Machine Learning for Time Series

Lecture 5: Change-point and Anomaly Detection

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

Master MVA
2023-2024

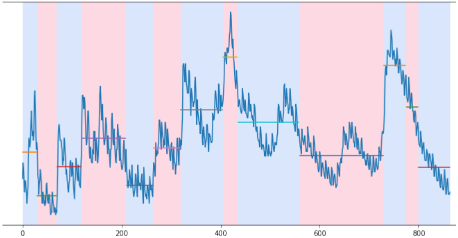
Contents

Contents

Discovering events in time series

- Time series are widely used for monitoring: finance, industry, healthcare, meteorology...
- When recorded for hours, days, weeks... data is likely to be redundant
- Two fundamental questions:
  - Were there significant changes in my data across time?
  - Was there something new or unusual in my data?
- Two ill-posed problems: what is a *significant change?* What is *new* or *unusual*?

Problem 1: Change-Point Detection

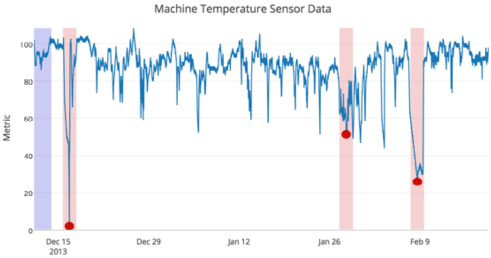


Change-Point Detection

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 but also the number of changes K
- Highly depends on the meaning given to change

Problem 2: Anomaly Detection



Anomaly Detection

Given a time series x, retrieve the set of samples $\mathcal{T}$ that corresponds to unusual phenomenon

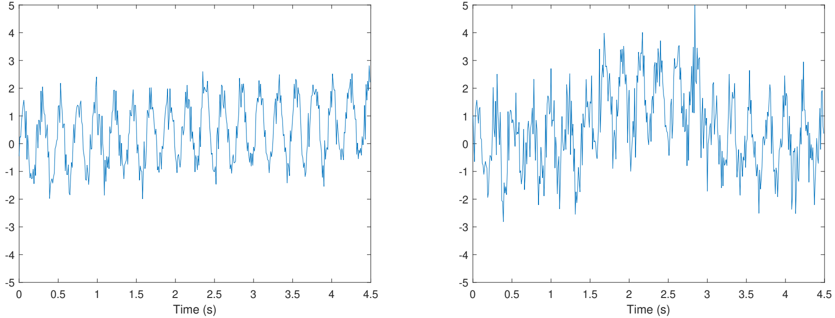
- May include isolated or contiguous samples (see Lecture 4 on outlier detection/removal)
- Highly depends on the meaning given to usual/unusual

Contents

Importance of stationarity

- As seen in Lecture 2, stationarity (e.g. at the wide sense) is a fundamental assumption when processing time series
- Necessity when using DFT, autocorrelation function or extracting features
- When observed during a long period of time, the system behaviour monitored by time series is likely to change over time, either smoothly or abruptly
- Several strategies can be used to deal with non-stationary time series, either simple or complex
- In some context, knowledge on these abrupt changes also carries relevant information on the system

Example



Smooth evolution vs. abrupt change

How to bypass the problem

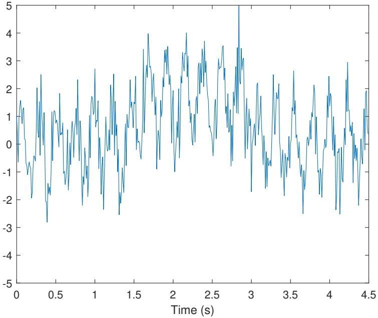
In a first order approximation, several strategies can be used to bypass the problem:

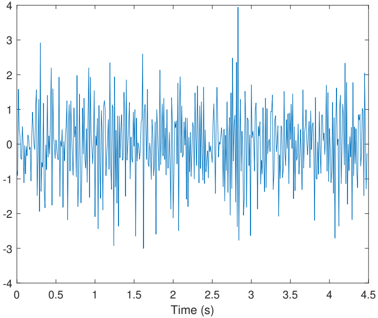
- If changes are smooth, the task can be seen as a detrending task: remove the slow phenomenon and only keep the seasonality that may be more stationary
- Divide the signal into small frames on which the signal is assumed to be stationary (see Lecture 2 on spectrogram)
- Instead of working on the original signal $x[n]$, we can work on the signal derivative

  $$x'[n] = x[n] - x[n-1]$$

  which in general has nicer stationarity properties
  Careful! This can imply to re-integrate the signal after processing, which can be a source of errors!

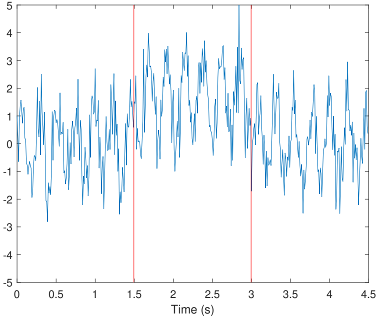
Example





Use of derivation to make the signal *more stationary*
Left : original signal / Right : first order derivative

Problem statement



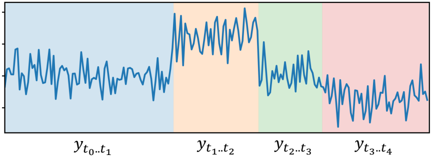
- When the changes are abrupt or when the estimation of the change-points is relevant in the context, we can use **change-point detection** methods
- Let assume that signal $x[n]$ undergoes abrupt changes at times

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

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

Problem statement

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

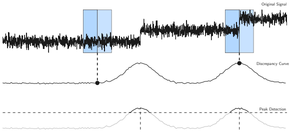


Cost function $c(.)$

- 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...)



Cost function

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

Convention : $t_0 = 0$, $t_{K+1} = N$
$a : b = [a, a+1, \ldots, b-1]$

- 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

Maximum likelihood estimation

Given a parametric family of distribution densities $f(\cdot|\theta)$ parametrized with $\theta \in \Theta$, a cost function can be derived:

$$c_{ML}(x[a:b]) = -\sup_{\theta} \sum_{n=a+1}^{b} \log f\left(x[n]|\theta\right)$$

- Corresponds to the assumption that on a regime, samples are i.i.d. according to a parametric distribution density
- On each regime, the parameters are estimated through maximum likelihood estimation
- This model can be adapted to several situations: change in mean, change in variance, change in both mean and variance...

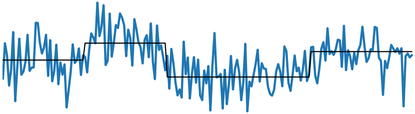
Change in mean

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

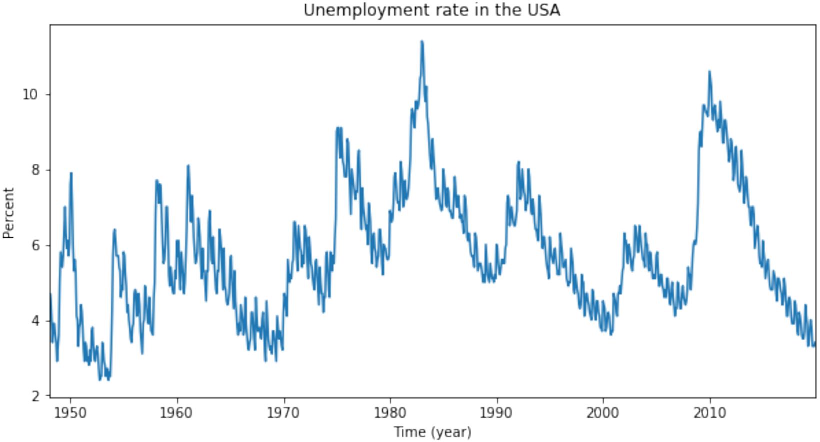
$$c_{L_2}(x[a:b]) = \sum_{n=a+1}^{b} \|x[n] - \mu_{a:b}\|_2^2$$

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

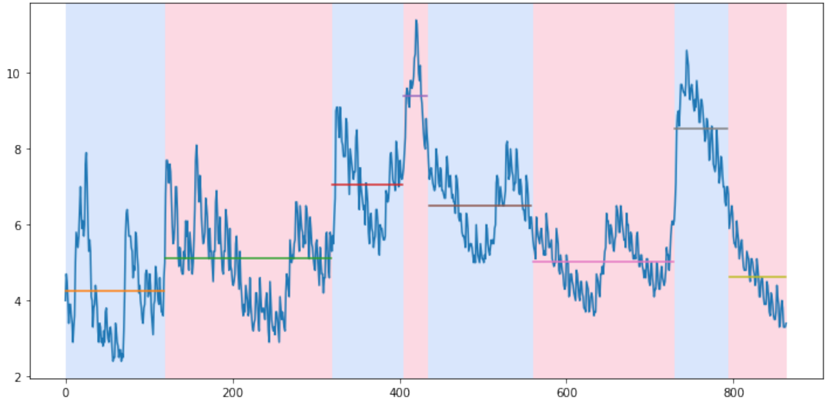
- Particular case of c_{ML} with Gaussian model with fixed variance
- Allows to detect changes in mean



