Intelligence Artificielle & Machine Learning pour la modélisation de séries temporelles et de signaux

Séance 8 : Détection de ruptures et d'anomalies

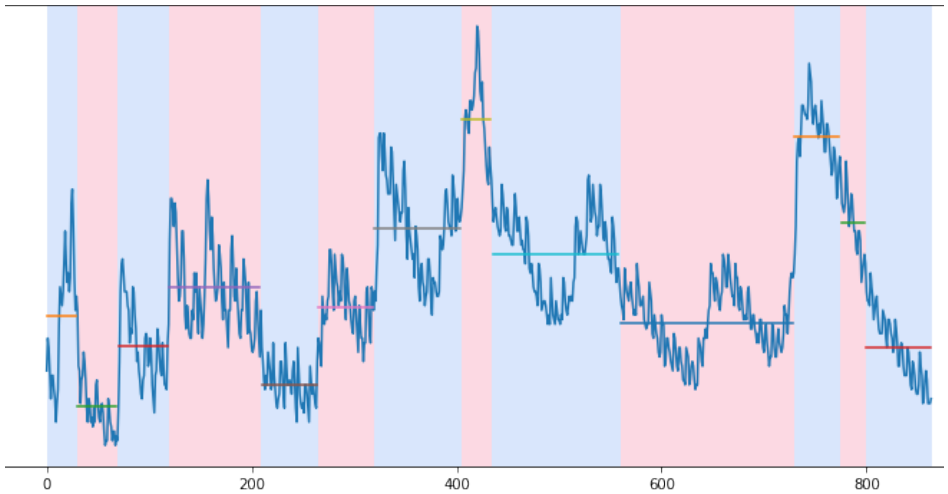
Laurent Oudre
laurent.oudre@ens-paris-saclay.fr

Diplôme ARIA
ENS Paris Saclay
2026-27

Découvrir des événements dans les séries temporelles

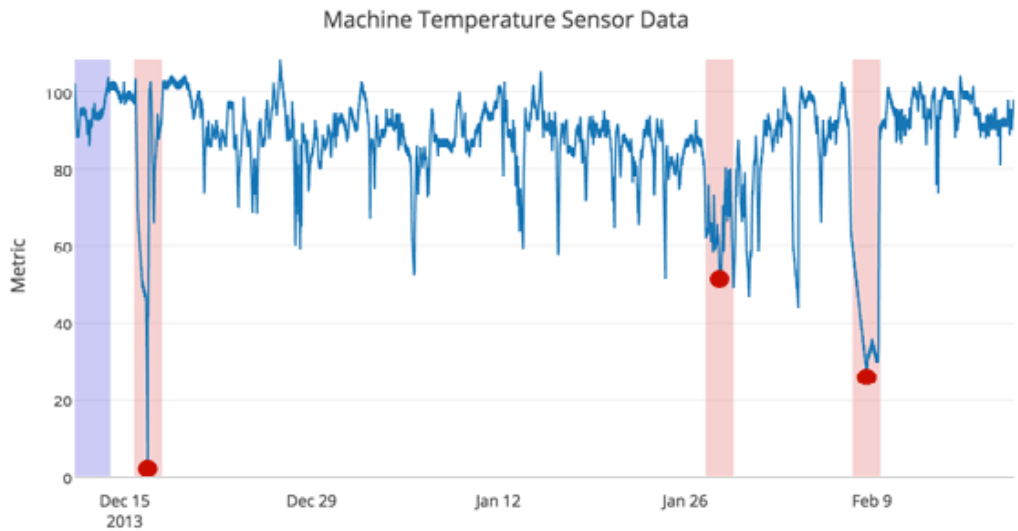
- Les séries temporelles sont très utilisées pour le monitoring : finance, industrie, santé, météorologie...
- Lorsque qu'elles durent des heures, des jours, des semaines... il est probable qu'il y ait de la redondance
- Deux questions fondamentales :
  - Y'a-t-il eu des changements significatifs dans mes données au cours du temps ?
  - Y'a-t-il eu quelque chose de nouveau ou d'inusuel dans mes données ?
- Deux problèmes mal posés : qu'entend-on par *changement significatif ?* par *nouveau* ou *inusuel* ?

Problème 1 : Détection de ruptures



- A partir d'une série temporelle, estimer le nombre de changements et leurs temps d'apparition
- Dépend de façon cruciale de la définition donnée à la notion de changement, mais aussi du niveau de granularité désiré

Problème 2 : Détection d'anomalies



- A partir d'une série temporelle, retrouver l'ensemble des échantillons anormaux
- Anomalie constituée d'échantillons isolés ou contigus
- Dépend également du sens donné à la notion d'usuel/inusuel

Programme de la séance

- Algorithmes offline pour la détection de ruptures (nombre de ruptures connu ou pas)
- Algorithmes pour la détection d'anomalies isolées et contextuelles

Session 8 : Change-point and anomaly detection

Plan du cours

1. Détection de ruptures

1.1 Importance de la stationnarité
1.2 Quelques fonctions de coût
1.3 Algorithmes de résolution
1.4 Calibration

2. Détection d'anomalies

2.1 Détection d'outliers
2.2 Détection d'événements inusuels

Plan du cours

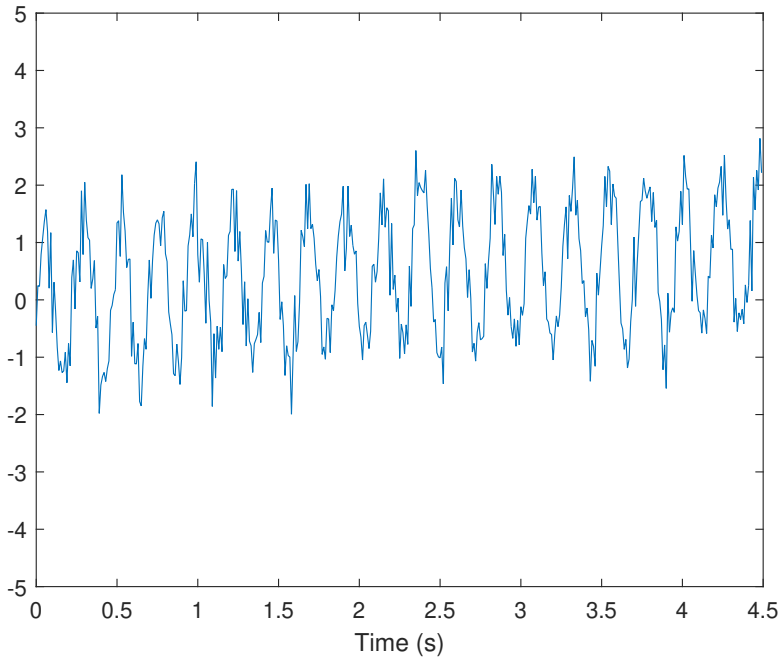
1. Détection de ruptures
1.1 Importance de la stationnarité
1.2 Quelques fonctions de coût
1.3 Algorithmes de résolution
1.4 Calibration

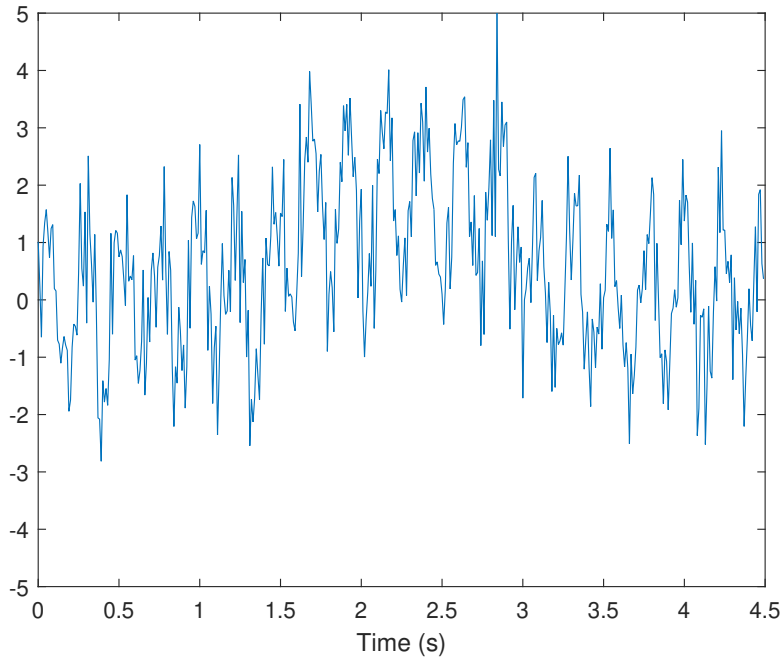
2. Détection d'anomalies

Importance de la stationnarité

- Comme nous avions vu dans les premières séances, la stationnarité (au sens large) est une supposition fondamentale en traitement du signal
- Il s'agit d'une hypothèse nécessaire pour calculer une DSP, une fonction d'autocorrélation etc...
- Néanmoins, lorsqu'on l'observe durant une grande période de temps, le comportement d'un système est susceptible de changer au cours du temps, soit de façon lente soit abrupte

