random interval contribute to classification
- **Dictionary-Based** 7
  - Time series transformed to bag of words
- **Shapelet-Based** 2
  - Learn from small shapes in time series

- 3 **Kernel-Based**
  - Rocket is a fast & accurate classifier
- 7 **Feature-Based**
  - Transform TS to classical tabular problem
  - Extend TS to clustering & survival analysis
- 6 **Deep Learning Based**
  - Will be out of scope
- 2 **Ensamble**
  - Collection of elite classifiers

Distance Based

- K-Nearest Neighbour (KNN)
  - 1-NN is a basic TS classifier and gold standard for TS classification

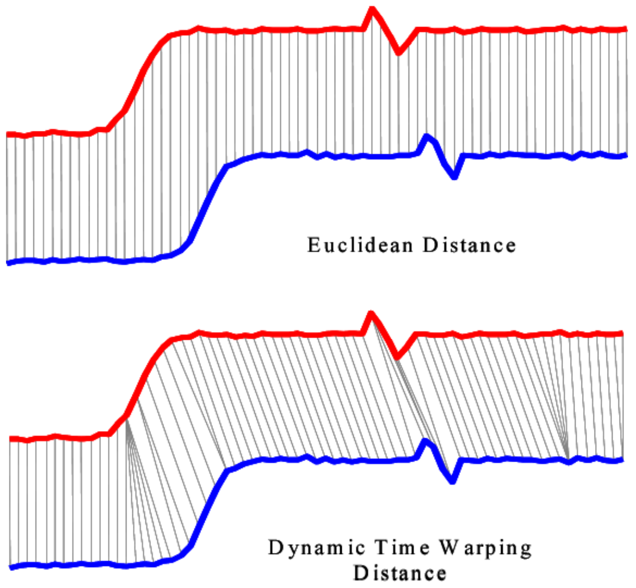
- Algorithm:
  1. Choose distance as a metric to define (dis)similarity between 2 TS:
  2. Calculated the distance among a TS with all TS in the training set
  3. The TS is assigned the class to which most of the KNN belong.

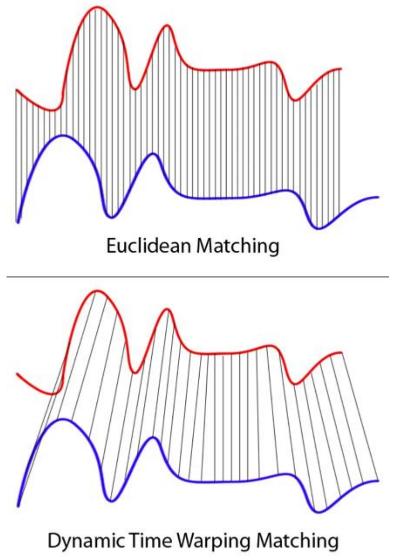


Distance Based [Metric Comparison]

- **p-norms:** [1 to 1 comparison]
  - 1st-norm [Manhattan distance]
  - 2nd-norm [Euclidian distance]

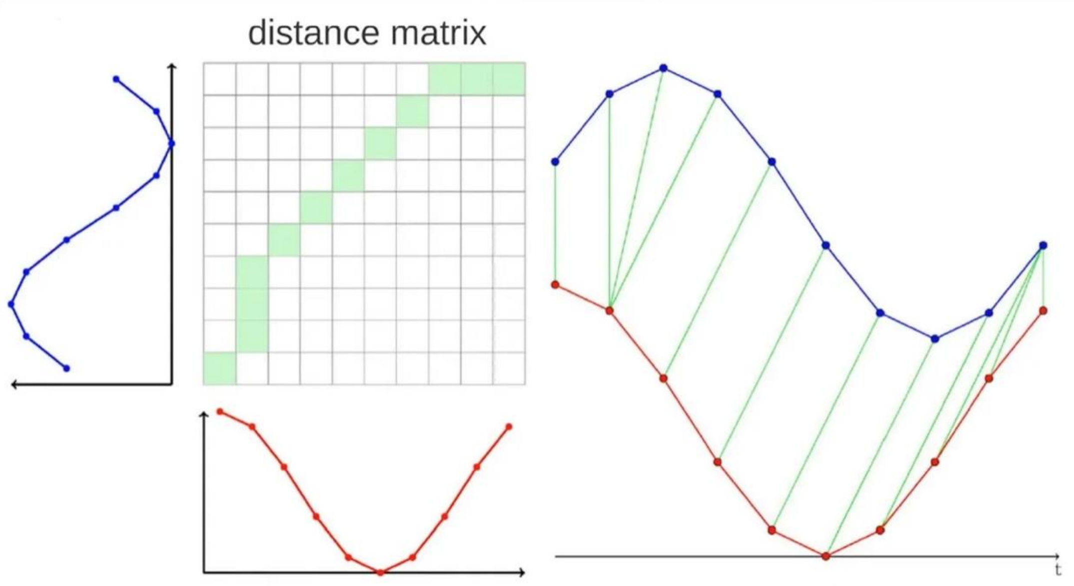
- **Dynamic Time Warping:** [Many to 1 comparison]
  - Advantage over p-norms
    - Compare time series which are
      - Asynchronous TS with different length & delay
  - Disadvantage over p-norms
    - Computationally expensive
    - Does not satisfy all properties of a distance