Example

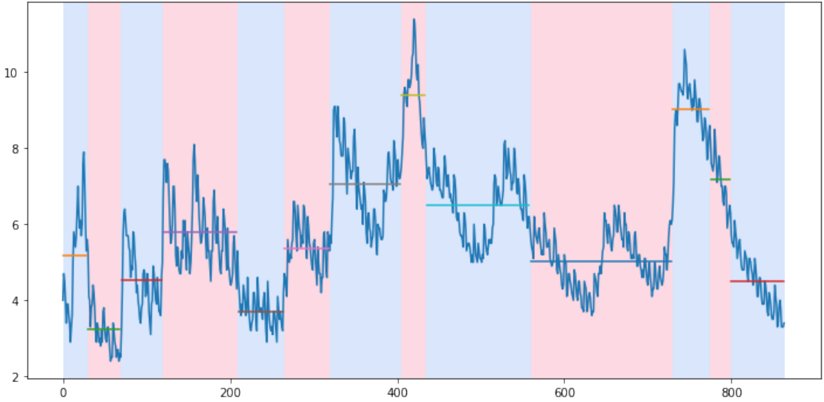

Unemployment rate in the USA
Percent
10
8
6
4
2
1950
1960
1970
1980
1990
2000
2010
Time (year)

Example: Change-Point Detection with c_{L_2}



$K = 7$

Example: Change-Point Detection with c_{L_2}



$K = 12$

Change in mean and variance

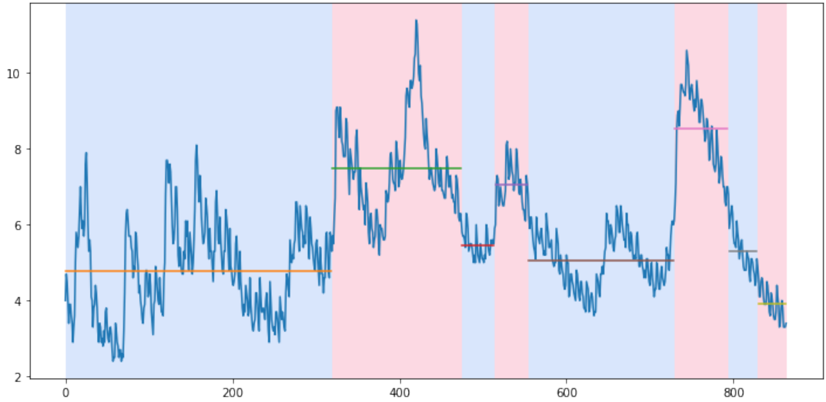
When the mean and variance change over time the cost function becomes

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

where $\mu_{a:b}$ and $\sigma^2_{a:b}$ are the empirical mean / variance of the segment $x[a:b]$.

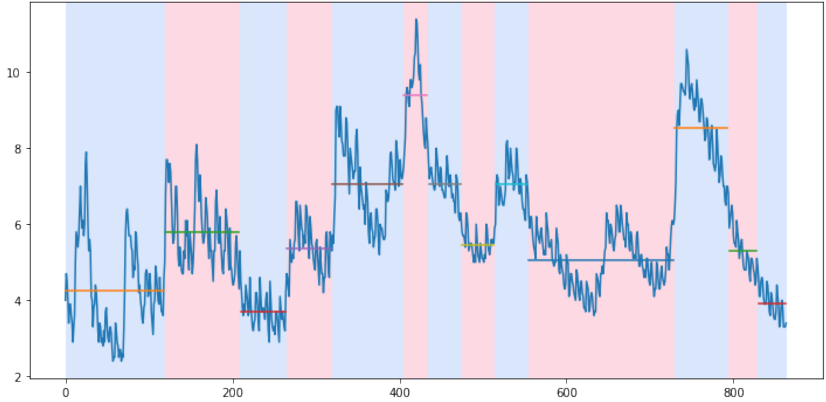
- Particular case of c_{ML} with Gaussian model with unknown mean and variance
- Can be adapted to multivariate time series by replacing the variance by the covariance matrix: in this case, changes of correlations between dimensions can also be detected [Lavielle, 1999]

Example: Change-Point Detection with c_Σ



$K = 7$

Example: Change-Point Detection with c_Σ



$K = 12$

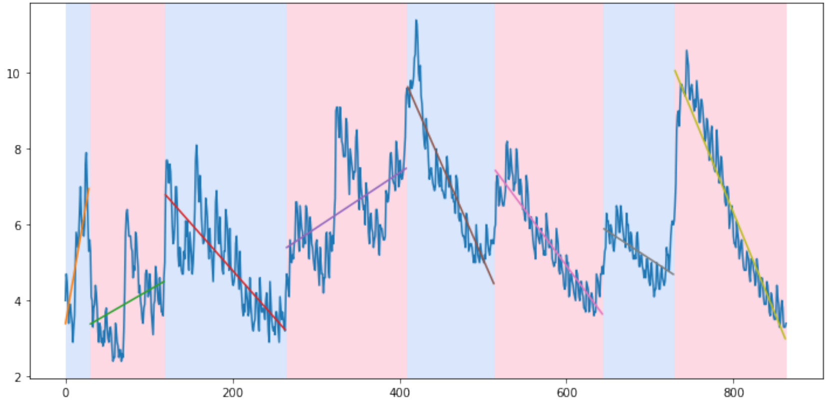
Change in slope and intercept

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

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

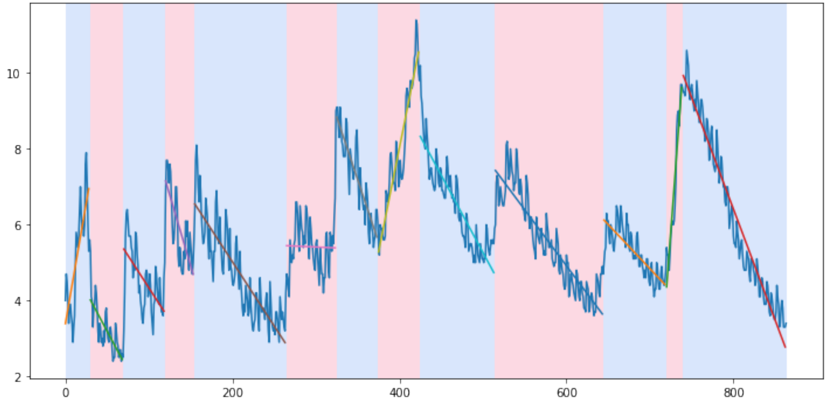
- Functions $\beta_1[n], \ldots, \beta_M[n]$ are covariate functions and we seek for changes in the regression parameters
- Allows to detect changes in trend, seasonality, etc... [Bai et al., 1998]
- For slope and intercept, we choose $\beta_1[n] = 1$ and $\beta_2[n] = n$

Example: Change-Point Detection with c_{linear}



$$K = 7$$

Example: Change-Point Detection with c_{linear}



$K = 12$

Rank-based cost functions

- In order to remove the need for a parametric probability model, one trick is to work on the notion of *rank* instead of the whole signal

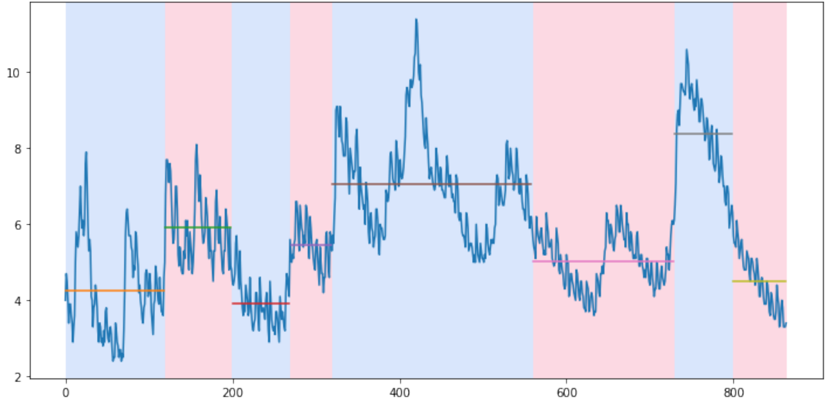
$$r[n] = \text{number of } i \text{ such that } x[i] < x[n]$$