Exemple





Evolution lente vs. changement abrupt

Comment contourner le problème

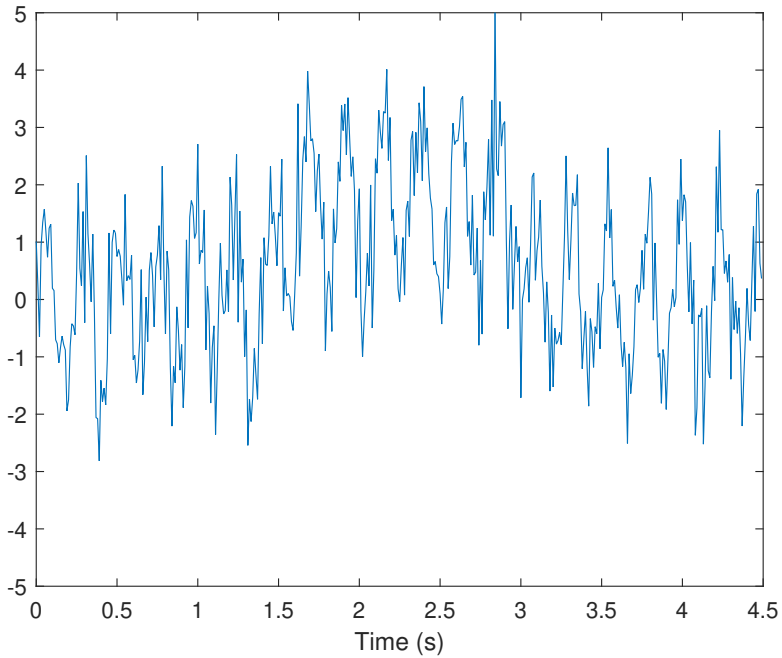
En première approximation, plusieurs stratégies peuvent être utilisées pour contourner le problème :

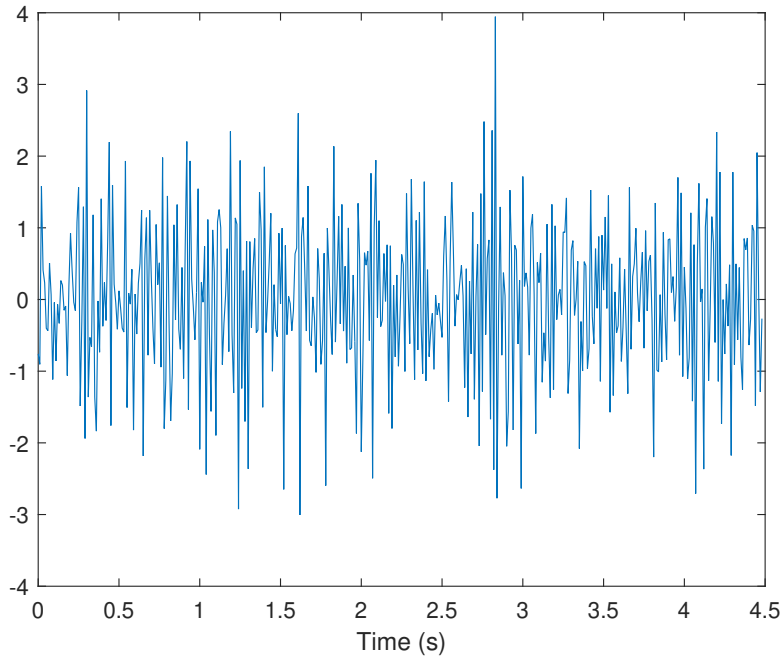
- Si les changements sont lents, cette tâche revient à du detrending (voir séances précedentes)
- On peut diviser le signal en trames sur lesquelles le signal est supposé stationnaire (c'est ce que l'on fait par exemple pour calculer un spectrogramme)
- Au lieu de travailler sur le signal de base $x[n]$, on peut travailler sur sa différence première (dérivée discrète)

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

qui a souvent de meilleures propriétés de stationnarité
Attention : s'il faut ré-intégrer ensuite le signal, ceci peut être source d'erreurs !

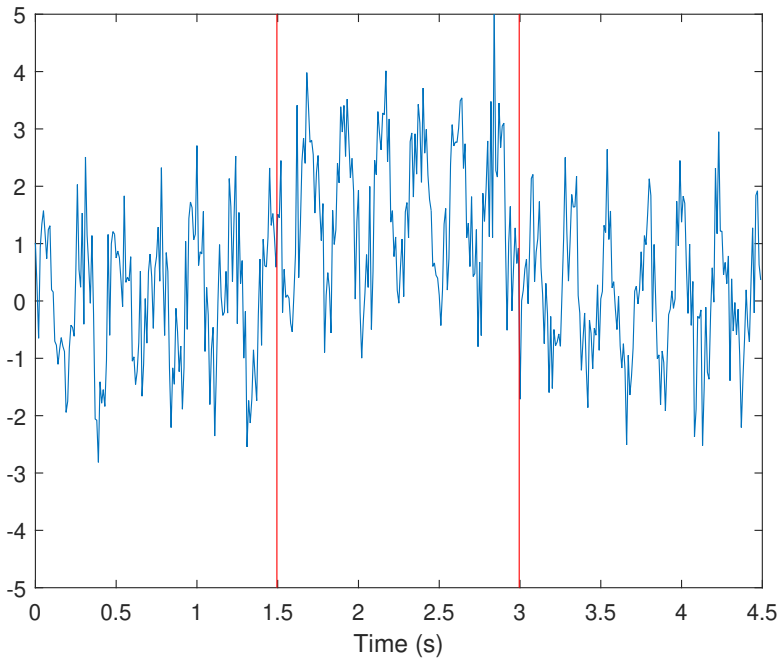
Exemple





Utilisation de la dérivée pour rendre le signal *plus stationnaire*
Gauche : signal de base / Droite : dérivée

Présentation du problème



- Dans le cas où les ruptures sont abruptes, on a recours à un algorithme de segmentation (ou détection de ruptures)
- On suppose alors que le signal $x[n]$ subit des changements aux temps

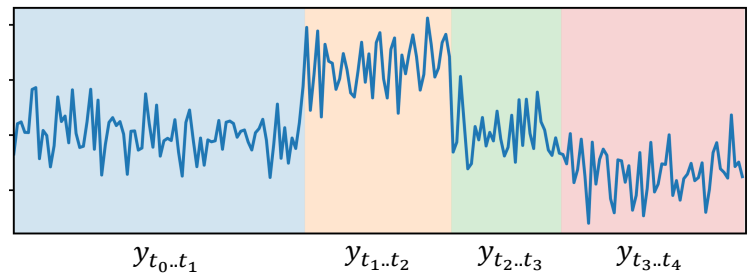
$$\mathcal{T}^* = (t_1^*, \ldots, t_{K^*}^*)$$

- But du problème : retrouver le nombre de ruptures K^* ainsi que les temps de rupture $\mathcal{T}^*$
- Deux hypothèses (dans un premier temps) : $K = K^*$ connu, et détection hors ligne

Résolution du problème

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

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

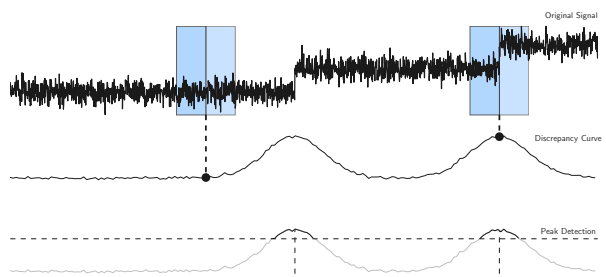


Fonction de coût $c(.)$

- Mesure l'homogénéité des sous-signaux
- Choix de $c(.)$ conditionne le type de ruptures que l'on veut détecter
- Souvent basée sur un modèle probabiliste génératif des données

Résolution du problème de minimisation

- Résolution optimale par programmation dynamique
- Résolution approchée (fenêtres glissantes...)



