Artificial Intelligence for Signal and Image Processing

Change-point detection with application to human gait analysis

Laurent Oudre
laurent.oudre@ens-paris-saclay.fr

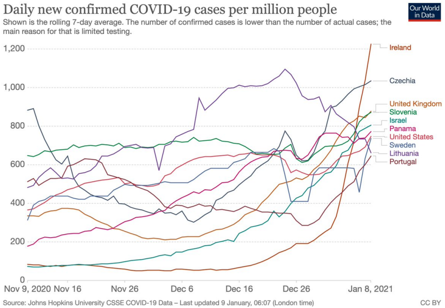
September, 10th 2021

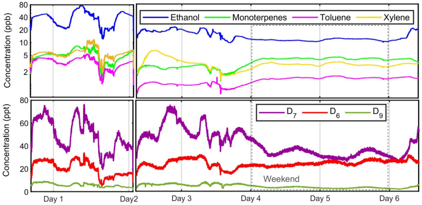
Contents

Change-point detection

- Wide-sense stationarity is one of the most common assumption in signal processing
- In several contexts, some changes occur in the data over time: most real time series are not stationary
- Change-point detection consists in estimating as precisely as possible the times where these changes occur
- Two main usecases:
  - Usecase 1: Retrieve these breakpoints to detect when the changes occurred
  - Usecase 2: Use the breakpoint information to divide the time series into smaller stationary signals

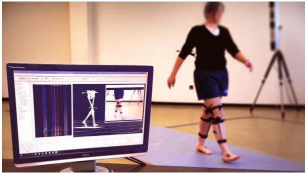
Usecase 1: monitoring

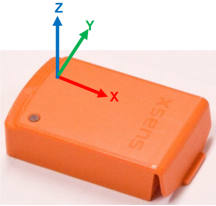


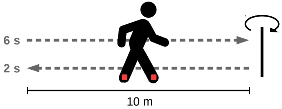


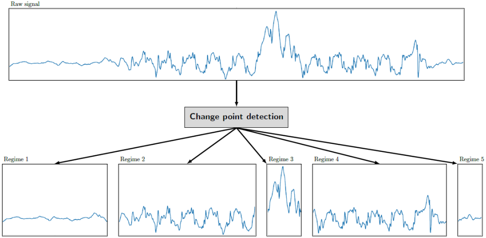
Usecase 2: gait analysis

- Human activity can be recorded with inertial measurement units (IMUs)
- Complex protocols with several activities create non-stationary time series

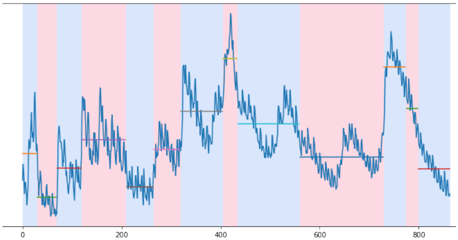








Problem statement: Change-Point Detection

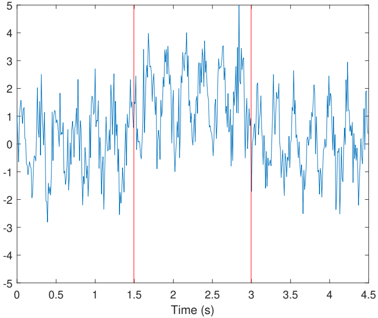


Change-Point Detection

Given a time series x, retrieve the times $(t_1, \ldots, t_K)$ where a significant change occurs

- Necessitates to estimate both the change-points and the number of changes K
- Highly depends on the meaning given to *change*

Problem statement



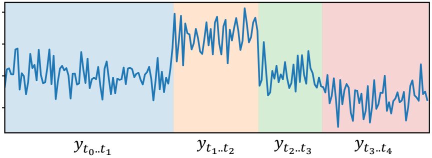
- Let assume that signal $x[n]$ undergoes abrupt changes at times

$$\mathcal{T}^* = (t_1^*, \ldots, t_{K^*}^*)$$

- Goal: retrieve the number of change-points K^* and their times $\mathcal{T}^*$
- Two assumptions: offline segmentation (but can easily be adapted to online setting) [Truong et al., 2020] and known number of changes K (will be discussed later)

Problem statement

$$(\hat{t}_1, \ldots, \hat{t}_K) = \underset{(t_1,\ldots,t_K)}{\operatorname{argmin}} \sum_{k=0}^{K} c(x[t_k : t_{k+1}])$$

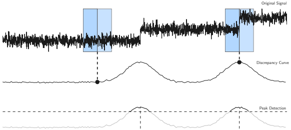


Cost function $c(.)$

- Measures the homogeneity of the segments
- Choosing $c(.)$ conditions the type of change-points that we want to detect
- Often based on a probabilistic model for the data

Problem solving

- Optimal resolution with dynamic programming
- Approximate resolution (sliding windows...)