- Robust and invariant with respect to amplitude changes: $r[n]$ corresponds to the rank of sample $x[n]$ in the time series $\mathbf{x}$
- Cost functions can be derived by detecting changes in mean and/or variance in the rank signal [Lung-Yut-Fong et al., 2015]

$$c_{rank}(x[a:b]) = \sum_{n=a+1}^{b} \|r[n] - \mu^r_{a:b}\|_2^2$$

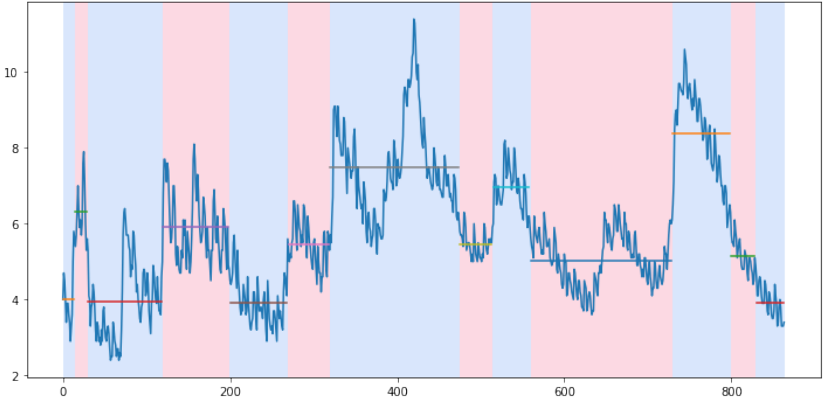
where $\mu^r_{a:b}$ is the empirical mean of the rank signal $r[a:b]$

Example: Change-Point Detection with c_{rank}



$K = 7$

Example: Change-Point Detection with c_{rank}



$$K = 12$$

Search method

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

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

- 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)
- Approximated resolution with windows: test for one unique change-point on a window

Optimal resolution

- By denoting

$$\mathcal{V}(\mathcal{T}, \mathbf{x}) = \sum_{k=0}^{K} c(x[t_k : t_{k+1}])$$

we can see that

$$\begin{aligned}
\min_{|\mathcal{T}|=K} V(\mathcal{T}, \mathbf{x}) &= \min_{0=t_0<t_1<\cdots<t_K<t_{K+1}=N} \sum_{k=0}^{K} c(x[t_k : t_{k+1}]) \\
&= \min_{t\leq N-K} \left[c(x[0:t]) + \min_{t_0=t<t_1<\cdots<t_{K-1}<t_K=N} \sum_{k=0}^{K-1} c(x[t_k : t_{k+1}]) \right] \\
&= \min_{t\leq N-K} \left[c(x[0:t]) + \min_{|\mathcal{T}|=K-1} V(\mathcal{T}, x[t:N]) \right]
\end{aligned}$$

- Recursive problem (just like DTW in Lecture 1): resolution with dynamic programming [Bai et al., 2003]
- Two steps: computation of the cumulative costs + determination of the change-points

Optimal resolution

Algorithm 1 Algorithm `Opt`

Input: signal $\{y_t\}_{t=1}^T$, cost function $c(\cdot)$, number of regimes $K \geq 2$.
for all (u, v), $1 \leq u < v \leq T$ **do**
  Initialize $C_1(u, v) \leftarrow c(\{y_t\}_{t=u}^v)$.
end for
for $k = 2, \ldots, K-1$ **do**
  for all $u, v \in \{1, \ldots, T\}, v - u \geq k$ **do**
    $C_k(u, v) \leftarrow \min_{u+k-1 \leq t < v} C_{k-1}(u, t) + C_1(t+1, v)$
  end for
end for
Initialize L, a list with K elements.
Initialize the last element: $L[K] \leftarrow T$.
Initialize $k \leftarrow K$.
while $k > 1$ **do**
  $s \leftarrow L(k)$
  $t^* \leftarrow \text{argmin}_{k-1 \leq t < s} C_{k-1}(1, t) + C_1(t+1, s)$
  $L(k-1) \leftarrow t^*$
  $k \leftarrow k - 1$
end while
Remove T from L
Output: set L of estimated breakpoint indexes.

Complexity of $\mathcal{O}(KN^2)$

Approximated resolution

- Main limitation of optimal resolution: high complexity. Prohibitive for long time series...
- Approximated resolution methods exist, which are based on the single change-point detection, which is way easier to perform
- Idea: consider a sliding window of length $2w$ and for each position, determine if there is a change or not
- How to detect a single change by using the cost functions?

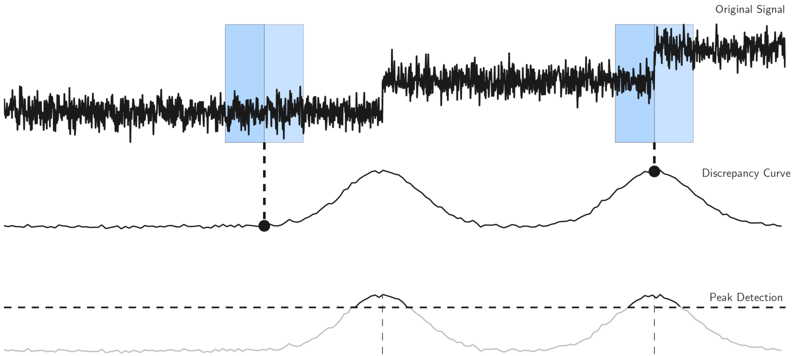
Discrepancy function

Given a window of length $2w$ centered on sample n, we compute the discrepancy function

$$d[n] = c(x[n-w:n+w]) - c(x[n:n+w]) - c(x[n-w:n])$$

- The discrepancy function $d[n]$ allows to compare
  - The homogeneity of the whole window $c(x[n-w:n+w])$
  - The homogenities of the right/left windows $c(x[n:n+w])$, $c(x[n-w:n])$
- Intuitively, if a change-point occurs at time n and if the window length w is well adapted, both subsegments $x[n-w:n]$ and $x[n:n+w]$ will be homogeneous (i.e. small values) and the whole segment $x[n-w:n+w]$ will be heterogeneous (i.e. large values)
- Large values for $d[n]$ suggests that a change-point is likely to appear at time n

Discrepancy function



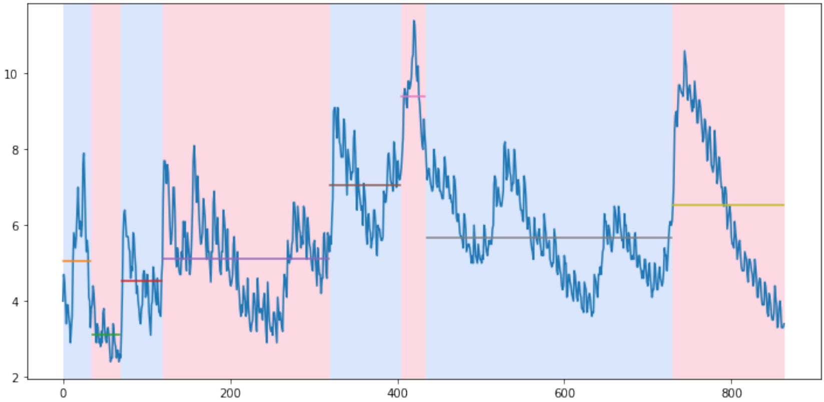
Sliding window approximated resolution

Algorithm 2 Algorithm `Win`

Input: signal $\{y_t\}_{t=1}^{T}$, cost function $c(\cdot)$, half-window width w, peak search procedure `PKSearch`.
Initialize $Z \leftarrow [0, 0, \dots]$ a T-long array filled with 0. ▷ Score list.
for $t = w, \dots, T - w$ **do**
  $p \leftarrow (t - w)..t$.
  $q \leftarrow t..(t + w)$.
  $r \leftarrow (t - w)..(t + w)$.
  $Z[t] \leftarrow c(y_r) - [c(y_p) + c(y_q)]$.
end for
$L \leftarrow$ `PKSearch`(Z) ▷ Peak search procedure.
Output: set L of estimated breakpoint indexes.