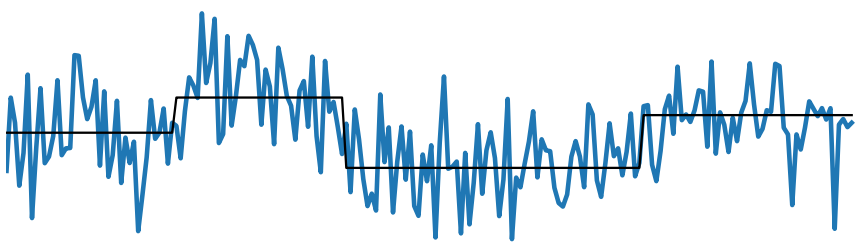
Changement de moyenne

La plus populaire : norme L2

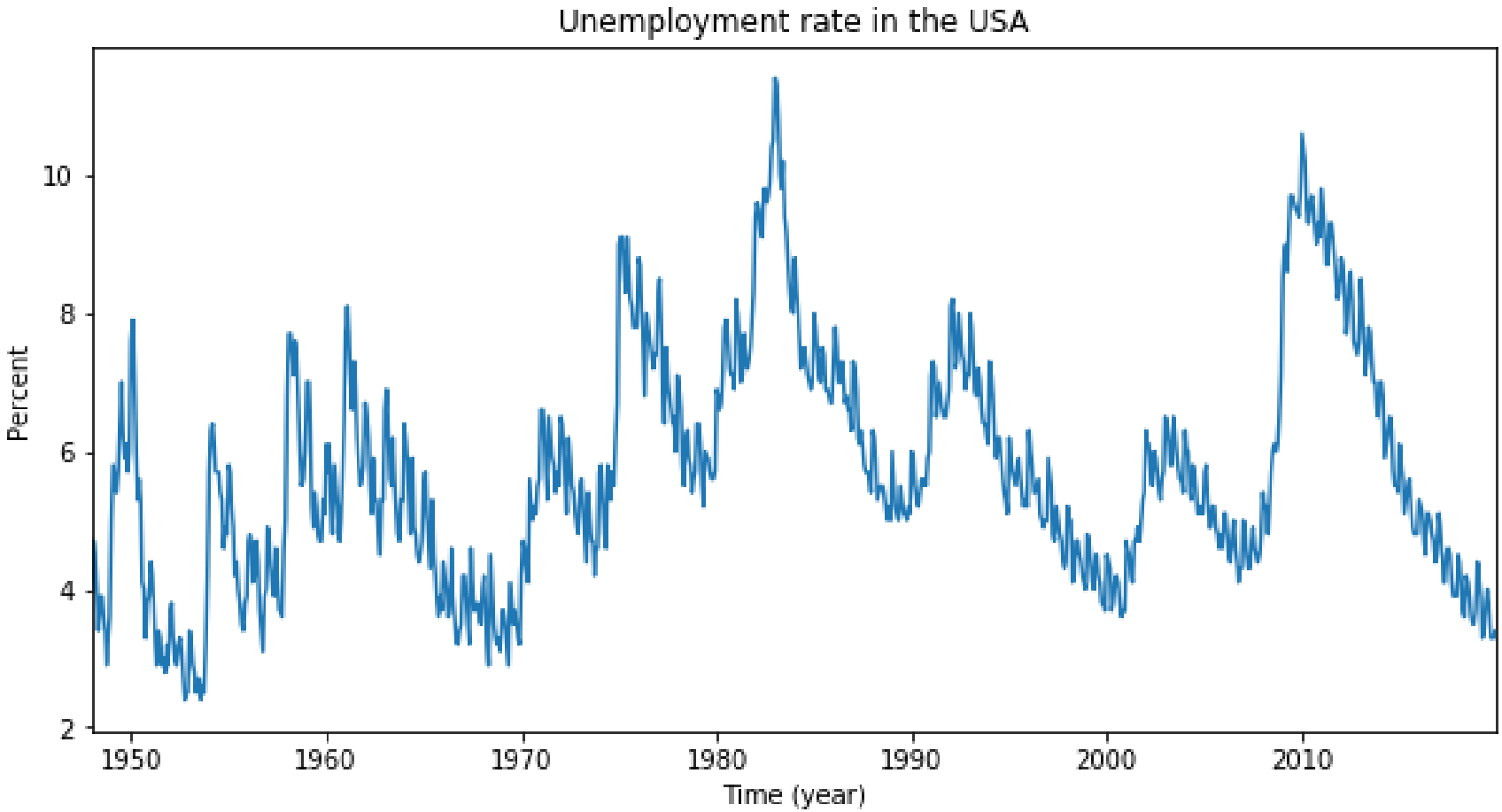
$$c_{L_2}(x[a:b]) = \sum_{n=a+1}^{b} \left\| x[n] - \overline{x[a:b]} \right\|_2^2$$

où $\overline{x[a:b]}$ est la moyenne empirique du sous-signal $x[a:b]$

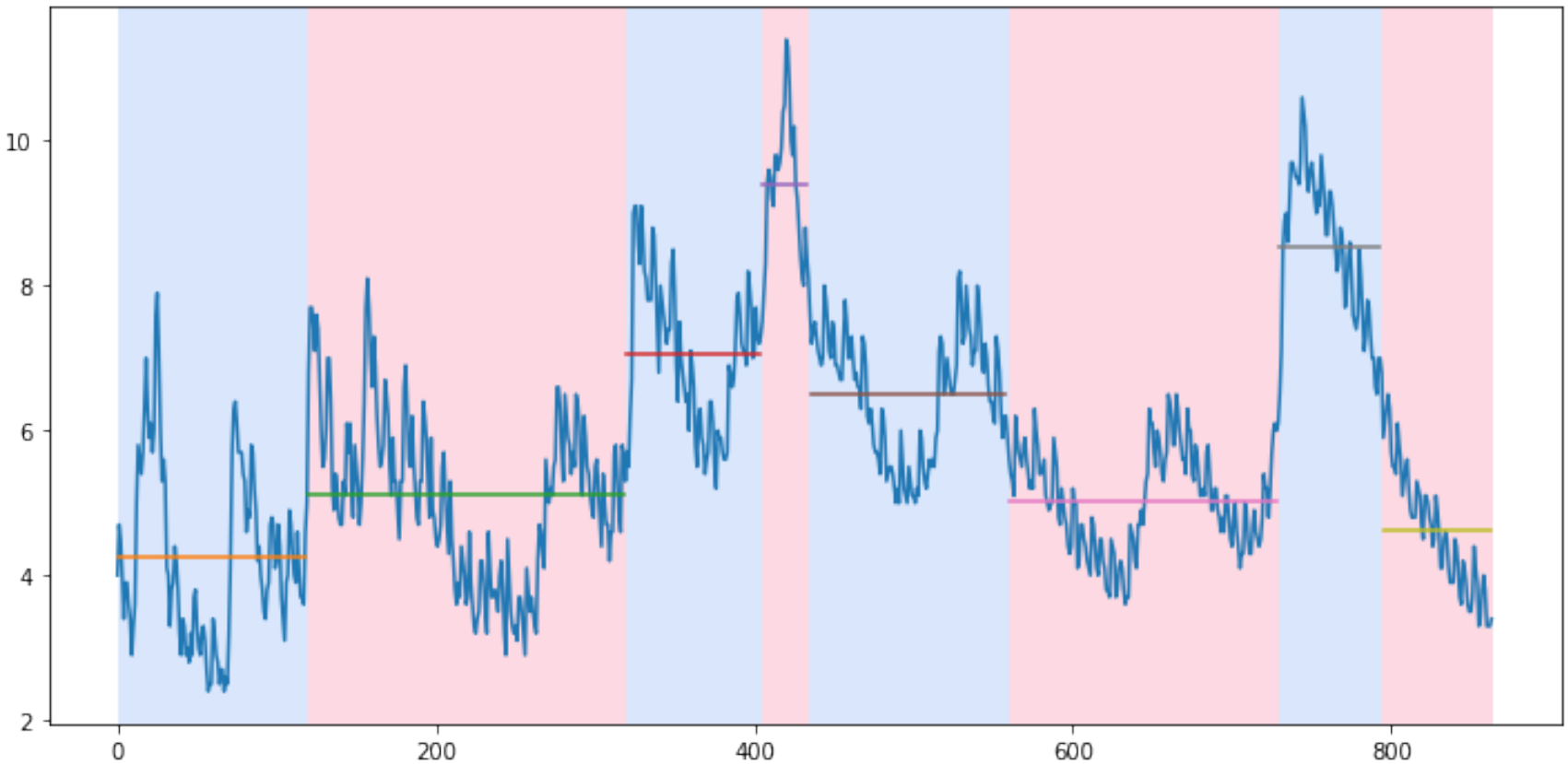
- Fonction de coût paramétrique liée à un modèle probabiliste
- Permet de détecter des changements de moyenne dans des signaux Gaussiens à variance fixe



Exemple

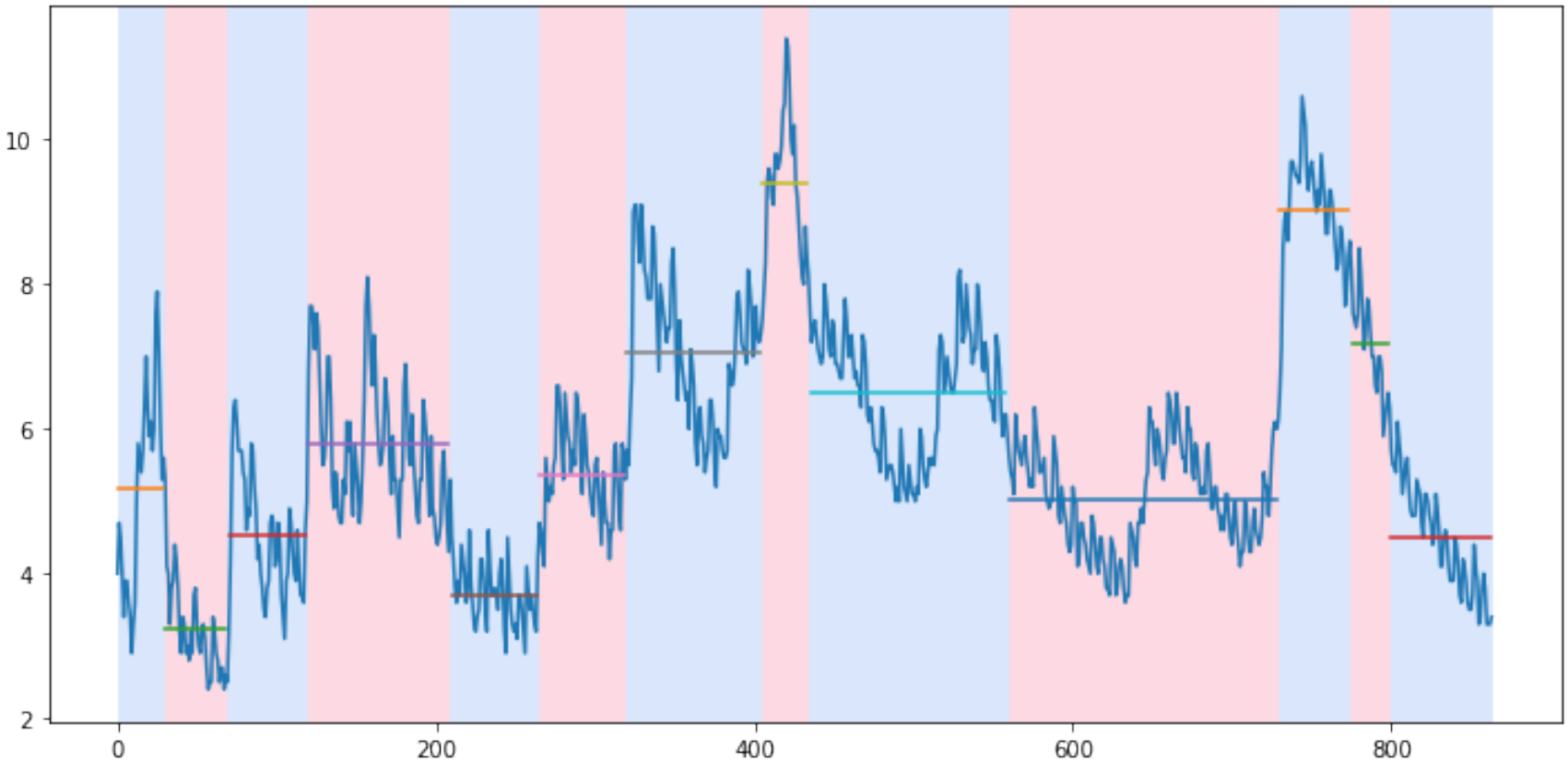

Unemployment rate in the USA
Percent
10
8
6
4
2
1950
1960
1970
1980
1990
2000
2010
Time (year)