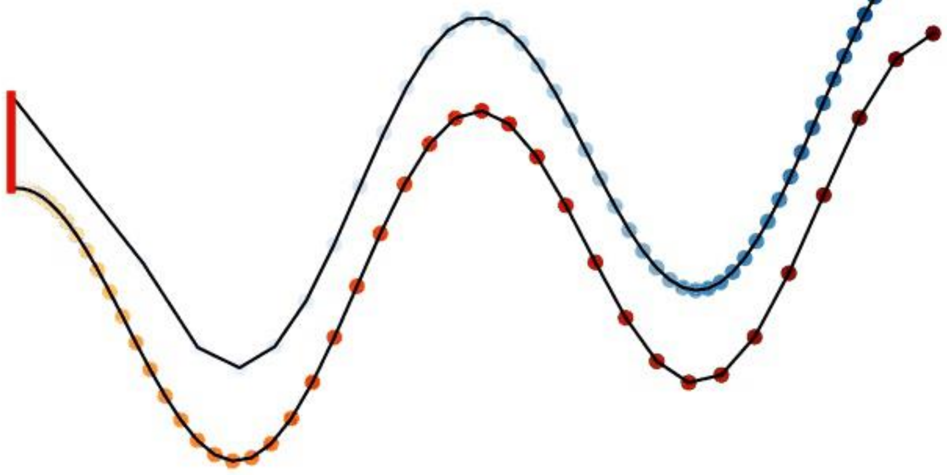
Distance Based [DTW Explained]

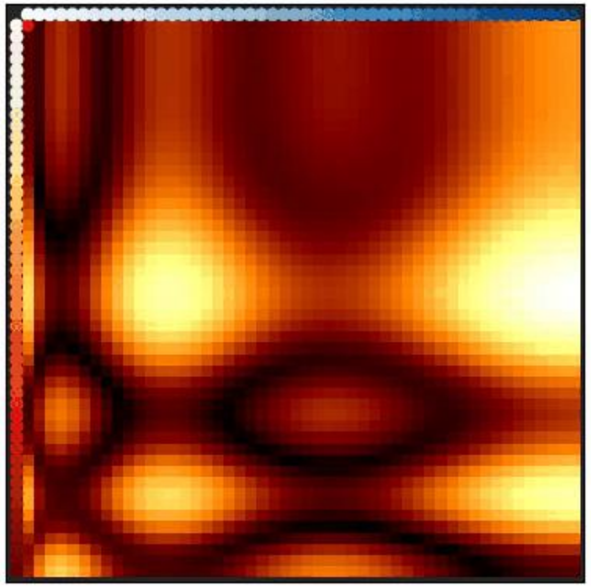
- Under the hood uses a p-norm (Euclidian) distance
- Warping path minimizes the distance between 2 time series
- Efficiently finding the warping path:
  - Overlap begin & end of signals [Boundary condition]
  - No backward moevement [Monotonicity condition]
  - Chess king move [Continuity condition]



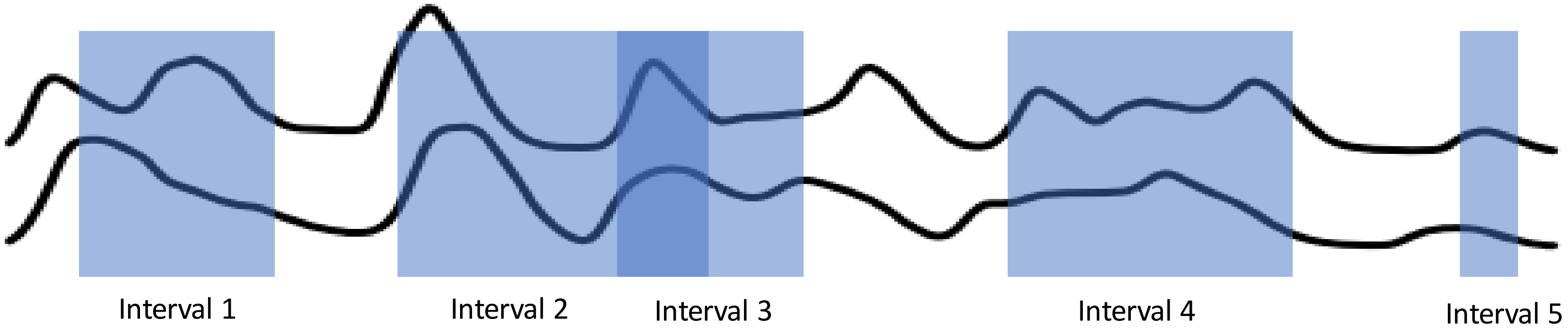
https://medium.com/mlearning-ai/what-is-dynamic-time-warping-253a6880ad12

Distance Based [More example DTW]





Interval-Based

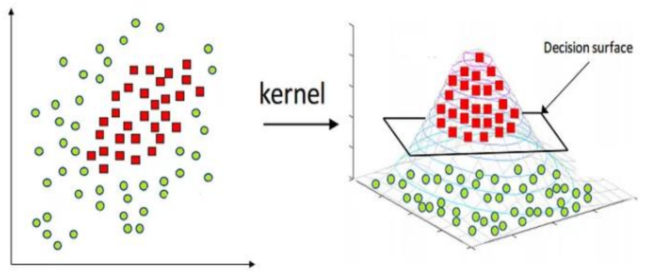


- Time Series Forrest:
  1. For each tree, *k* intervals are randomly selected
  2. Summarize Interval to *m* features [mean, standard deviation, slope]
  3. Each time series is a single feature vector of length *mk*
  4. Decision tree is grown on these vectorized time series
  5. New TS are classified using the majority vote of all tress in the forest