Contents

Cost function

$$(\hat{t}_1, \ldots, \hat{t}_K) = \underset{(t_1, \ldots, t_K)}{\operatorname{argmin}} \sum_{k=0}^{K} c(x[t_k : t_{k+1}])$$

Convention : $t_0 = 0$, $t_{K+1} = N$

- Function $c(.)$ is characteristic of the notion of *homogeneity*
- The most common cost functions are linked to parametric probabilistic models: in this case change-points are defined as changes in the parameters of a probability density function [Basseville et al., 1993]
- Non parametric cost functions can also be introduced when no model is available

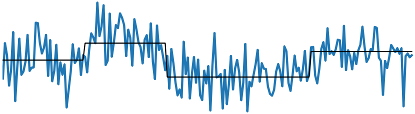
Change in mean

The most popular is indubitably the L2 norm [Page, 1955]

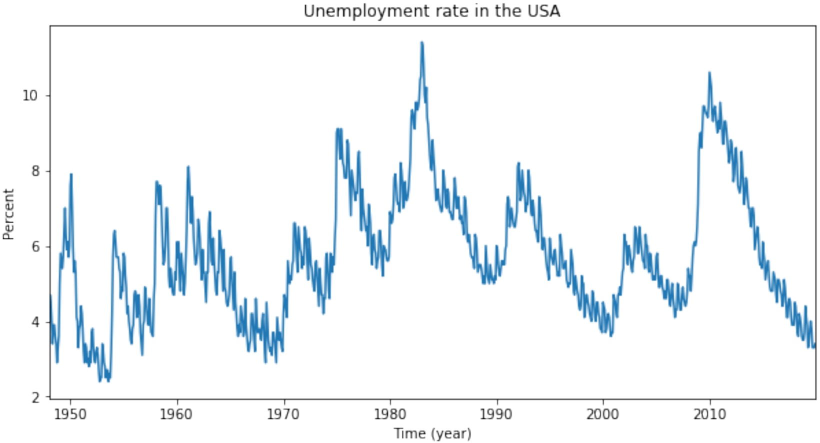
$$c_{L_2}(x[a:b]) = \sum_{n=a+1}^{b} \|x[n] - \mu_{a:b}\|_2^2$$

where $\mu_{a:b}$ is the empirical mean of the segment $x[a:b]$.

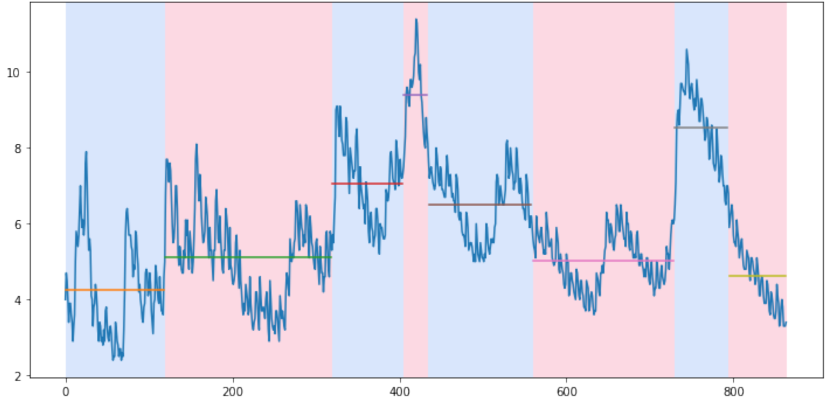
- Particular case of c_{ML} with Gaussian model with fixed variance
- Allows to detect changes in mean



Example

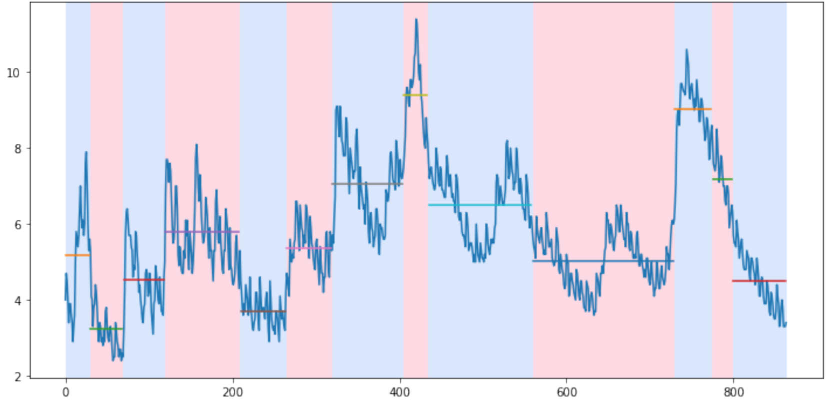

Unemployment rate in the USA
Percent
Time (year)
2
4
6
8
10
1950
1960
1970
1980
1990
2000
2010