Exemple : Détection avec c_{L_2}



$K = 7$

Exemple : Détection avec c_{L_2}



$K = 12$

Changement de moyenne et variance

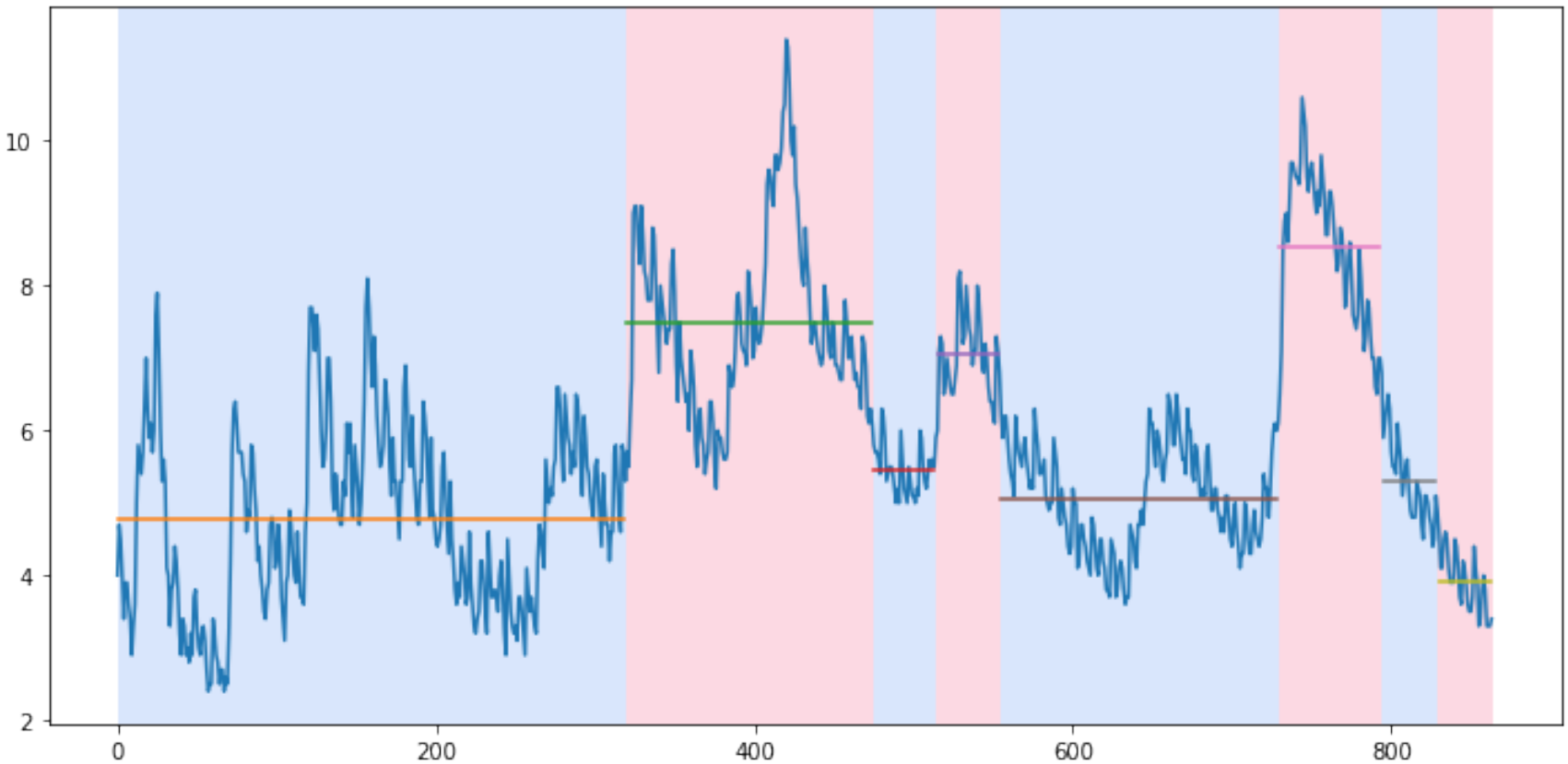
Lorsque la moyenne et la variance changent avec le temps, la fonction de coût devient

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

où $\mu_{a:b}$ et $\sigma^2_{a:b}$ sont les moyennes/variances empiriques du segment $x[a:b]$.

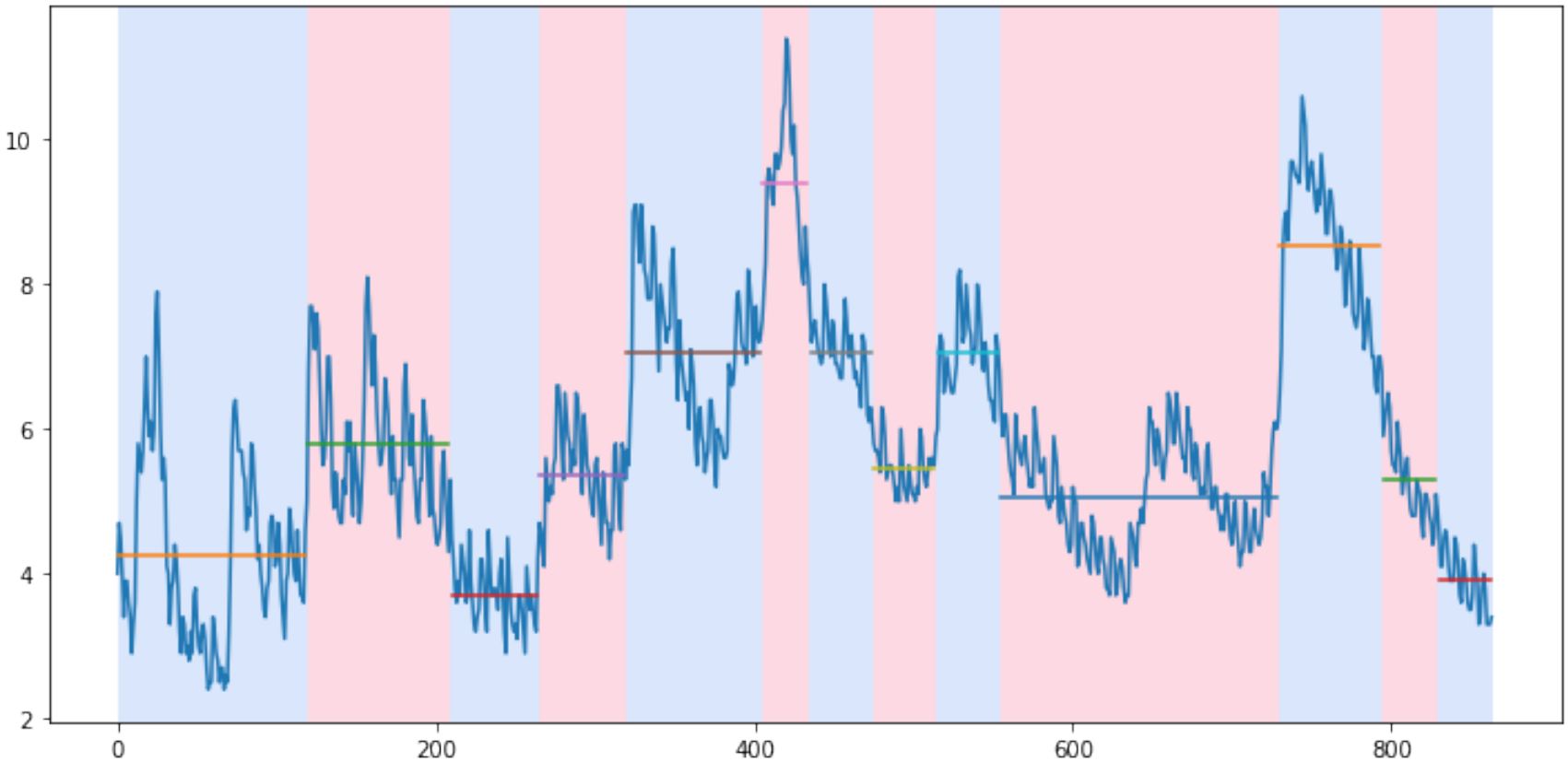
- Revient à un changement de moyenne/variance dans des signaux gaussiens
- On peut adapter cette fonction de coût pour les signaux multivariés en remplaçant la variance par une matrice de covariance : dans ce cas on peut détecter par exemple des changements de corrélations entre les dimensions. [Lavielle, 1999]