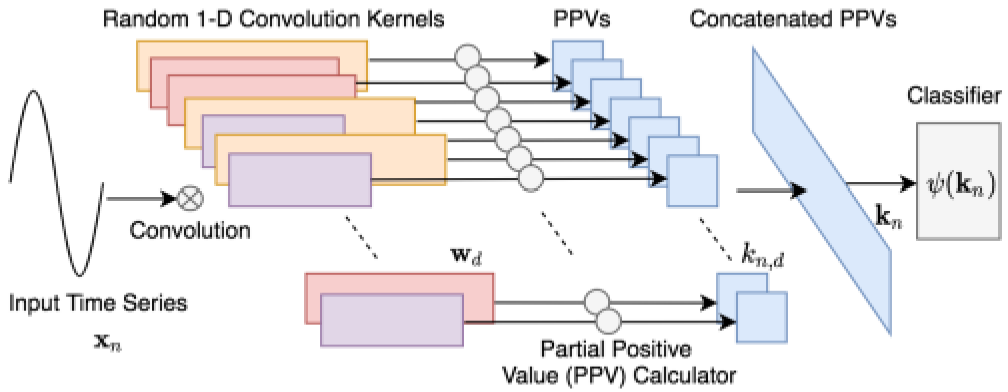
- Canonical Interval Forest (CIF) uses catch22, [Improved model DrCIF]
- RISE: Extract spectral features like fourier, (partial) autocorrelation, auto-regressive

Kernel based

Rocket transform time series data into a higher dimensional space



https://medium.com/@zxr.nju/what-is-the-kernel-trick-why-is-it-important-98a98db0961d



S-Rocket: Selective Random Convolution Kernels for Time Series Classification

- Advantages:
  - Fast training time and high accuracy.
  - Useful for large datasets, as it can process multiple time series simultaneously.

Shapelet [Intuitive Example]

Lexiang Ye · Eamonn Keogh, Time series shapelets: a novel technique that allows accurate, interpretable and fast classification

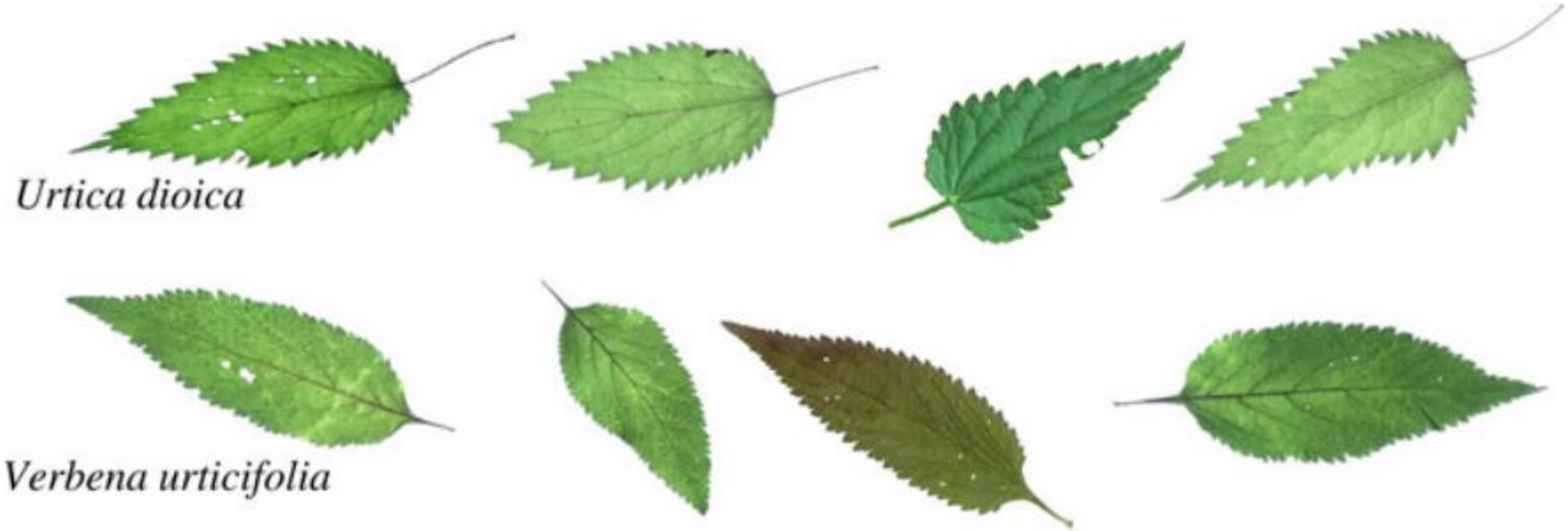


Fig. 1 Samples of leaves from two species. Note that several leaves have insect-bite damage

Shapelet [Intuitive Example]