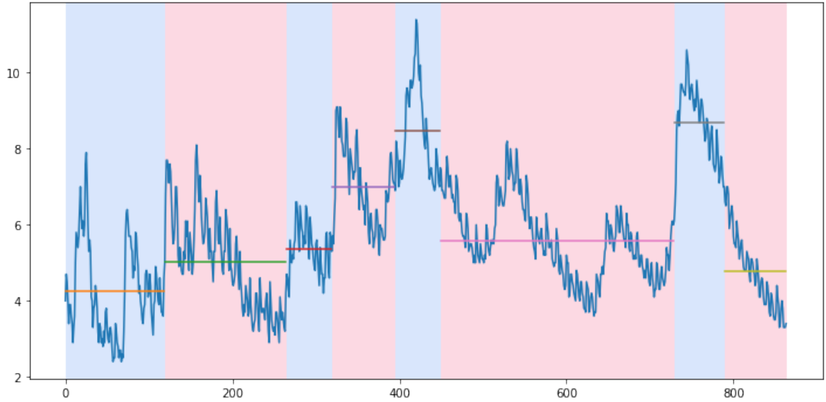
- Computation of the discrepancy function + peak search procedure to detect the K largest peaks
- Complexity of $\mathcal{O}(N)$

Example: Sliding window CPD with c_{L_2}



$$K = 7, w = 20$$

Example: Sliding window CPD with c_{L_2}



$K = 7, w = 40$

How to set w

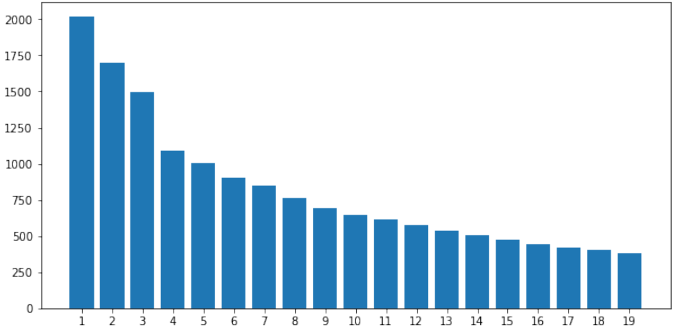
- Parameter w should correspond to the smallest length of stationarity: the discrepancy function makes sense if the two subsegments (right and left) are homogeneous
- Window length is also necessarily smaller than the smallest regime length
- But careful! In order to be relevant, the window length should be large enough so that there is enough samples to properly estimate the homogeneity
- Other vision: statistical tests between two sets of samples, each containing w samples. Good estimation requires a sufficient number of samples.

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

Heuristics for finding the number of change-points

- One easy solution is to test a set of change-points number K from 1 to K_{max} and to compute the sum of costs $\mathcal{V}(\mathcal{T}, \mathbf{x})$
- The *optimal* number of change-points can be estimated by searching for an elbow on the curve of $\mathcal{V}(\mathcal{T}, \mathbf{x})$ as a function of K



Penalized Change-Point Detection

- Intuitively, the optimal number of change-points is the one that allows the best compromise between the sum of costs $\mathcal{V}(\mathcal{T}, \mathbf{x})$ and the number of ruptures $|\mathcal{T}|$
- We actually had the same problem in various tasks: order estimation in AR models (Lecture 3), number of atoms in dictionary learning (Lecture 3 & 4), etc... In all cases the higher the order (and the number of parameters), the better the reconstruction
- **Model selection problem**: find the best model among a class of models
- Penalized change-point detection

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

Penalized change-point detection

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

- Joint estimation of the change-point times and the number of change-points
- Parameter β penalizes the introduction of a new change-point in the model: an additional change-point should decrease the sum of costs $\mathcal{V}(\mathcal{T}, \mathbf{x})$ by at least β
- Luckily, this problem is even easier to solve than the original one with fixed K!

Pruning strategy

- Given two times s and t such that $t < s < N$, remark that if

$$\min_{\mathcal{T}} \left[V(\mathcal{T}, x[0:t]) + \beta|\mathcal{T}| \right] + c(x[t:s]) \geq \min_{\mathcal{T}} \left[V(\mathcal{T}, x[0:s]) + \beta|\mathcal{T}| \right]$$

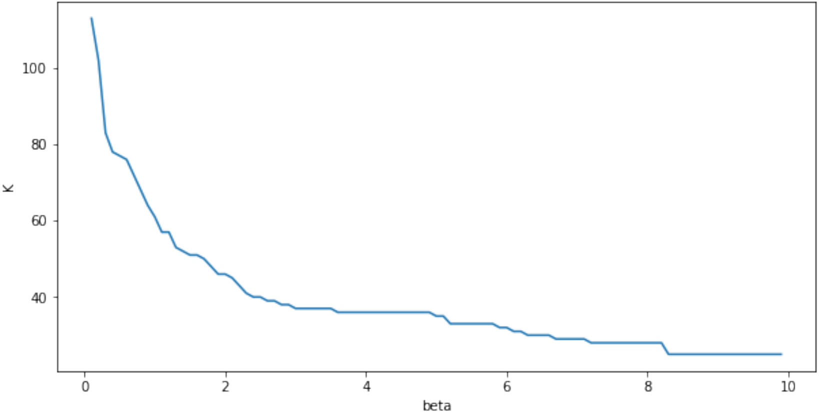
  then t cannot be the last change point prior to N (demo in the last slides).
- Considerable speed-up since most times will not satisfy this criterion
- **Pruned Exact Linear Time (PELT)** algorithm: under the assumption that regime lengths are randomly drawn from a uniform distribution, the complexity of PELT is $\mathcal{O}(N)$ [Killick et al., 2012]
- Optimal algorithm: exact solution

PELT algorithm

Algorithm 3 Algorithm `Pelt`

Input: signal $\{y_t\}_{t=1}^{T}$, cost function $c(\cdot)$, penalty value β.
Initialize Z a $(T+1)$-long array; $Z[0] \leftarrow -\beta$.
Initialize $L[0] \leftarrow \emptyset$.
Initialize $\chi \leftarrow \{0\}$. ▷ Admissible indexes.
for $t = 1, \ldots, T$ **do**
  $\hat{t} \leftarrow \operatorname{argmin}_{s \in \chi} \left[Z[s] + c(y_{s..t}) + \beta \right]$.
  $Z[t] \leftarrow \left[Z[\hat{t}] + c(y_{\hat{t}..t}) + \beta \right]$
  $L[t] \leftarrow L[\hat{t}] \cup \{\hat{t}\}$.
  $\chi \leftarrow \{s \in \chi \,:\, Z[s] + c(y_{s..t}) \leq Z[t]\} \cup \{t\}$
end for
Output: set $L[T]$ of estimated breakpoint indexes.

How to choose β?



β can be difficult to set: no explicit formula between β and K

Model selection criterion

Two popular criteria can be used to estimate the relevance of a model

- **Bayesian information criterion (BIC)** [Schwarz, 1978]

$$\mathrm{BIC} = k \log N - 2 \log \hat{L}$$

- **Akaike information criterion (AIC)** [Akaike, 1974]

$$\mathrm{AIC} = 2k - 2 \log \hat{L}$$

where

- k is the number of parameters
- N is the number of samples
- $\hat{L}$ is the maximum value of the likelihood function for the model

Standard criterion

In the context of change-point detection with L2 cost function, these criteria provide estimates for the β parameter:

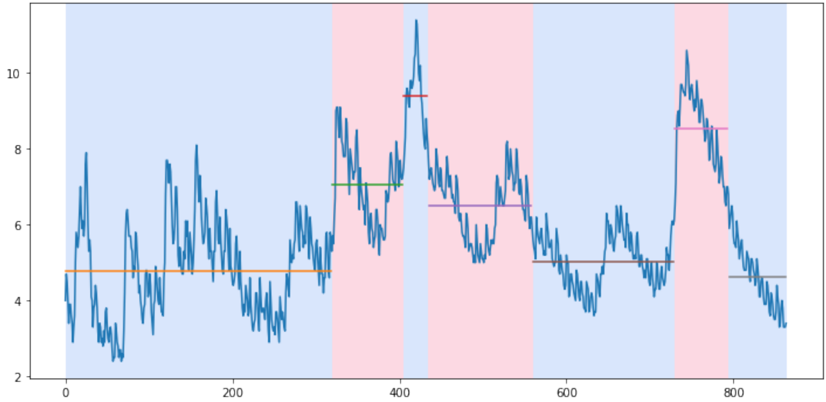
- Bayesian information criterion (BIC) for $L2$ change-point detection

$$\beta = 4\sigma^2 \log N$$

- Akaike information criterion (AIC) for $L2$ change-point detection

$$\beta = 4\sigma^2$$

Example



Results obtained with the BIC criterion

Supervised Change-Point Detection

- Parameter β can also be learned from a collection of annotated signals $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}$ with annotations $\mathcal{T}_*^{(1)}, \dots, \mathcal{T}_*^{(M)}$ [Truong et al., 2017]
- By denoting

$$V_\beta(\mathcal{T}, \mathbf{x}) = \sum_{k=0}^{K} c(x[t_k : t_{k+1}]) + \beta K$$

  the relevance of a value of β can be assessed by computing the excess penalized risk

$$E(\mathbf{x}^{(\ell)}, \beta) = V_\beta(\mathcal{T}_*^{(\ell)}, \mathbf{x}^{(\ell)}) - \min_{\mathcal{T}} V_\beta(\mathcal{T}, \mathbf{x}^{(\ell)})$$

- This quantity reflects how well we can approach the annotated segmentation with a given value of β