Example: Change-Point Detection with c_{L_2}



$K = 7$

Example: Change-Point Detection with c_{L_2}



$$K = 12$$

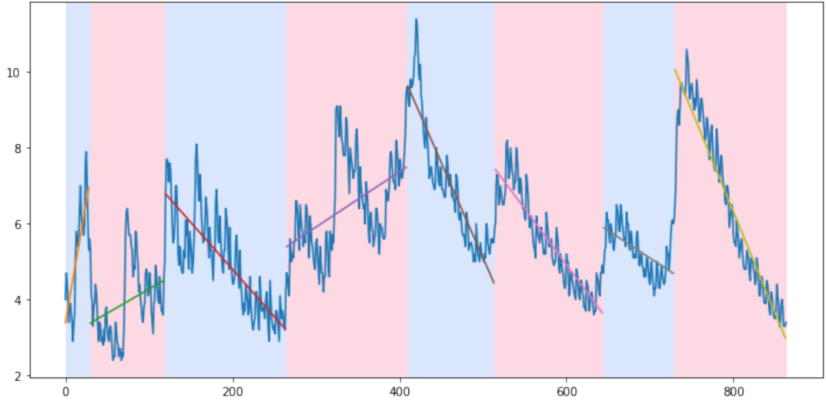
Change in slope and intercept

Change in slope and intercept can be handled in the general context of piecewise linear regression

$$c_{linear}(x[a:b]) = \min_{\boldsymbol{\alpha}} \sum_{n=a+1}^{b} \left\| x[n] - \sum_{i=1}^{M} \alpha_i \beta_i[n] \right\|_2^2$$

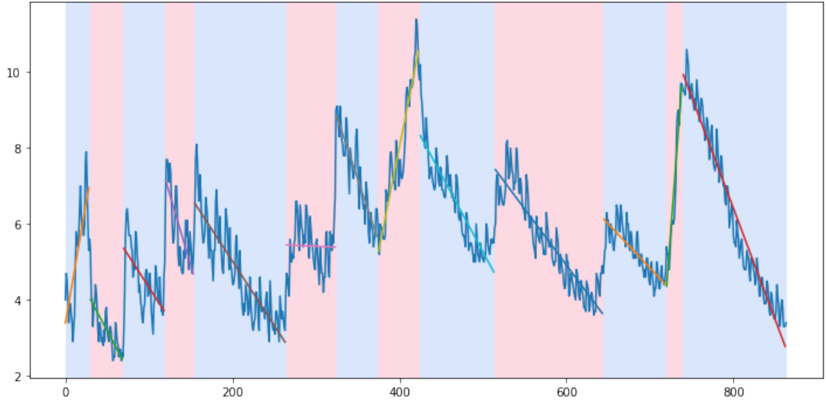
- Functions $\beta_1[n], \ldots, \beta_M[n]$ are covariate functions and we seek for changes in the regression parameters
- Allows to detect changes in trend, seasonality, etc... [Bai et al., 1998]
- For slope and intercept, we choose $\beta_1[n] = 1$ and $\beta_2[n] = n$

Example: Change-Point Detection with c_{linear}



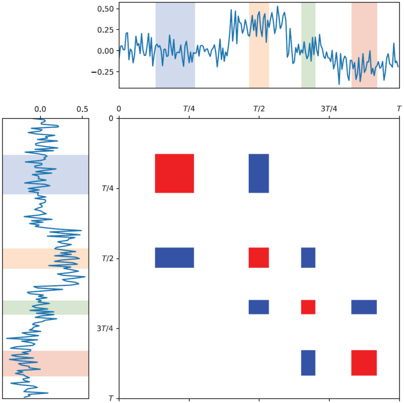
$K = 7$

Example: Change-Point Detection with c_{linear}



$K = 12$

Supervised change-point detection



- Supervised approach based on metric learning [Truong et al., 2019]
- Use a few annotated examples to learn an adequate cost function
- Samples belonging to adjacent regimes form a set of constraints that can be used for metric learning
- Allows to relax the need for an off-the-shelf cost function

C. Truong, L. Oudre and N. Vayatis. Supervised kernel change point detection with partial annotations. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pages 3147-3151, Brighton, UK, 2019.