Transform a 2-dimensional leaf shape into a 1-dimensional representation

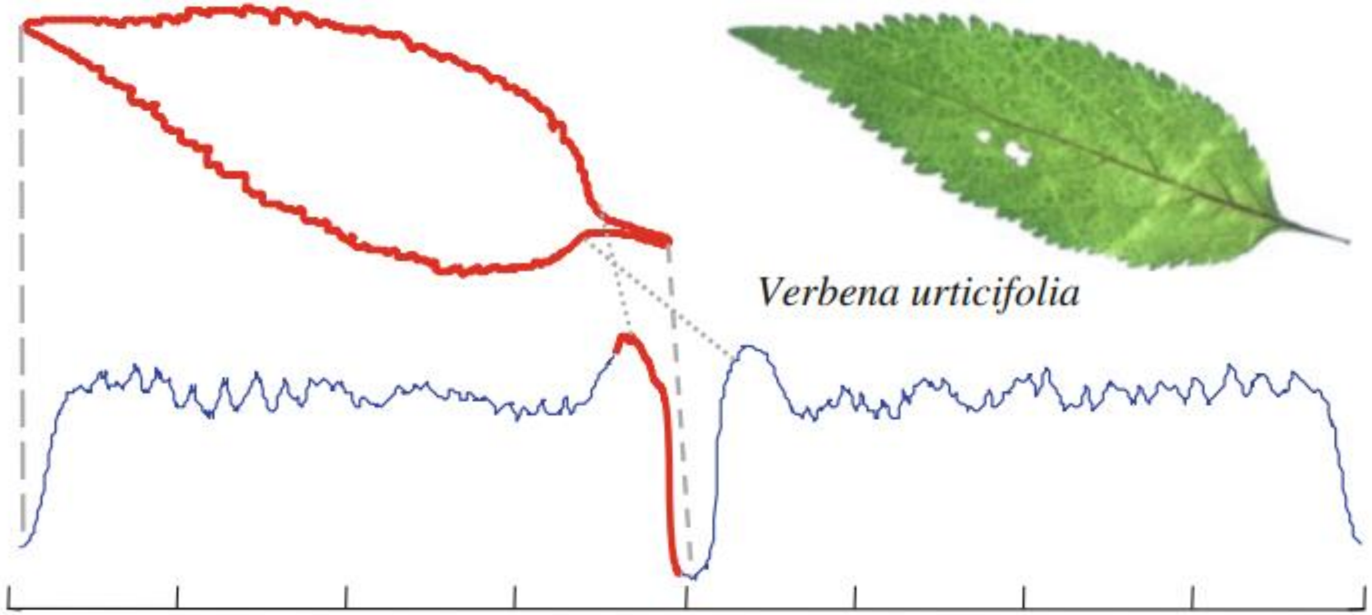


Fig. 2 A shape can be converted into a one dimensional "time series" representation. The reason for the highlighted section of the time series will be made apparent shortly

Shapelet [Intuitive Example]

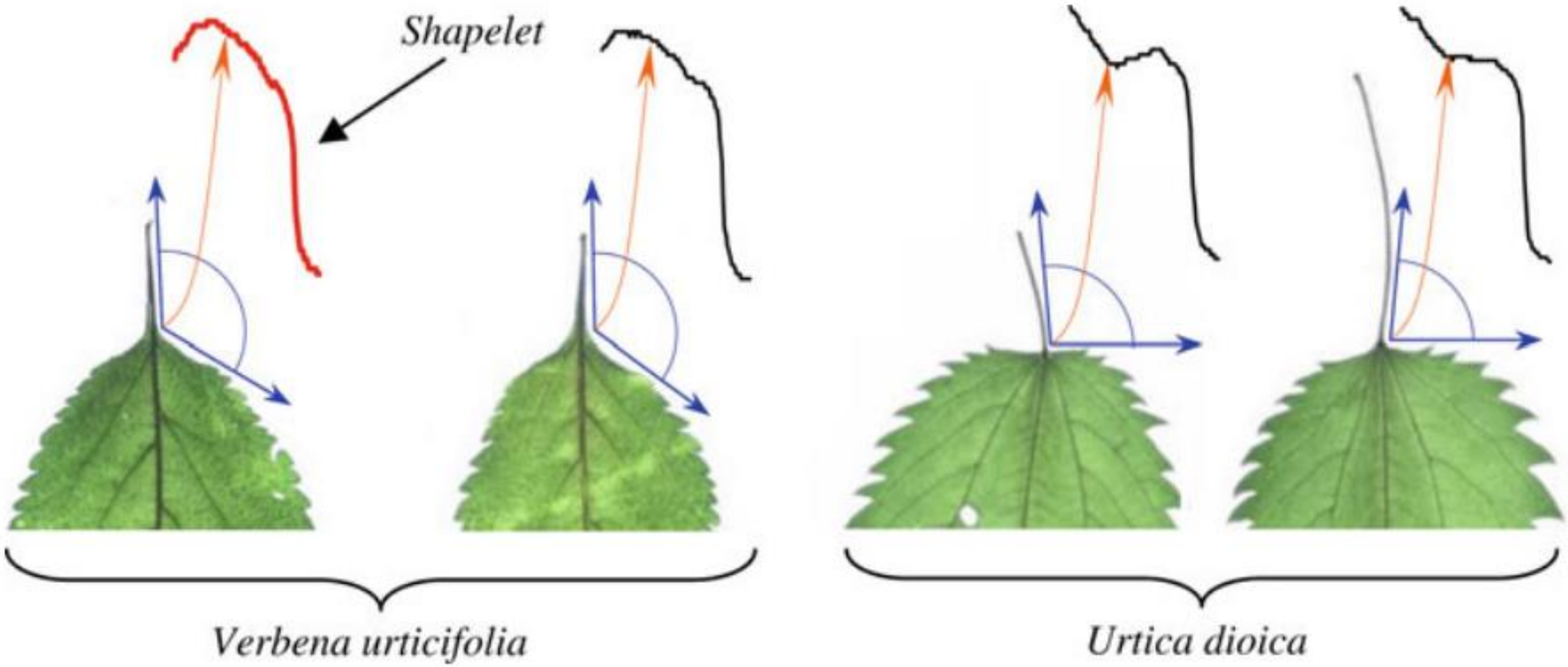
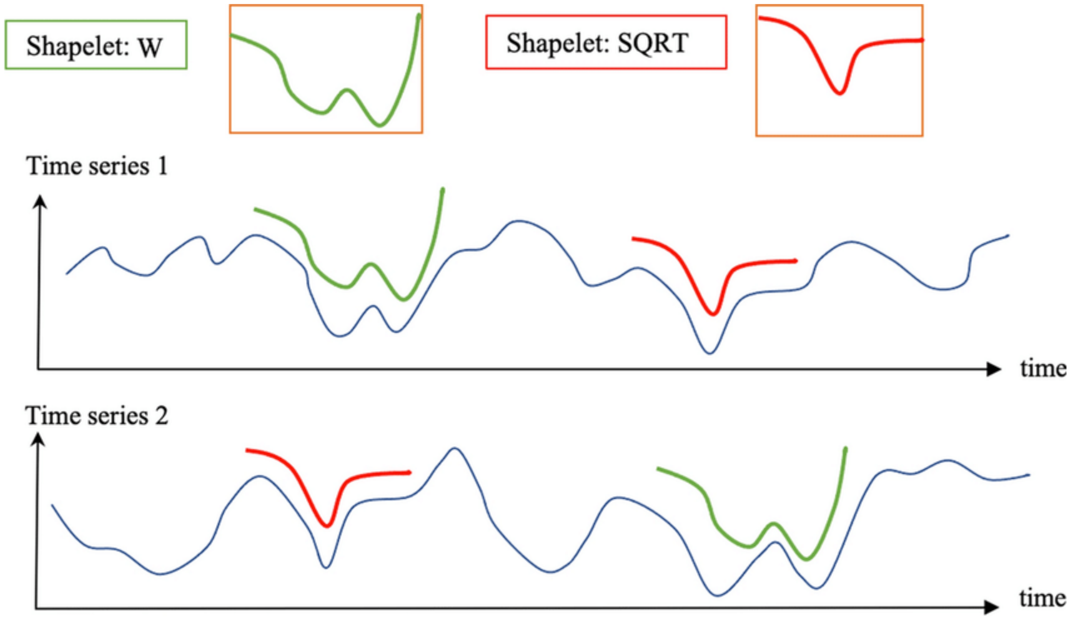