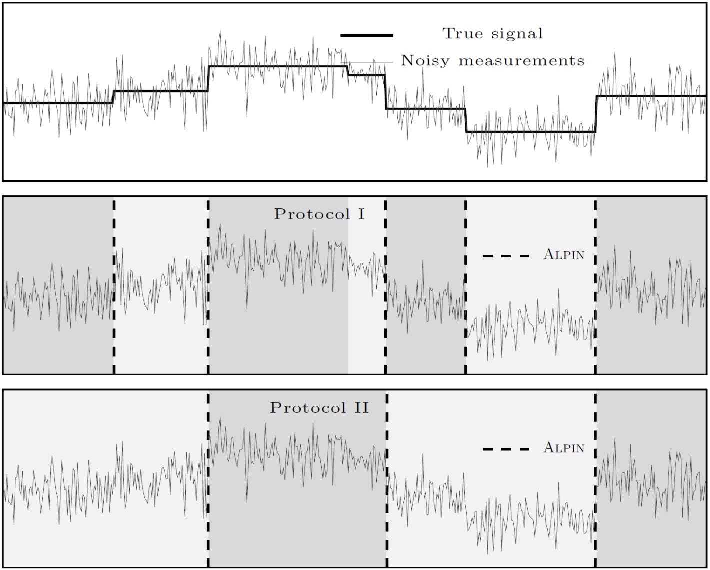
Supervised Change-Point Detection

- The function $\beta \mapsto E(\mathbf{x}^{(\ell)}, \beta)$ is a convex function of β, which can be easily optimized with off-the-shelf solvers
- The final optimization problem writes

$$\beta_{opt} = \text{argmin}_{\beta > 0} \frac{1}{M} \sum_{\ell=1}^{M} E(\mathbf{x}^{(\ell)}, \beta)$$

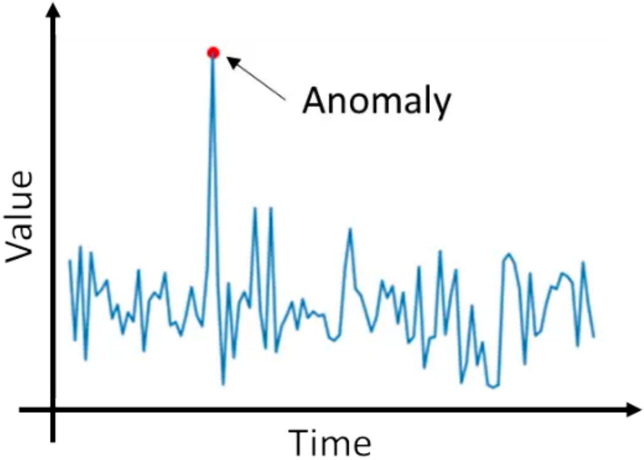
- Experimental results show that with only a few annotated examples, it is possible to find an adequate range for β

Example


True signal
Noisy measurements
Protocol I
ALPIN
Protocol II
ALPIN

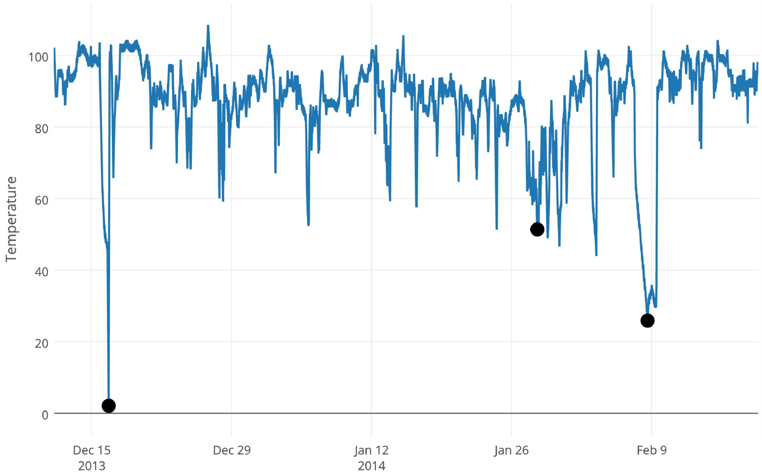
Contents

Introductory example



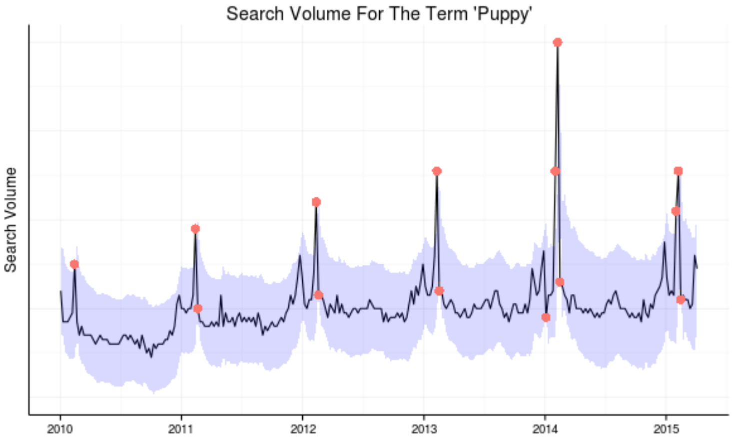
Easy: an anomaly is a too small or too large value (outlier)

Introductory example



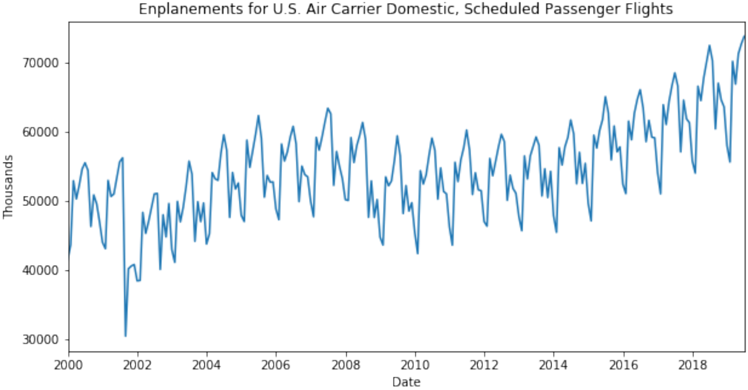
More complex: some small/large values are anomalies

Introductory example



Anomalies depend in the previous values

Introductory example



Anomalies correspond to unusual events

Anomaly Detection

Anomalies can take various forms and have different meanings [Chandola et al., 2009]:

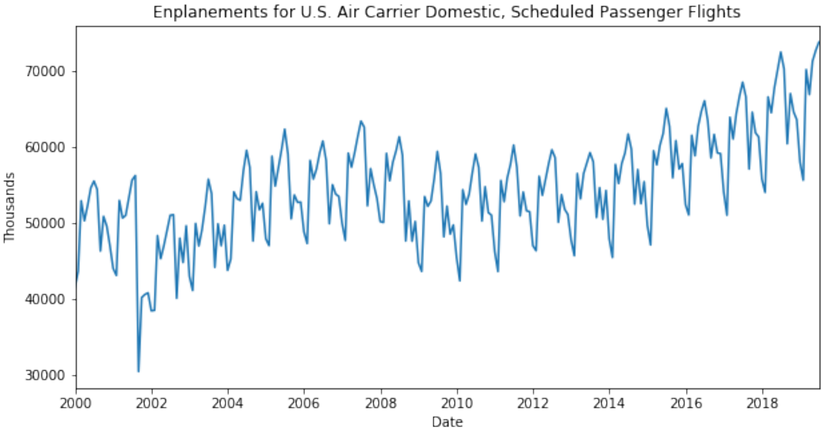
- Outliers, i.e. isolated samples with exceptionally large/low values
- Bursts of outliers, i.e. segments that do not coincide with what is observed usually in the time series (in terms of values)
- Unusual events that breaks the regularity within the time series

Outlier detection

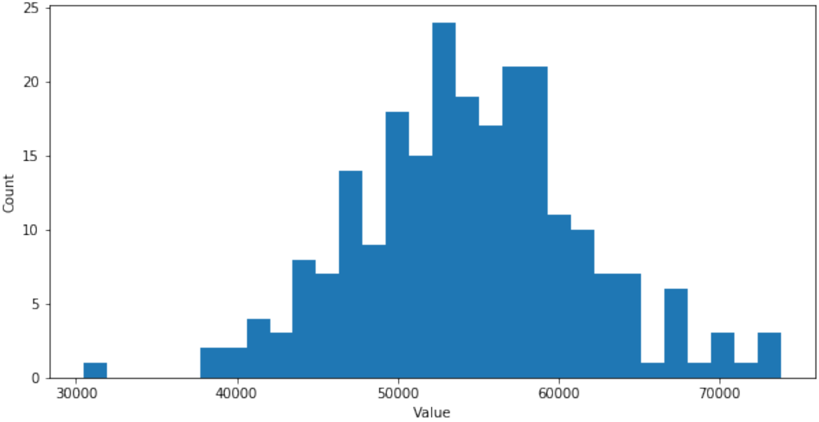
Simple anomalies (isolated samples or contiguous samples) can be detected with techniques already described in Lecture 4:

- Statistical methods:
  - Global: Histogram visualization to detect aberrant values (see Lecture 4)
  - Adaptive: Threshold-based methods on sliding windows (mean/standard deviation or median)
- Model-based methods:
  - Residual and prediction error (trend+seasonality, sinusoidal or AR model)