Contents

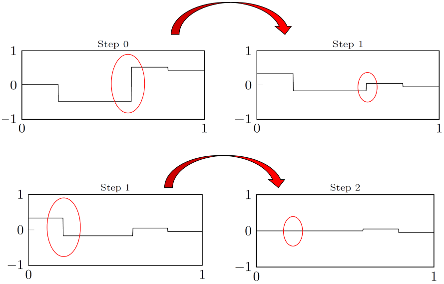
Search method

$$(\hat{t}_1, \ldots, \hat{t}_K) = \underset{(t_1,\ldots,t_K)}{\operatorname{argmin}} \sum_{k=0}^{K} c(x[t_k : t_{k+1}])$$

Convention : $t_0 = 0$, $t_{K+1} = N$

- Several methods can be used to solve this problem with a fixed K
- Optimal resolution with dynamic programming: find the true solution of the problem (but costly : Complexity of $\mathcal{O}(N^2)$)
- Approximated resolution with windows: test for one unique change-point on a window (necessitates some extra parameters and less precise)

Greedy change-point detection



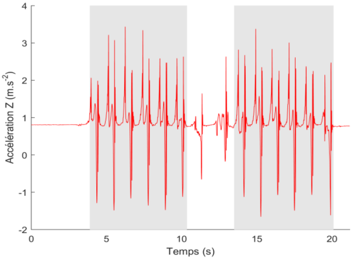
- Step 1 : Detection of one unique change
  Breakpoint is set when the left and right subsignals are the most homogeneous
- Step 2 : Projection to remove the breakpoint
  Breakpoint is smoothed and the process is re-iterated

- Algorithm stops when K breakpoints have been detected
- Wide class of cost functions can be used (included kernels [Truong et al., 2019])
- Each detection/projection has linear complexity
- Consistency results have been proved

C. Truong, L. Oudre, N. Vayatis. Greedy kernel change point detection. IEEE Transactions on Signal Processing, 67(24):6204-6214, 2019.

Results on gait analysis

- 262 gait protocols from various subjects (healthy, neurological, orthopedic, stroke...)
- 2D-signals: vertical acceleration and angular velocity around the vertical axis from the lower back sensor
- Aim: retrieve the different phases of the protocol (stop, walk, U-turn, walk, stop)



- Maximal error: 1.13 seconds vs. more than 3 seconds for approached methods
- Approximately the same maximal error than optimal resolution (1.13 s vs 1.29 s)... but way faster (1 min 30 vs. > 2 days)

Contents

Finding the number of change points

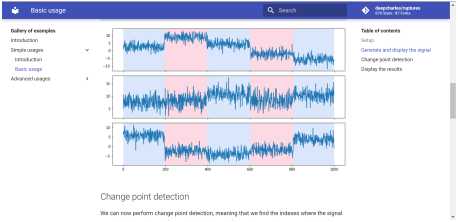
- In all previously described algorithms, the number of change-point K was supposed to be known
- In practice, this parameter is difficult to set: as such, the total cost $\mathcal{V}(\mathcal{T}, \mathbf{x})$ will always decrease when K increases...
- Three solutions
  - Use heuristics by testing several values of K
  - Use a penalized formulation of the CPD problem to seek for a compromise between reconstruction error and complexity
  - Use supervised approaches from annotated signals [Truong et al., 2018]

C. Truong, L. Oudre and N. Vayatis. Penalty Learning for Changepoint Detection. In Proceedings of the European Signal Processing Conference (EUSIPCO), pages 1614-1618, Kos Island, Greece, 2017.

Contents

The `ruptures` package

ruptures

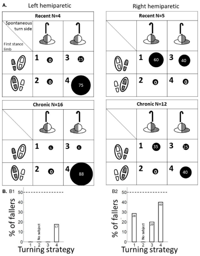


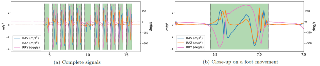
https://centre-borelli.github.io/ruptures-docs/

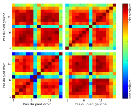
- Package in Python implementing most of the offline approaches for change-point detection
- Many cost functions, resolution algorithms and parametrizations
- More than 1 Million downloads!

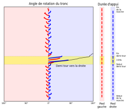
C. Truong, L. Oudre, N. Vayatis. Selective review of offline change point detection methods. Signal Processing, 167:107299, 2020

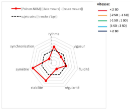
Use for biomedical research

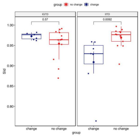












- Study of U-turn for post-stroke patients [Barrois-Müller et al., 2017]
- Step analysis for multiple sclerosis patients [Vienne-Jumeau et al., 2020]
- Comparison of gait exercises through pattern matching techniques [Vienne-Jumeau et al., 2019]

Thank you for your attention