Fig. 3 Here, the shapelet hinted at in Fig. 2 (in both cases shown with a *bold line*) is the subsequence that best discriminates between the two classes

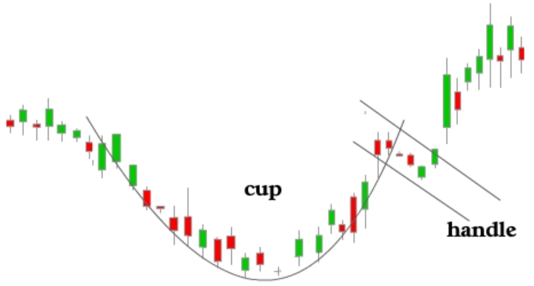
Shapelet [More Example]

- Technical analyst look for trading patterns/shapelets

https://en.wikipedia.org/wiki/Recession_shapes



[A Customized Machine Learning Algorithm for Discovering the Shapes of Recovery: Was the Global Financial Crisis Different?]()

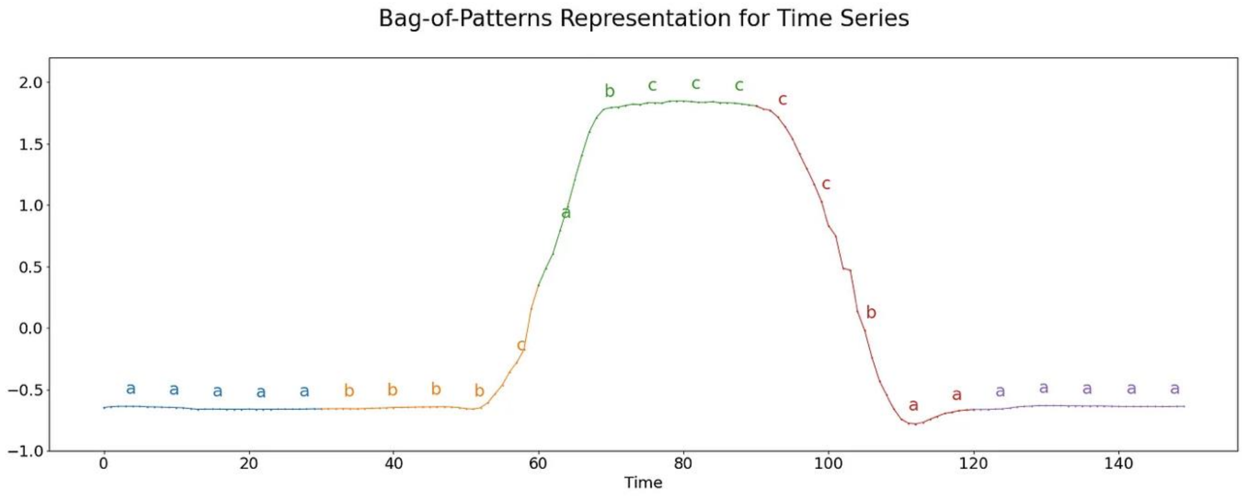


Shapelet [Discovery]

- Shapelets are chosen such
  - They can be used as a representative of a class,
  - Distinguish different classes from each other.

- How to make/discover shapelets:
  - Make them manually
  - Automatic:
    - Algorithms that extract shapelets
      - Disadvantage: Computationally complex
    - Algorithms that learn shapelets
      - Advantage: computationally faster & shapelets are not biased to training data

Dictionary Based Methods



Bag of patterns (BoP):

1. Run a sliding window of a specific length over the TS
2. Transform each subsequence into a word (with a specific length and a fixed set of letters)
3. Classify using word distributions [histogram]

- Symbolic Fourier Approximation (SFA), Individual BOSS, BOSS Ensemble, BOSS in Vector Space, contractable BOSS, Randomized BOSS, WEASEL

Feature Based Method[Summary]

- The idea is

Time Series
Series1
Series2

→

Var 1	...	Var n
10	...	25
8	...	17

- Then classical methods like clustering & survival analysis can be applied to time series

- 2013: hctsa paper
  - 20k time series and ~9k operations
  - hctsa Implementation in Matlab

Highly comparative time-series analysis: the empirical structure of time series and their methods

Ben D. Fulcher[1], Max A. Little[1] and Nick S. Jones[1,2]