Example

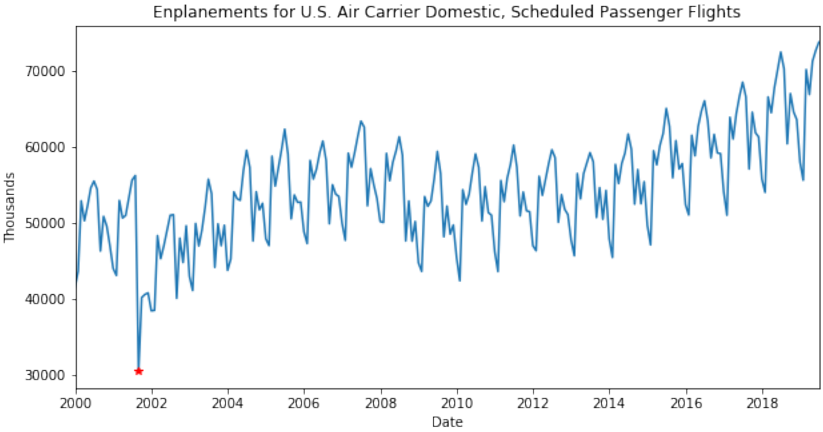

Enplanements for U.S. Air Carrier Domestic, Scheduled Passenger Flights
Thousands
70000
60000
50000
40000
30000
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
Date

Example: Histogram



One outlier can be considered as an anomaly

Example: Histogram



Only one detected anomaly

Adaptive statistical methods

- The main idea is to use sliding window and to perform a statistical test for outlier detection
- Contrary to histogram, these methods allow to take into account the local context but careful, time information is lost! Only the distribution of values is used for detection.
- Multitude tests can be used but the most common are
  - **Mu/sigma** [Roberts, 2000]:

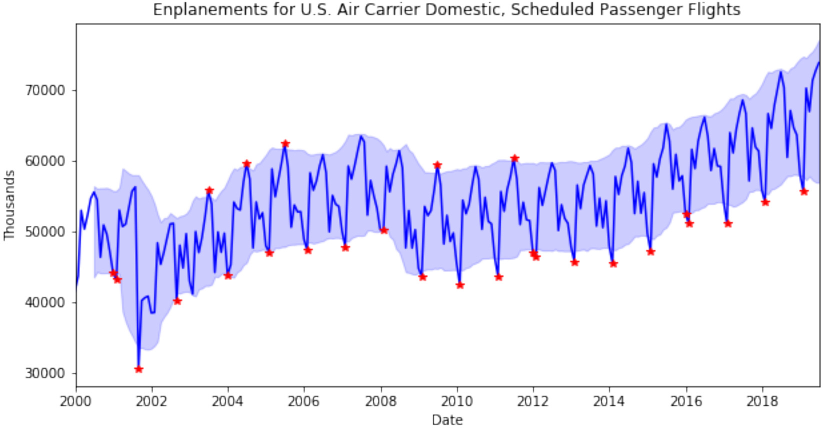
    $$|x[n] - \mu_n| > \lambda \sigma_n$$

    where μ_n and σ_n are respectively the local mean/standard deviation around sample n and λ a threshold.
    Under i.i.d. Gaussian assumption, $\lambda = 1 \rightarrow 68\%$, $\lambda = 2 \rightarrow 95\%$, $\lambda = 3 \rightarrow 99.7\%$
  - **Median/median absolute deviation** [Leys et al., 2013]:

    $$|x[n] - \mathrm{med}_n| > \lambda \, \mathrm{mad}_n$$

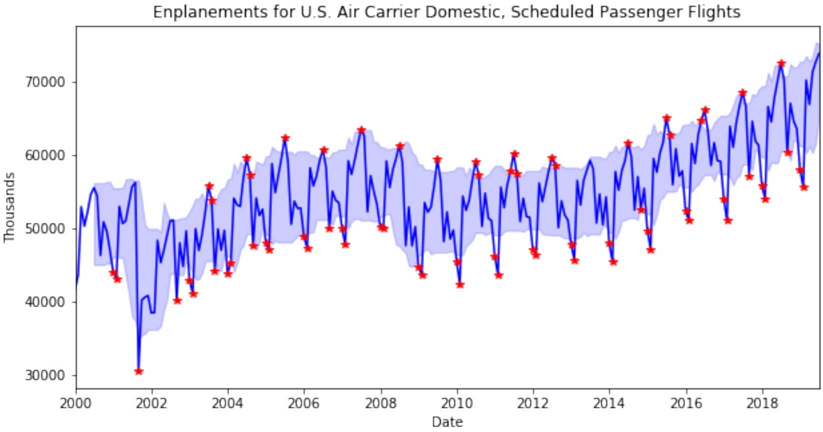
    where med_n and mad_n are respectively the local median/median absolute deviation around sample n and λ a threshold.

Example: Mu-Sigma



Mu-Sigma, $\lambda = 1.5$, window length of 12 samples

Example: Med-Mad

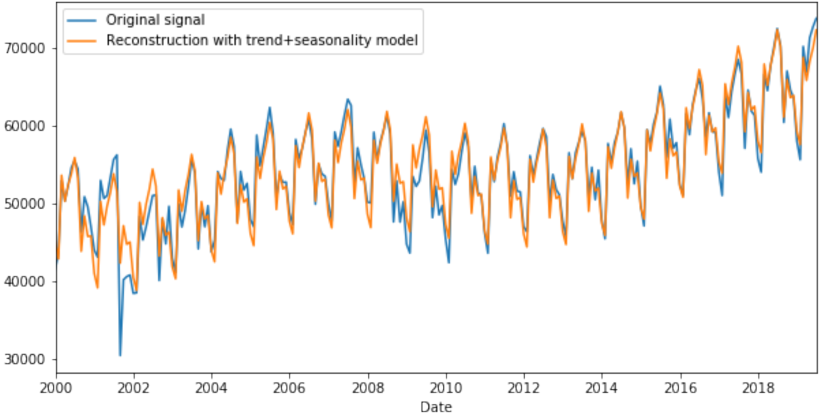


Med-Mad, $\lambda = 1.5$, window length of 12 samples

Model-based anomaly detection

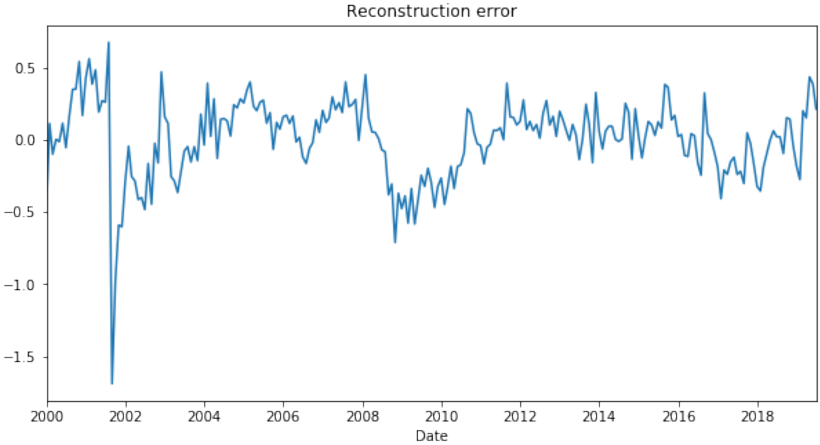
- Idea: use a time series model to detect anomaly [Yamanishi et al., 2002; Hill et al., 2010]
- Advantage: truly takes into account the temporal aspects
- Three steps:
  1. Choose an adequate model and learn the parameters
  2. Compute the prediction/signal reconstruction
  3. Anomalies are samples that diverge from the model

Example: trend+seasonality



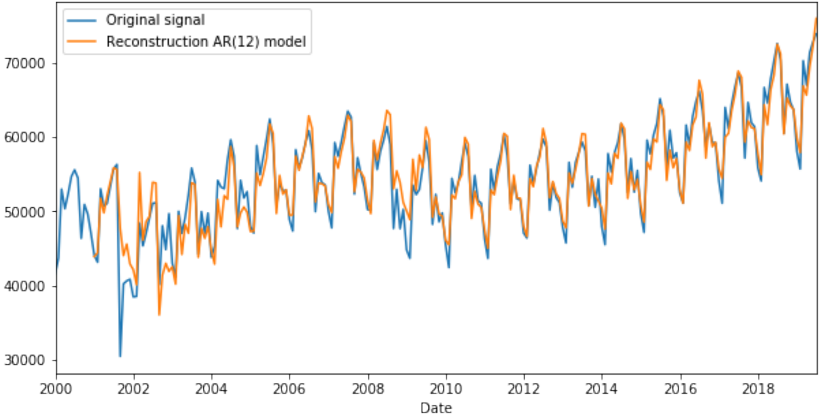
- Trend: polynomial of degree 4
- Seasonality: cosine/sine functions with fundamental frequencies multiples of $\frac{1}{12}$