Exemple : Détection avec c_Σ



$K = 7$

Exemple : Détection avec c_Σ



$K = 12$

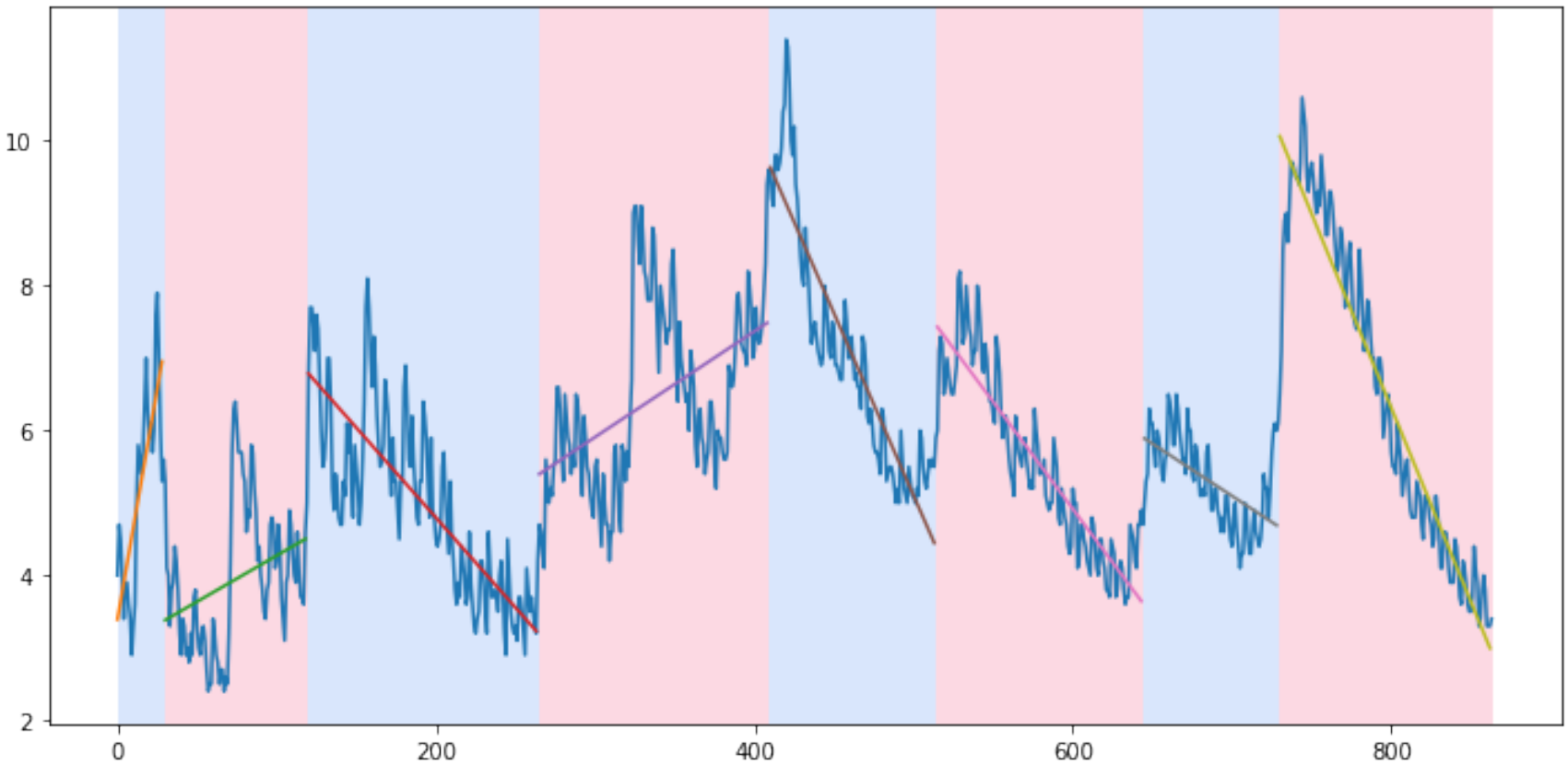
Changement de pente/ordonnée à l'origine

Des changements de pente et d'ordonnée à l'origine peuvent être détectés dans le contexte général de la régression linéaire par morceaux

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

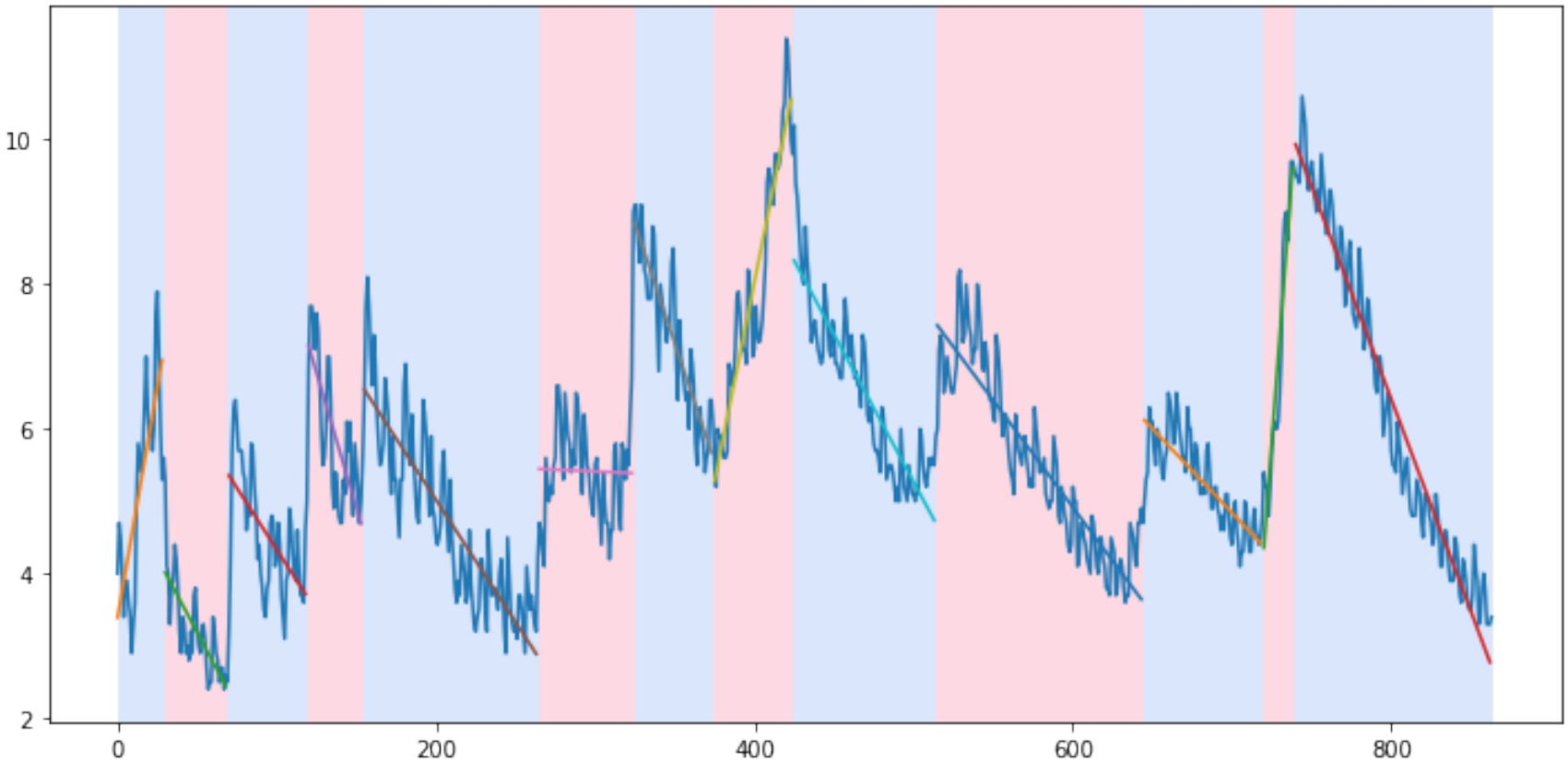
- On fixe des fonctions $\beta_1[n], \ldots, \beta_M[n]$ et on cherche des changements dans les paramètres de régression
- On peut détecter des changements de tendance, de saisonnalité, de paramètres AR, etc... [Bai et al., 1998]
- Pour la pente et l'ordonnée à l'origine, on peut choisir les fonctions $\beta_1[n] = 1$ et $\beta_2[n] = n$

Exemple : Détection avec c_{linear}



$K = 7$

Exemple : Détection avec c_{linear}



$K = 12$

Algorithmes de résolution

Nous allons considérer dans le cours deux méthodes de résolution correspondant à deux cas :

- Si on suppose K connu : problème standard

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

- Si on suppose K inconnu : problème pénalisé

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

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

Résolution optimale pour K connu

- En notant

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

on peut voir que

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

- Il s'agit d'un problème récursif que l'on peut résoudre avec de la programmation dynamique [Bai et al., 2003]
- Deux étapes : calcul des coûts cumulatifs et détermination des ruptures

Résolution optimale pour K connu

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**
  $s \leftarrow L(k)$
  $t^* \leftarrow \operatorname{argmin}_{k-1 \leq t < s} C_{k-1}(1, t) + C_1(t+1, s)$
  $L(k-1) \leftarrow t^*$
  $k \leftarrow k - 1$
end while
Remove T from L
Output: set L of estimated breakpoint indexes.

Complexité de $\mathcal{O}(KN^2)$

Détection de ruptures pénalisée

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

- Estimation jointe des ruptures et du nombre de ruptures
- Le paramètre β pénalise l'introduction d'une nouvelle rupture dans le modèle : une rupture additionnelle doit faire décroître la somme des coûts d'au moins β
- Ce problème est en réalité encore plus simple à résoudre que le problème original avec K fixé !

Stratégie de pruning

- Etant donnés deux temps s et t tels que $t < s < N$, on peut remarquer que si

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

  alors t ne peut pas être la dernière rupture avant N.
- Accélération considérable car ceci va permettre d'éliminer beaucoup de candidats
- On appelle cet algorithme **Pruned Exact Linear Time (PELT)** : sous de bonnes hypothèses il a une complexité de $\mathcal{O}(N)$ [Killick et al., 2012]

Algorithme PELT

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 $t = 1, \dots, T$ **do**
  $\hat{t} \leftarrow \text{argmin}_{s \in \chi} \left[Z[s] + c(y_{s..t}) + \beta \right]$.
  $Z[t] \leftarrow \left[Z[\hat{t}] + c(y_{\hat{t}..t}) + \beta \right]$
  $L[t] \leftarrow L[\hat{t}] \cup \{\hat{t}\}$.
  $\chi \leftarrow \{s \in \chi \ : \ Z[s] + c(y_{s..t}) \leq Z[t]\} \cup \{t\}$
end for
Output: set $L[T]$ of estimated breakpoint indexes.