[1]Department of Physics, University of Oxford, Oxford, UK
[2]Department of Mathematics, Imperial College London, London, UK

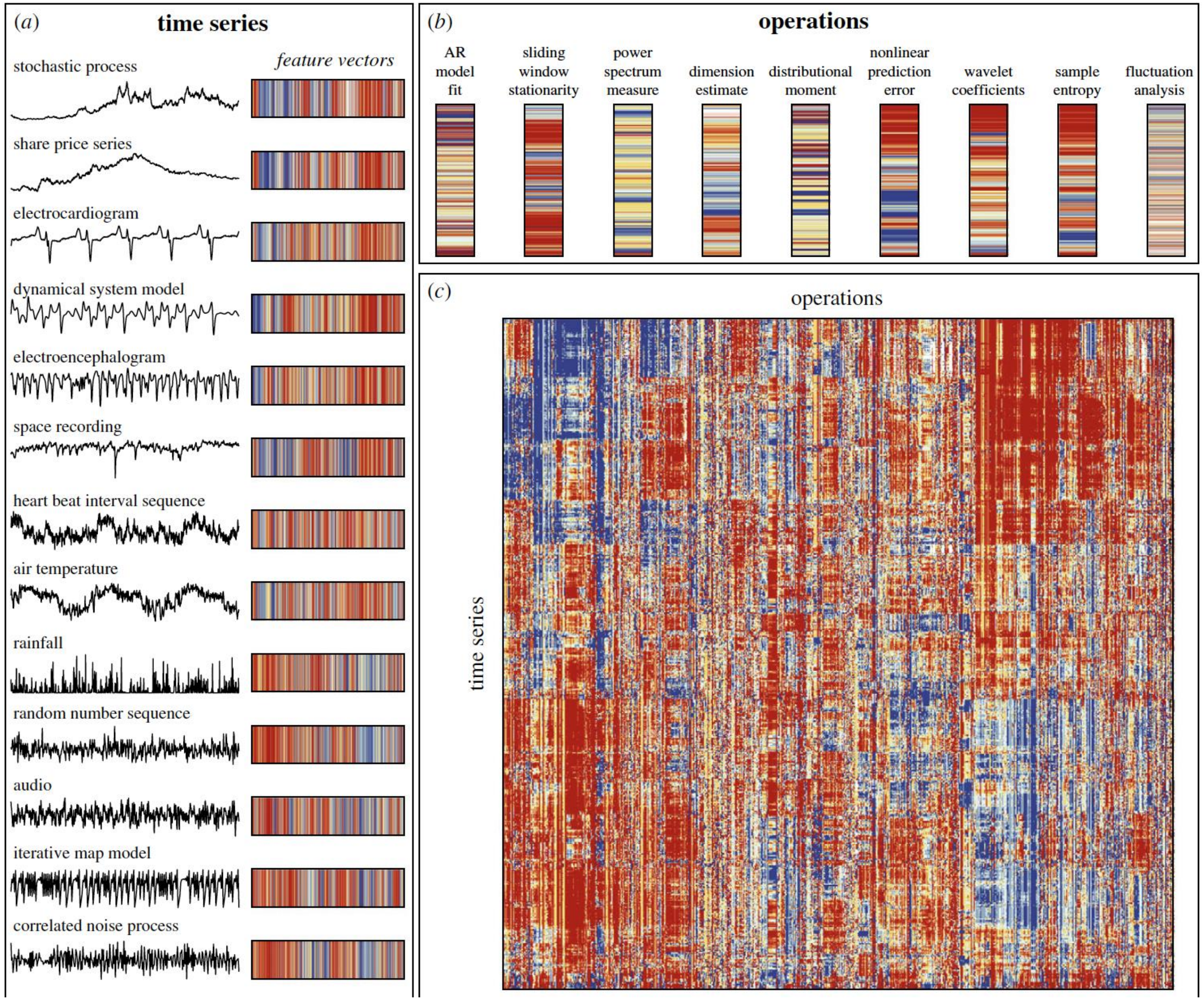
hctsa highly comparative time-series analysis

https://hctsa-users.gitbook.io/hctsa-manual/

- 2020: catch22: CAnonical Time-series Characteristics selected from hctsa [python]
  - Used 93 time-series datasets (147,000 time series) with 4791 hctsa features
  - Found 22 strongest predictors from 4791, with just 7% drop in performance

Feature Based Method [hctsa]

Highly comparative time-series analysis: the empirical structure of time series and their method



- Row: 875 interdisciplinary time series.
- Column: 8651 operation
- Normalized Elements of the matrix: blue (low operation outputs) to red (high operation outputs).
- Rows and columns have been reordered using **linkage clustering** to place similar time series and similar operations close to one another

Feature Based Method [Catch22 Features]

HCTSA

- basic statistics of the distribution
  - location, spread, Gaussianity and outlier properties),
- linear correlations
  - autocorrelations and features of the power spectrum),
- stationarity
  - StatAv, sliding window measures and unit root tests, prediction errors),
- information theoretic and entropy measures
  - (e.g. auto-mutual information, approximate entropy and Lempel–Ziv complexity),
- methods from the physical nonlinear timeseries analysis literature
- (e.g. correlation dimension, Lyapunov exponent estimates and surrogate data analysis),
- linear and nonlinear model fits
  - Goodness of fit and parameter values from ARMA, GARCH, Gaussian process, and state space models