Example: trend+seasonality



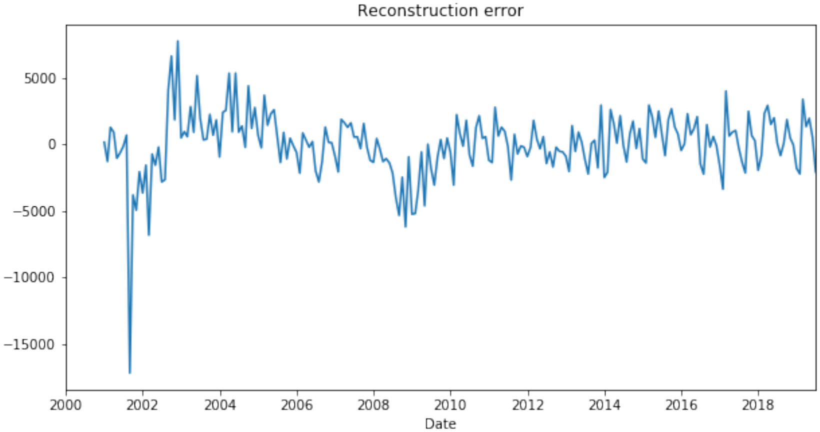
Not only large/small values but also temporal progression

Example: AR model



AR model with $p = 12$

Example: AR model



Anomaly also changes the prediction of the next p samples

Distance-based methods

- Some anomalies may be more complex to detect as they are not characterized by aberrant values but by a new behavior that was not previously seen in the time series
- In this case, anomalies can only be defined as a divergence from a normal behavior
- This task is the dual of the task already seen in Lecture 1 (Pattern Detection/Extraction), and the same techniques can therefore be used
- Instead of searching for repetitive patterns, we are searching for non-repetitive events!

Unsupervised anomaly detection

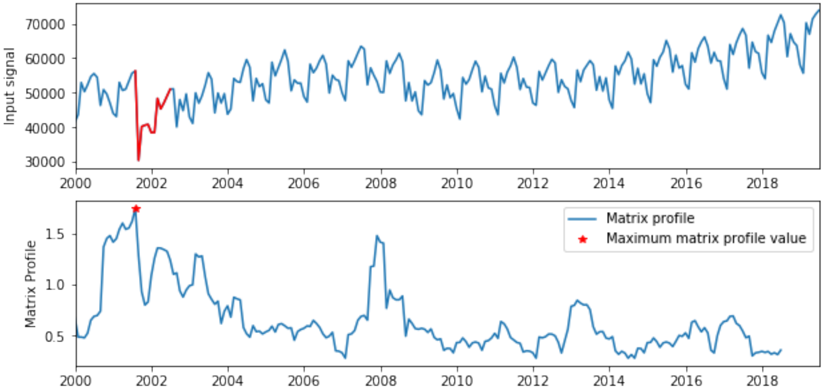
- Reminder : **Matrix profile** [Yeh et al., 2016] : given a pattern length L, compute

$$m[n] = \min_{i>n+L \text{ or } i<n-L} d(x[n : n+L-1], x[i : i+L-1])$$

- Small matrix profiles values indicate that the subsequence has been found elsewhere in the time series, suggesting that it could be a pattern
- Efficient computation with normalized Euclidean distance (see Lecture 1)

What about large values in the matrix profile?

Example: matrix profile



Matrix profile with window of length $L = 12$ months

Matrix profile

- By examining large values on the matrix profile, anomalies can be detected
- Subsequences that are *far* from all subsequences in the signal: likely to correspond to new behaviors
- Advantages: no need for a parametric model
- Necessitates to have a rough idea of the scale of the anomaly (parameter L)

Other distance-based approaches

Clustering approaches can be used for detecting anomalies:

1. Divide the signal into (possibly) overlapping subsequences
2. Perform clustering on the subsequences (k-Means, spectral clustering etc...)
3. Subsequences that are *far* from their centroids/medoids are likely to be outliers

More details in [Schmidl et al., 2022; Boniol et al., 2022]

A word on supervised approaches

- Semi-supervised approaches: use knowledge on normality
  - Learn a model on normality and detect anomalies in the residual or as a derivation from normality
  - Example : use annotated templates representing normal behavior, retrieve them in the signal up to a measure of fit (see Lecture 1), and detect all segments in the time series that do not correspond to a known pattern as anomalies
- Supervised approaches: supervised classification techniques can also be used: SVM, random forests, neural networks...

Contents

Evaluation of event detection methods

- Several time series ML tasks are event detection tasks: pattern recognition (Lecture 1), change-point detection, anomaly detection...
- In order to benchmark the tested methods, one need to use
  - An annotated dataset where the events of interested have been highlighted
  - Some relevant metrics of evaluation
- What does a *good detection* mean ?

Point-based vs. range-based

- When the events consist of single points, the method can be assessed with the standard precision/recall metric:

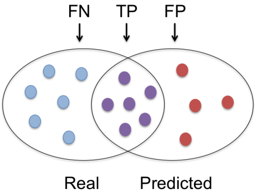
$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

  where TP is the number of true positive, FP the number of false positive and FN the number of false negative

- These metrics are comprised between 0 and 1, and we can plot the precision/recall curve to benchmark the methods

What if the events are range-based?



Range-based detection: example

- Event is correctly detected
- BUT the detection is a bit delayed: would not be suitable for e.g. anomaly detection in industrial monitoring

Range-based detection: example

- Event is correctly detected
- BUT the duration of the event is poorly estimated

Range-based detection: example

- Event is correctly detected
- BUT the two annotated events are detected as one single event

Metrics for event detection

Several principles can be taken into account [Tatbul et al., 2018]:

- *Existence:* Catching the existence of the event (even by predicting only a single point), by itself, might be valuable for the application.
- *Size:* The larger the size of the correctly predicted portion of the event, the higher the recall score.
- *Position:* In some cases, not only size, but also the relative position of the correctly predicted portion of the event might matter to the application.
- *Cardinality:* Detecting the event with a single prediction range may be more valuable than doing so with multiple different ranges in a fragmented manner.

Formulation

- We consider a set of predicted intervals $P = \{P_1, \ldots, P_{N_P}\}$ and a set of real intervals $R = \{R_1, \ldots, R_{N_R}\}$, the recall can be computed as:

$$\text{recall} = \frac{1}{N_R} \sum_{i=1}^{N_R} \text{recall}(R_i, P)$$

- The term $\text{recall}(R_i, P)$ will be defined as a weighted sum of several terms that will assess how well event R_i has been detected
- The same definition can be computed for the precision, but this time as

$$\text{precision} = \frac{1}{N_P} \sum_{i=1}^{N_P} \text{precision}(R, P_i)$$

Formulation

- Existence (only used for recall):

$$\text{existence}(R_i, P) = \begin{cases} 1 & \text{if } \sum_{j=1}^{N_p} |R_i \cap P_j| \geq 1 \\ 0 & \text{elsewhere} \end{cases}$$

- Size/position (used for precision and recall):

$$\text{size\_position}(R_i, P) = \sum_{j=1}^{N_p} w(R_i, R_i \cap P_j)$$

where $w(A, B)$ is an overlap score (between 0 and 1) between $R_i \cap P_j$ and the *desirable* portion of R_i (can voluntary introduce a bias if e.g. we wish to detect the event R_i in advance, or the middle part of R_i, etc...)

- Cardinality (used for position and recall):

$$\text{cardinality}(R_i, P) = \begin{cases} 1 & \text{if } R_i \text{ overlaps with at most one } P_j \\ \gamma(R_i, P) & \text{elsewhere} \end{cases}$$

where γ is a penalty function

Formulation

- Recall:

$$\text{recall}(R_i, P) = \alpha\,\text{existence}(R_i, P) + (1-\alpha)\,\text{cardinality}(R_i, P) \times \text{size\_position}(R_i, P)$$

- Precision:

$$\text{precision}(R, P_i) = \text{cardinality}(R, P_i) \times \text{size\_position}(R, P_i)$$

- Note that different functions w and γ can be used for precision and recall, and several choices can be used (see [Tatbul et al., 2018] and associated mini-project for details)
- If all events in R and P are single-point, $\alpha = 0$, $\gamma(.,.) = 1$ and $w(.,.)$ is the percentage of common points, then this definition is compliant with the regular precision/recall definition

How to choose the parameters

- Several possible parametrizations are provided in the original article: depends on the usecases
- One simpler solution is to compute the Intersection Over Union (IoU) metric between the detected segment and the true segment

$$\mathrm{IoU} = \frac{|P_i \cap R_j|}{|P_i \cup R_j|}$$

  and use a threshold value (e.g. 25%, 50%, 75%...) as a detection criteria
- Performances can be provided with different threshold values so as to give a better idea of the accuracy of the detection method