Calibration

- D'autres fonctions de coût peuvent être utilisées, si l'on veut repérer des changements de variance, de fréquence fondamentale, etc...
- Il existe des méthodes supervisées permettant d'apprendre à partir d'exemples annotés une bonne fonctions de coût
- Lorsque le paramètre K n'est pas connu, on peut également essayer plusieurs valeurs de K entre 1 et K_{max} et observer les résultats ou l'erreur de modélisation
- Il existe des méthodes supervisées ou des heuristiques pour avoir un ordre de grandeur du paramètre β dans PELT

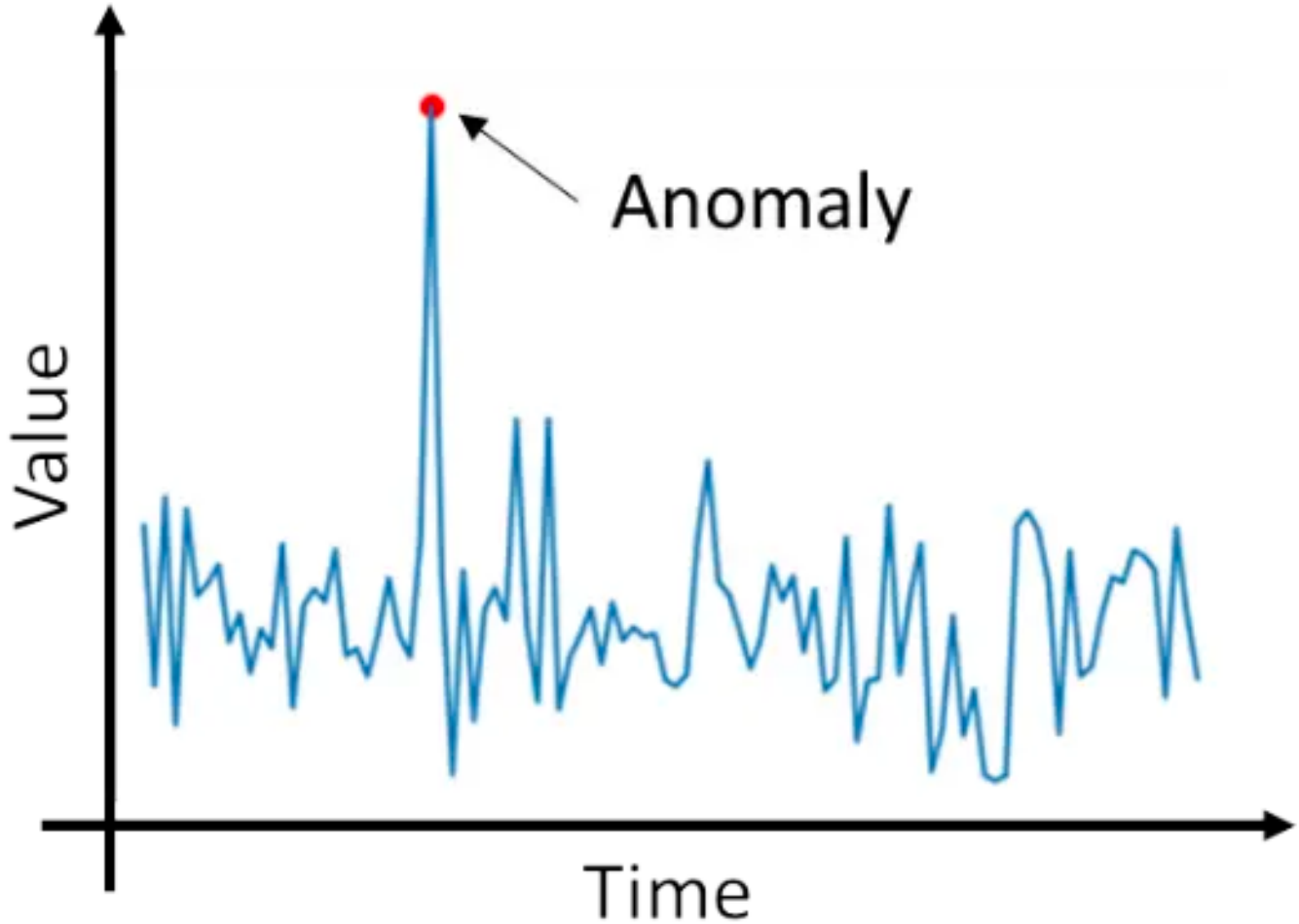
$$\beta = 4\sigma^2 \log N \quad \text{(critère BIC pour la fonction de coût } c_{L_2})$$

Plan du cours

1. Détection de ruptures

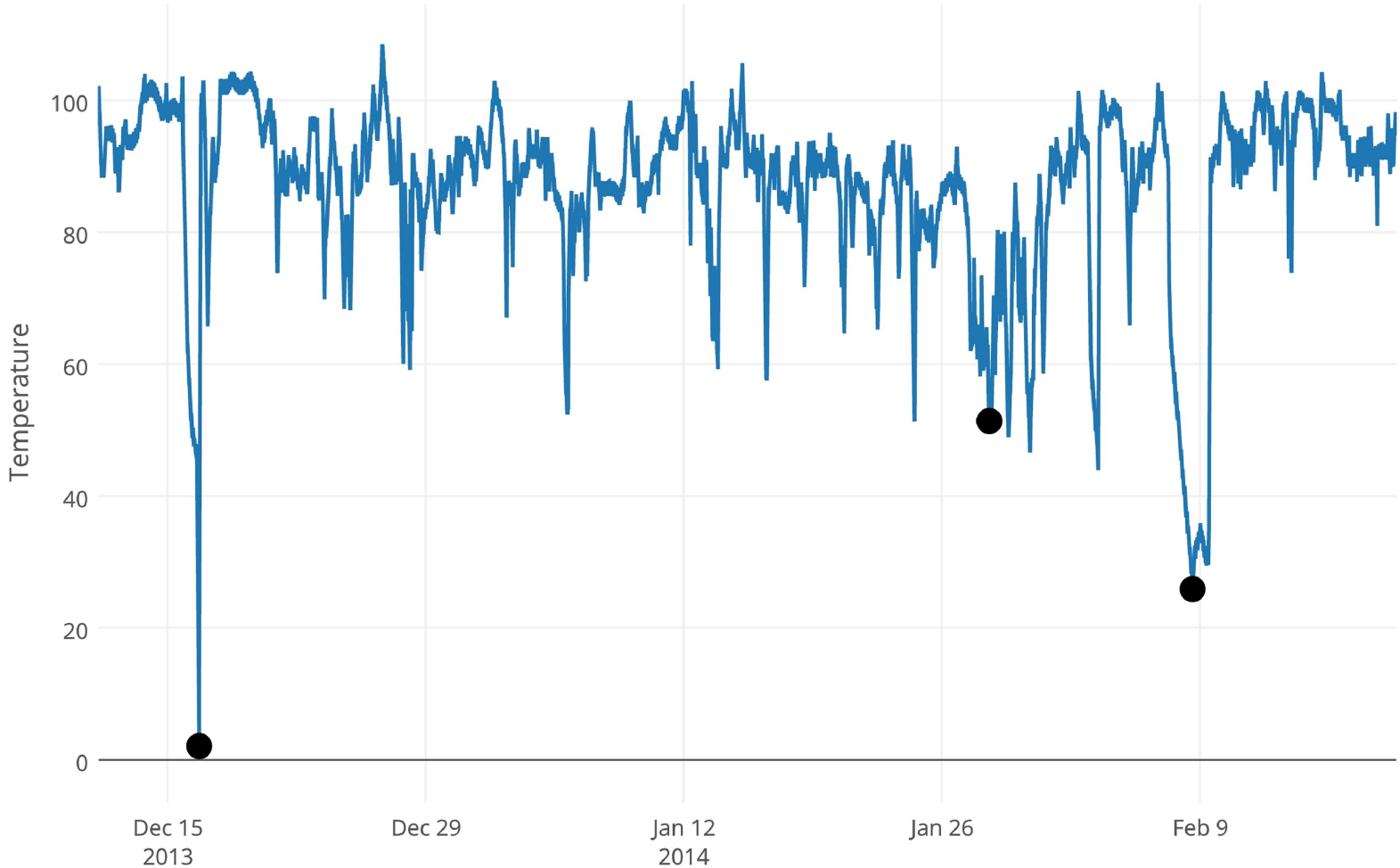
2. Détection d'anomalies
2.1 Détection d'outliers
2.2 Détection d'événements inusuels