Catch 22

hctsa feature name	Description
Distribution	
DN_HistogramMode_5	Mode of z-scored distribution (5-bin histogram)
DN_HistogramMode_10	Mode of z-scored distribution (10-bin histogram)
Simple temporal statistics	
SB_BinaryStats_mean_longstretch1	Longest period of consecutive values above the mean
DN_OutlierInclude_p_001_mdrmd	Time intervals between successive extreme events above the mean
DN_OutlierInclude_n_001_mdrmd	Time intervals between successive extreme events below the mean
Linear autocorrelation	
CO_f1ecac	First $1/e$ crossing of autocorrelation function
CO_FirstMin_ac	First minimum of autocorrelation function
SP_Summaries_welch_rect_area_5_1	Total power in lowest fifth of frequencies in the Fourier power spectrum
SP_Summaries_welch_rect_centroid	Centroid of the Fourier power spectrum
FC_LocalSimple_mean3_stderr	Mean error from a rolling 3-sample mean forecasting
Nonlinear autocorrelation	
CO_trev_1_num	Time-reversibility statistic, $\langle (x_{t+1} - x_t)^3 \rangle_t$
CO_HistogramAMI_even_2_5	Automutual information, $m = 2, \tau = 5$
IN_AutoMutualInfoStats_40_gaussian_fmmi	First minimum of the automutual information function
Successive differences	
MD_hrv_classic_pnn40	Proportion of successive differences exceeding 0.04σ (Mietus 2002)
SB_BinaryStats_diff_longstretch0	Longest period of successive incremental decreases
SB_MotifThree_quantile_hh	Shannon entropy of two successive letters in equiprobable 3-letter symbolization
FC_LocalSimple_mean1_tauresrat	Change in correlation length after iterative differencing
CO_Embed2_Dist_tau_d_expfit_meandiff	Exponential fit to successive distances in 2-d embedding space

Ensemble

- HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles
  - HIVE-COTE 1.0 [2018]
  - HIVE-COTE 2.0 [2021]

- List of classifiers in HIVE-COTE 2.0:
  - **Dictionary-Based:** Temporal dictionary ensemble
  - **Kernel-Based:** a ROCKET ensemble
  - **Interval-Based:** Diverse representation canonical interval forest (DrCIF)
  - **Shapelet-Based:** Shapelet transform classifier (STC)

Thank you for your attention!

Any Questions or suggestions?

Any application that could benefit from TS Classifiation?

Appendix

Classification vs Regression vs Forecasting

Classification

	Feature 1	Feature 2	Label
Sample 1	Series1	Series3	Class A
Train Sample	Series2	Series4	Class B
Test Sample	Series5	Series6	?

Forecasting

	time	Value
Time Unit 1	2022-05-02	75
Time Unit 2	2022-05-03	25
Time Unit 3	2022-05-04	?

Regression

	Feature 1	Feature 2	Target
Sample 1	Series1	Series3	10
Sample 2	Series2	Series4	25
Sample 3	Series5	Series6	?

Clustering

Feature 1	Feature 2	Target
Series1	Series3	10
Series2	Series4	25
Series5	Series6	?

Forecasting

Time series to a table of features and a target

Time	Sales	Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
2020-02-01	35	UK	12	100	NaN	NaN	NaN	35
2020-02-02	30	UK	15	120	NaN	NaN	35	30
2020-02-03	23	UK	13	116	NaN	35	30	23
2020-02-04	21	UK	14	120	35	30	23	21
2020-02-05	40	UK	23	101	30	23	21	40
2020-02-06	31	UK	25	90	23	21	40	31
2020-02-07	?	UK	x_{T+1}	190	21	40	31	?

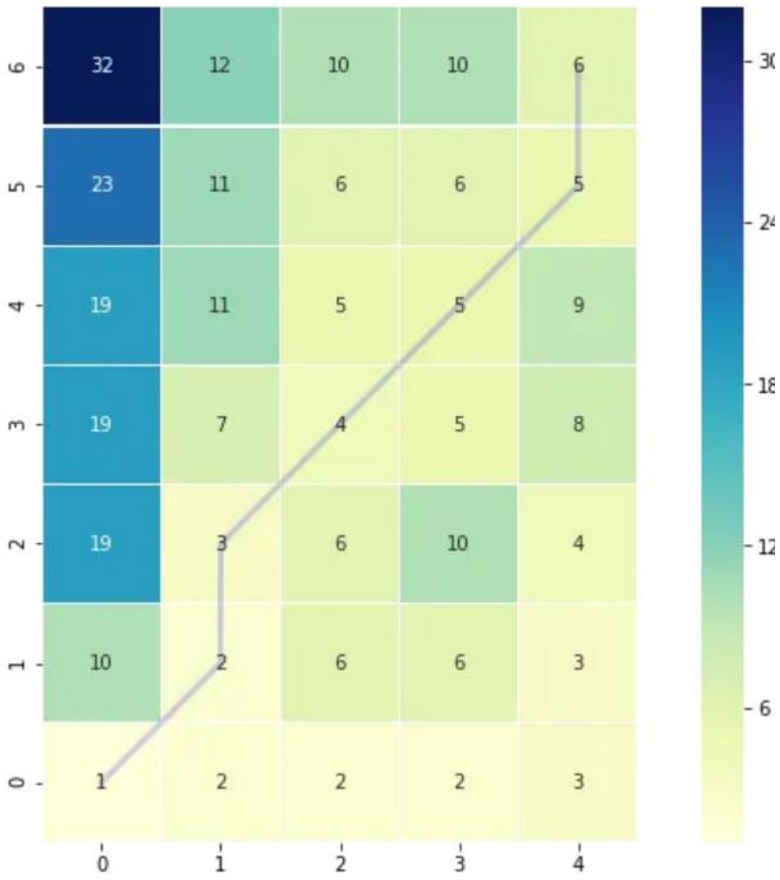
Static features.

Features with unknown values in the future.

Features with known values in the future.