References

- Truong, C., Oudre, L., & Vayatis, N. (2020). Selective review of offline change point detection methods. Signal Processing, 167, 107299.
- Basseville, M., & Nikiforov, I. V. (1993). Detection of abrupt changes: theory and application (Vol. 104). Englewood Cliffs: prentice Hall.
- Page, E. S. (1955). A test for a change in a parameter occurring at an unknown point. Biometrika, 42(3/4), 523-527.
- Lavielle, M. (1999). Detection of multiple changes in a sequence of dependent variables. Stochastic Processes and their applications, 83(1), 79-102.
- Bai, J., & Perron, P. (1998). Estimating and testing linear models with multiple structural changes. Econometrica, 47-78.
- Lung-Yut-Fong, A., Lévy-Leduc, C., & Cappé, O. (2015). Homogeneity and change-point detection tests for multivariate data using rank statistics. Journal de la Société Française de Statistique, 156(4), 133-162.
- Bai, J., & Perron, P. (2003). Computation and analysis of multiple structural change models. Journal of applied econometrics, 18(1), 1-22.
- Killick, R., Fearnhead, P., & Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association, 107(500), 1590-1598.
- Schwarz, G. (1978). Estimating the dimension of a model. The annals of statistics, 6(2), 461-464.
- Akaike, H. (1974). A new look at the statistical model identification. IEEE transactions on automatic control, 19(6), 716-723.
- Truong, C., Oudre, L., & Vayatis, N. (2017). Penalty learning for changepoint detection. In 2017 25th European Signal Processing Conference (EUSIPCO) (pp. 1569-1573). IEEE.
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 1-58.
- Roberts, S. W. (2000). Control chart tests based on geometric moving averages. Technometrics, 42(1), 97-101.
- Leys, C., Ley, C., Klein, O., Bernard, P., & Licata, L. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. Journal of Experimental Social Psychology, 49(4), 764-766.
- Yamanishi, K., & Takeuchi, J. I. (2002, July). A unifying framework for detecting outliers and change points from non-stationary time series data. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 676-681).
- Hill, D. J., & Minsker, B. S. (2010). Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. Environmental Modelling & Software, 25(9), 1014-1022.
- Yeh, C. C. M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H. A., ... & Keogh, E. (2016, December). Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In 2016 IEEE 16th international conference on data mining (ICDM) (pp. 1317-1322). Ieee.
- Schmidl, S., Wenig, P., & Papenbrock, T. (2022). Anomaly detection in time series: a comprehensive evaluation. Proceedings of the VLDB Endowment, 15(9), 1779-1797.
- Boniol, P., Paparrizos, J., Kang, Y., Palpanas, T., Tsay, R. S., Elmore, A. J., & Franklin, M. J. (2022). Theseus: navigating the labyrinth of time-series anomaly detection. Proceedings of the VLDB Endowment, 15(12), 3702-3705.
- Tatbul, N., Lee, T. J., Zdonik, S., Alam, M., & Gottschlich, J. (2018). Precision and recall for time series. arXiv preprint arXiv:1803.03639.

List of possible topics/projects

- How to contribute to the `ruptures` package (see with C. Truong) https://centre-borelli.github.io/ruptures-docs/
- Truong, C., Oudre, L., & Vayatis, N. (2017). Penalty learning for changepoint detection. In 2017 25th European Signal Processing Conference (EUSIPCO) (pp. 1569-1573). IEEE.
  How to learn the penalty for change point detection
- Fearnhead, P., & Rigaill, G. (2019). Changepoint detection in the presence of outliers. Journal of the American Statistical Association, 114(525), 169-183.
  How to detect change-points in presence of outliers
- Kim, H., Kim, B., Chung, D., Yoon, J., & Ko, S.-K. (2022). SoccerCPD: formation and role change-point detection in soccer matches using spatiotemporal tracking data. Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (SIGKDD), 3146–3156.
  How to use change point detection to study soccer games
- Fearnhead, P., Maidstone, R., & Letchford, A. (2019). Detecting Changes in Slope With an L0 Penalty. Journal of Computational and Graphical Statistics, 28(2), 265–275.
  How to introduce a sparsity penalty into change-point detection problems
- Runge, V., Hocking, T. D., Romano, G., Afghah, F., Fearnhead, P., & Rigaill, G. (2020). gfpop: an R Package for Univariate Graph-Constrained Change-Point Detection. ArXiv E-Prints ArXiv:2002.03646.
  How to introduce graph constraints into change-point detection problems
- Jewell, S. W., Hocking, T. D., Fearnhead, P., & Witten, D. M. (2020). Fast nonconvex deconvolution of calcium imaging data. Biostatistics, 21(4), 709–726.
  How to apply change-point detection to biology

List of possible topics/projects

- Chin, S. C., Ray, A., & Rajagopalan, V. (2005). Symbolic time series analysis for anomaly detection: A comparative evaluation. Signal Processing, 85(9), 1859-1868.
  How to use symbolic representation for detecting anomalies
- Chandola, V., Banerjee, A., & Kumar, V. (2010). Anomaly detection for discrete sequences: A survey. IEEE transactions on knowledge and data engineering, 24(5), 823-839.
  How to detect anomalies in discrete time series
- Boniol, P., Linardi, M., Roncallo, F., & Palpanas, T. (2020, April). Automated Anomaly Detection in Large Sequences. In 2020 IEEE 36th International Conference on Data Engineering (ICDE) (pp. 1834-1837). IEEE.
  How to use detect anomalies in large time series
- Nakamura, T., Imamura, M., Mercer, R., & Keogh, E. MERLIN: Parameter-Free Discovery of Arbitrary Length Anomalies in Massive Time Series Archives. In Proc. 20th IEEE Intl. Conf. Data Mining.
  How to detect anomalies with different lengths
- Tatbul, N., Lee, T. J., Zdonik, S., Alam, M., & Gottschlich, J. (2018). Precision and recall for time series. arXiv preprint arXiv:1803.03639.
  How to assess event detection techniques
- Schmidl, S., Wenig, P., & Papenbrock, T. (2022). Anomaly detection in time series: a comprehensive evaluation. Proceedings of the VLDB Endowment, 15(9), 1779-1797.
  Wonderful article with tons of references, implementations, etc...
- Boniol, P., Paparrizos, J., Kang, Y., Palpanas, T., Tsay, R. S., Elmore, A. J., & Franklin, M. J. (2022). Theseus: navigating the labyrinth of time-series anomaly detection. Proceedings of the VLDB Endowment, 15(12), 3702-3705.
- Wenig, P., Schmidl, S., & Papenbrock, T. (2022). TimeEval: a benchmarking toolkit for time series anomaly detection algorithms. Proceedings of the VLDB Endowment, 15(12), 3678-3681.

PELT pruning strategy

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

- We want to show that given two times s and t such that $t < s < N$, if

$$\min_{\mathcal{T}} \left[V(\mathcal{T}, x[0 : t]) + \beta|\mathcal{T}| \right] + c(x[t : s]) \geq \min_{\mathcal{T}} \left[V(\mathcal{T}, x[0 : s]) + \beta|\mathcal{T}| \right]$$

  then t cannot be the last change point prior to N

- Given two times s and t such that $t < s < N$, assume that

$$\min_{\mathcal{T}} \left[V(\mathcal{T}, x[0 : t]) + \beta|\mathcal{T}| \right] + c(x[t : s]) \geq \min_{\mathcal{T}} \left[V(\mathcal{T}, x[0 : s]) + \beta|\mathcal{T}| \right]$$

- By adding $c(x[s : N]) + \beta$ to this equation, and using the fact that $c(x[t : s]) + c(x[s : N]) \leq c(x[t : N])$ (adding a breakpoint decreases the sum of costs), it goes that

$$\min_{\mathcal{T}} \left[V(\mathcal{T}, x[0 : t]) + \beta|\mathcal{T}| \right] + c(x[t : N]) + \beta \geq \min_{\mathcal{T}} \left[V(\mathcal{T}, x[0 : s]) + \beta|\mathcal{T}| \right] + c(x[s : N]) + \beta$$

PELT pruning strategy

- Now, if $t < s$ is the last change point prior to N, then

$$\min_{\mathcal{T}} \left[V(\mathcal{T}, x[0:s]) + \beta|\mathcal{T}| \right] = \min_{t} \left[\min_{\mathcal{T}} \left[V(\mathcal{T}, x[0:t]) + \beta|\mathcal{T}| \right] + c(x[t:s]) + \beta \right]$$

- But we also have

$$\min_{\mathcal{T}} \left[V(\mathcal{T}, x[0:t]) + \beta|\mathcal{T}| \right] + c(x[t:N]) + \beta \geq \min_{\mathcal{T}} \left[V(\mathcal{T}, x[0:s]) + \beta|\mathcal{T}| \right] + c(x[s:N]) + \beta$$

- This shows that s is a better choice than t in the first equation, and thus t cannot be the last change point prior to N