Quelques exemples



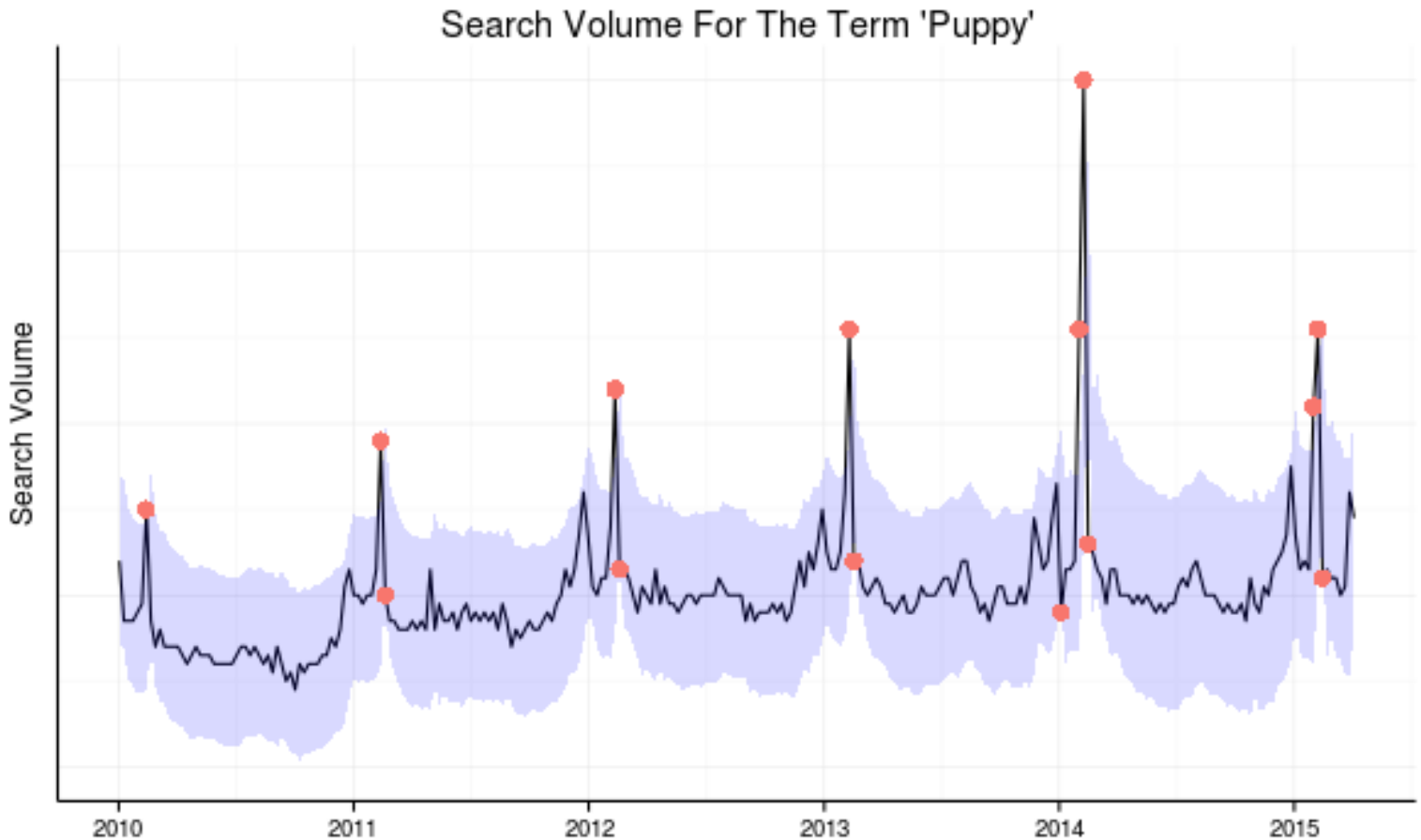
Facile : une anomalie est une valeur trop grande/petite (outlier)

Quelques exemples



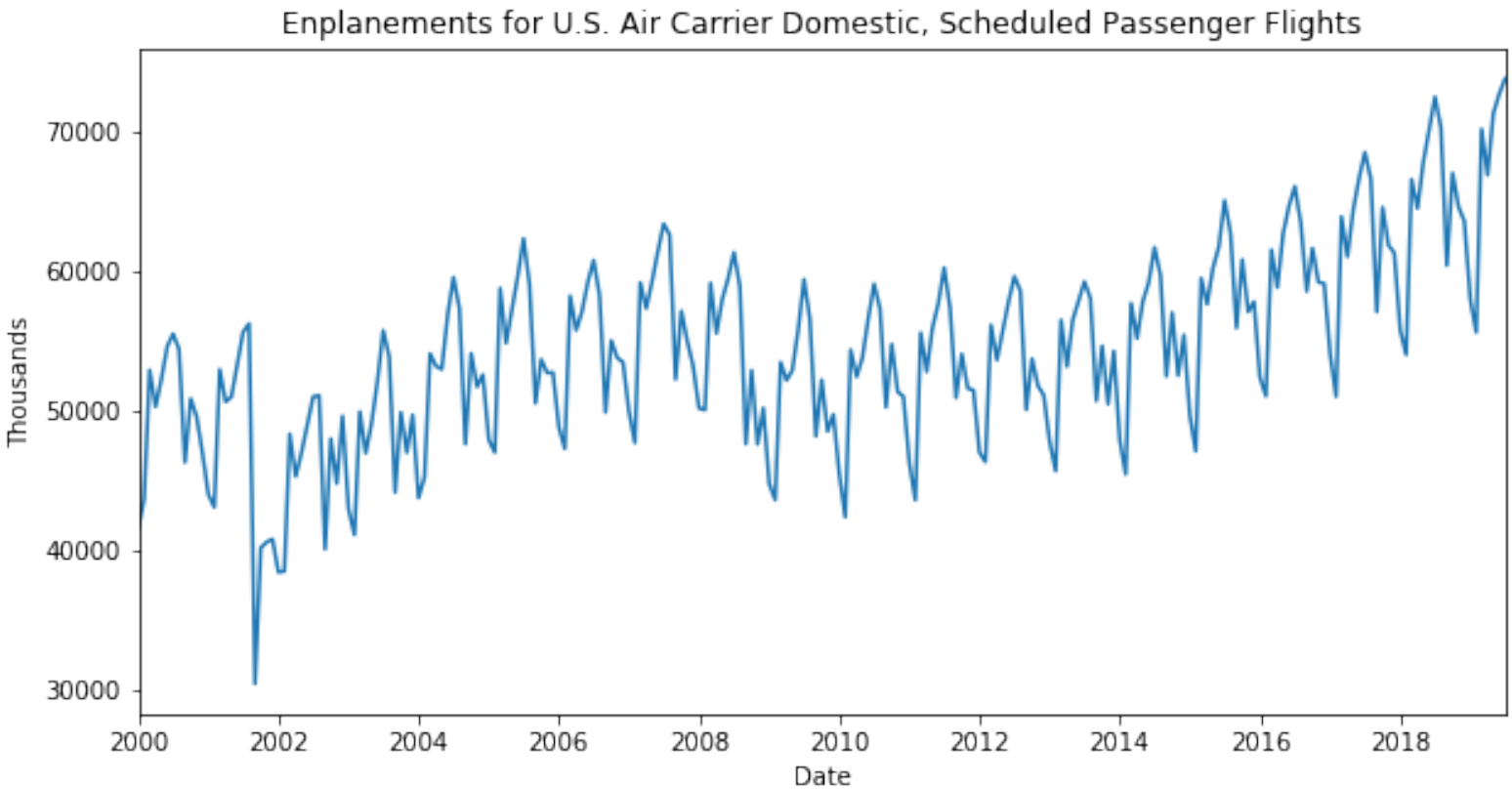
Plus compliqué : *certaines* valeurs trop grandes/petites sont des anomalies

Quelques exemples



Les anomalies dépendent des valeurs précédentes

Quelques exemples



Les anomalies correspondent à des *événements* inusuels

Détection d'anomalies

Comme nous l'avons vu, la notion d'anomalie peut prendre différents sens [Chandola et al., 2009] :

- Outliers, i.e. échantillons isolés avec des valeurs particulièrement grandes/petites
- Groupes d'outliers, i.e. segments qui ne coïncident pas avec ce qui est observé habituellement dans la série temporelle (en terme d'amplitude)
- Événements inusuels qui cassent la régularité de la série temporelle