Features derived from past values of y_t (e.g., lag features)

Dynamic Time Warping



Convolution Theory

- 1-D convolution Kernel

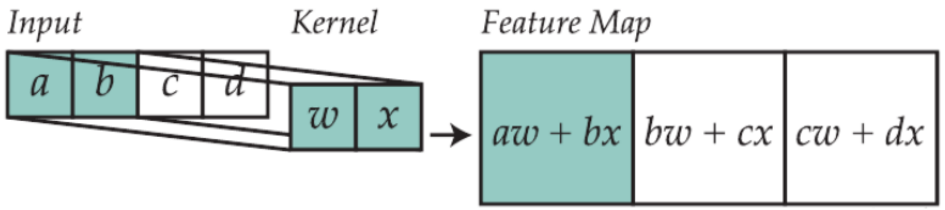
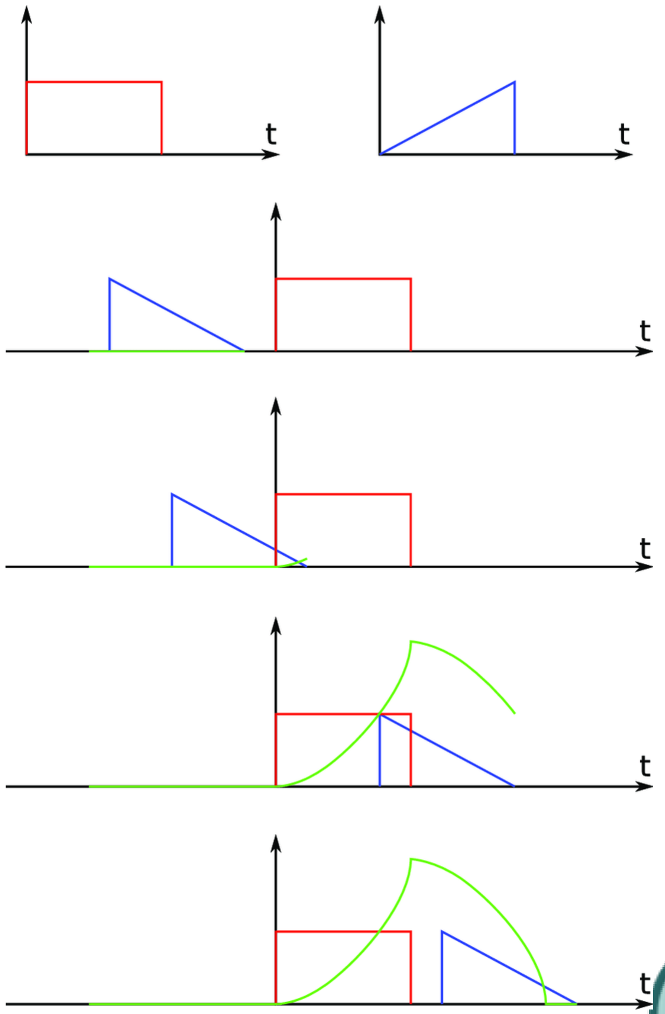


Figure 2. Example of how a 1D kernel is used to convolve the input.

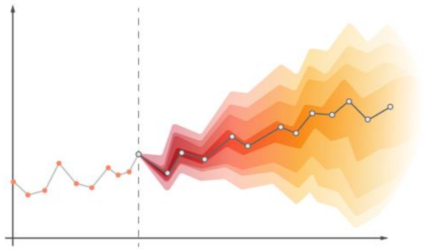
Evolutionary Design of Convolutional Neural Networks for Human Activity Recognition in Sensor-Rich Environments



Forecasting

- Most of the time forecasting problems are only one signals sample and not a cohort of samples
- During implementation care should be given to in-sample vs out-of-sample forecast

Univariate forecasting



Feature 1 Window[10]	Prediction horizon [3]
$x_1, x_2, \ldots . x_{10}$	$\hat{x}_{11}, \hat{x}_{12}, \hat{x}_{13}$
$x_2, x_2, \ldots . x_{11}$	$\hat{x}_{12}, \hat{x}_{13}, \hat{x}_{14}$
$x_3, x_4, \ldots . x_{12}$	$\hat{x}_{13}, \hat{x}_{14}, \hat{x}_{15}$

Multivariate forecasting

Feature 1 Window[10]	Feature 2 Window[10]	Prediction horizon [3]
$x_1, x_2, \ldots . x_{10}$	$y_1, y_2, \ldots . y_{10}$	$\hat{x}_{11}, \hat{x}_{12}, \hat{x}_{13}$
$x_2, x_2, \ldots . x_{11}$	$y_2, y_2, \ldots . y_{11}$	$\hat{x}_{12}, \hat{x}_{13}, \hat{x}_{14}$
$x_3, x_4, \ldots . x_{12}$	$y_3, y_4, \ldots . y_{12}$	$\hat{x}_{13}, \hat{x}_{14}, \hat{x}_{15}$

- In **supervised regression**, we predict *label/target variables* from *feature variables*, in a cross-sectional set-up. This is after training on label/feature examples.

- In **forecasting**, we predict *future values* from *past values*, of *the same variable*, in a temporal/sequential set-up. This is after training on the past.

http://www.sktime.net/en/latest/examples/01a_forecasting_sklearn.html

- In the common data frame representation:
  - In **supervised regression**, we predict entries in a column from other columns. For this, we mainly make use of the statistical relation between those columns, learnt from examples of complete rows. <u>The rows are all assumed exchangeable.</u>
  - In **forecasting**, we predict new rows, assuming temporal ordering in the rows. For this, we mainly make use of the statistical relation between previous and subsequent rows, learnt from the example of the observed sequence of rows. <u>The rows are not exchangeable, but in temporal sequence.</u>