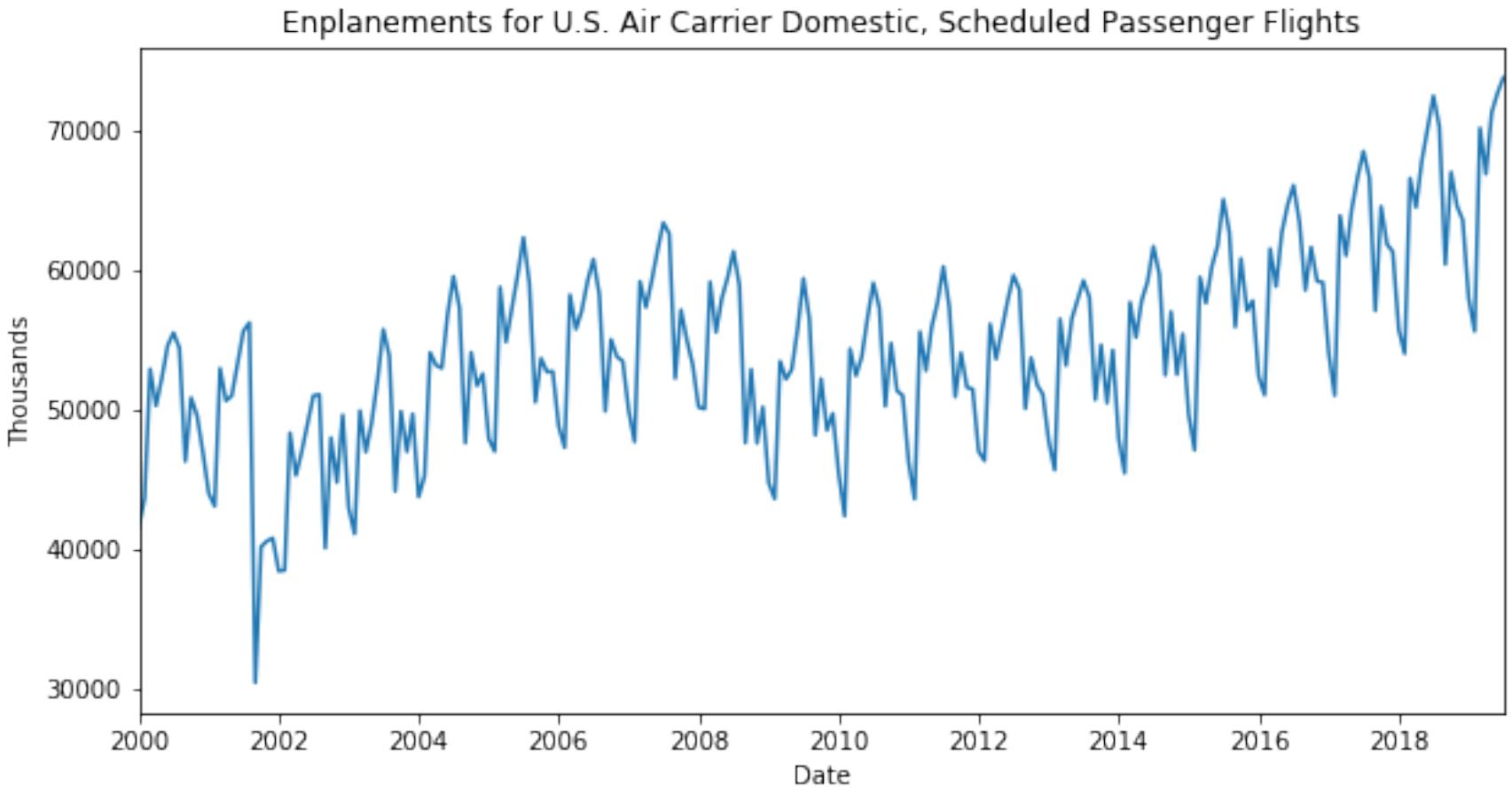
Détection d'outliers

Nous avons déjà vu dans les séances précédentes des méthodes pour détecter/supprimer du bruit impulsionnel et qui peuvent être utilisées dans ce contexte. On peut citer également

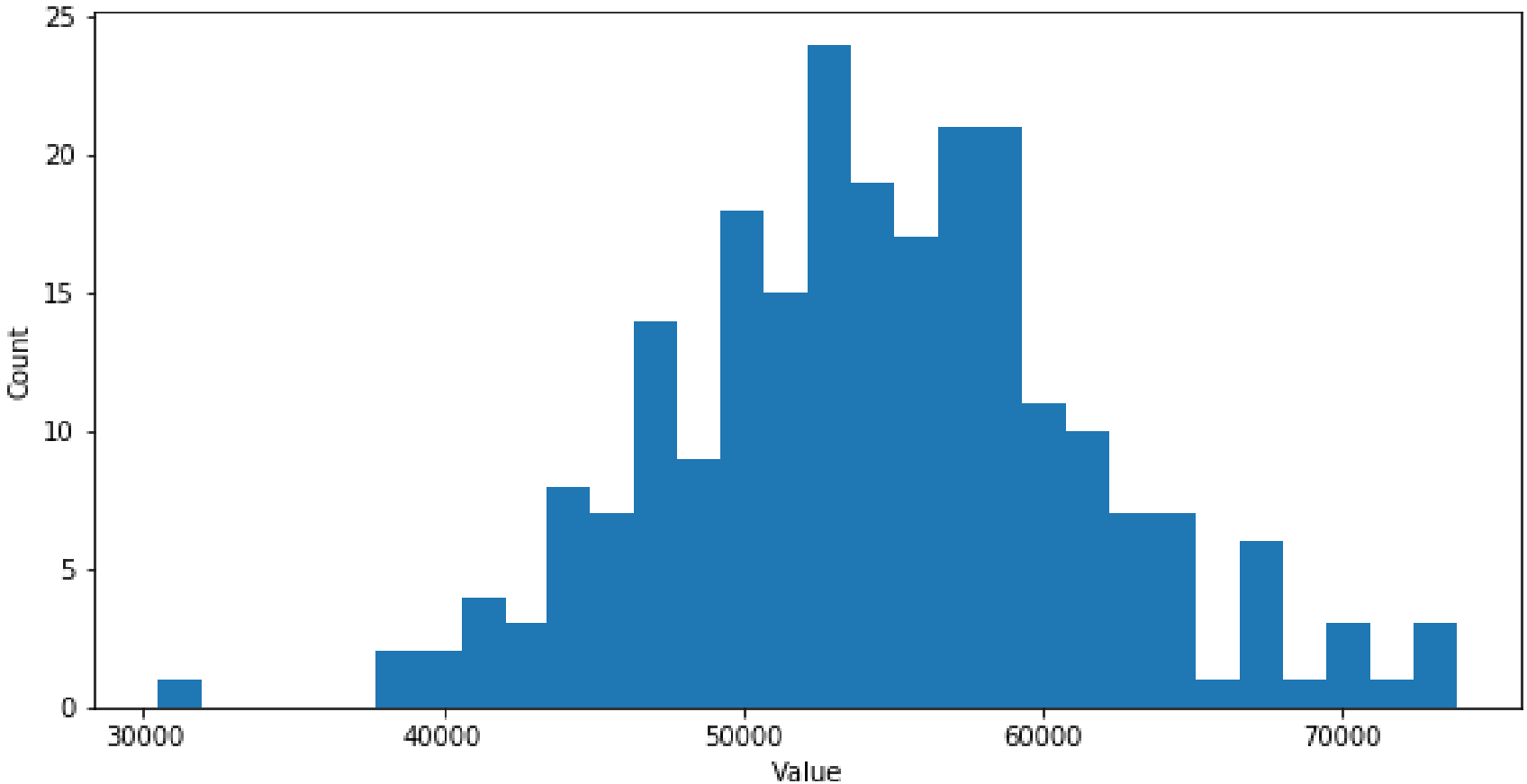
- Méthodes statistiques standard : visualisation d'histogrammes pour détecter les valeurs aberrantes
- Méthodes basées sur des modèles paramétriques : tendance + saisonnalité, modèle AR....

On se contentera donc dans cette partie de montrer des exemples d'application

Exemple : méthodes statistiques

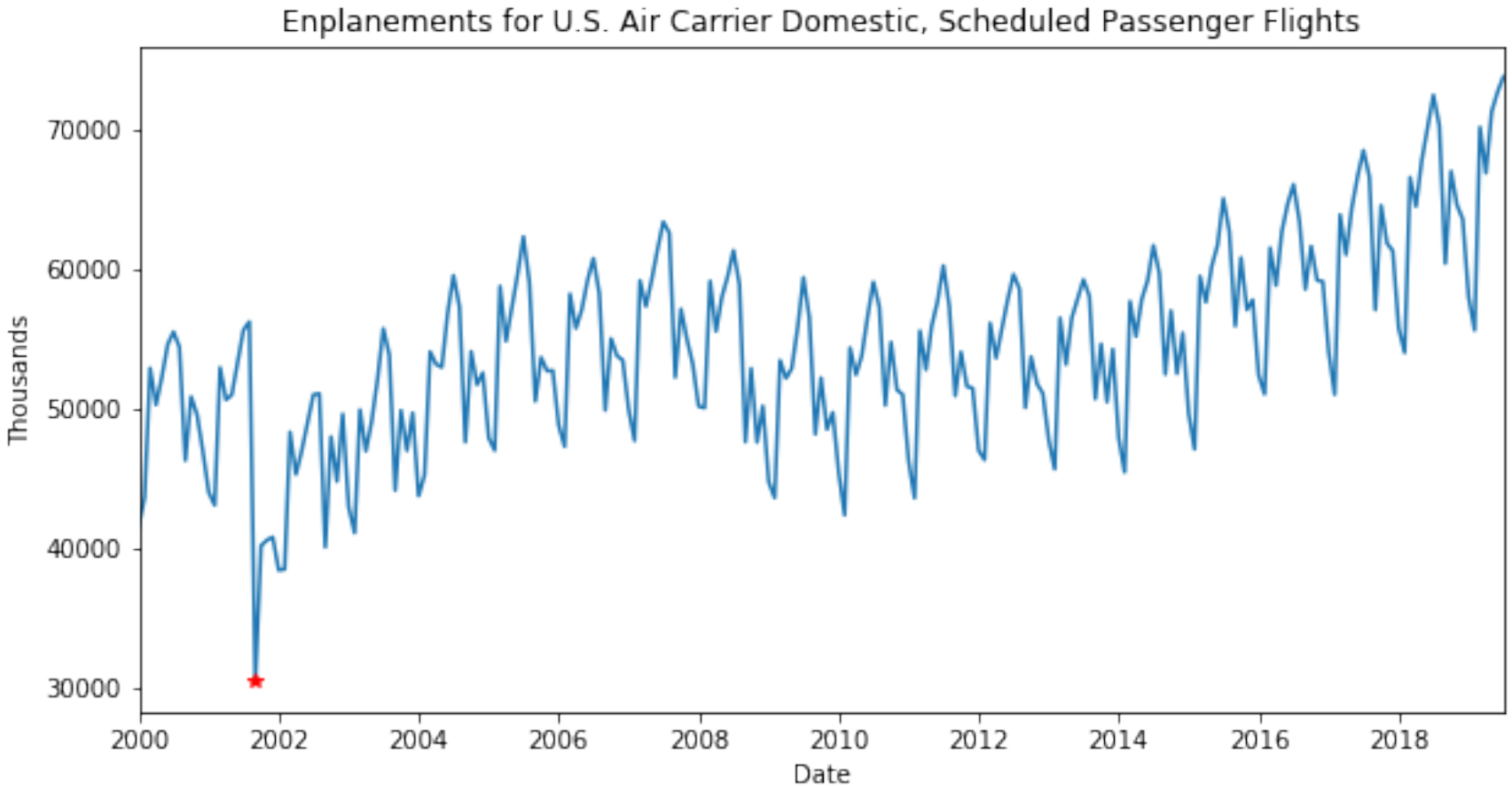


Exemple : méthodes statistiques



Un outlier peut être considéré comme une anomalie

Exemple : méthodes statistiques

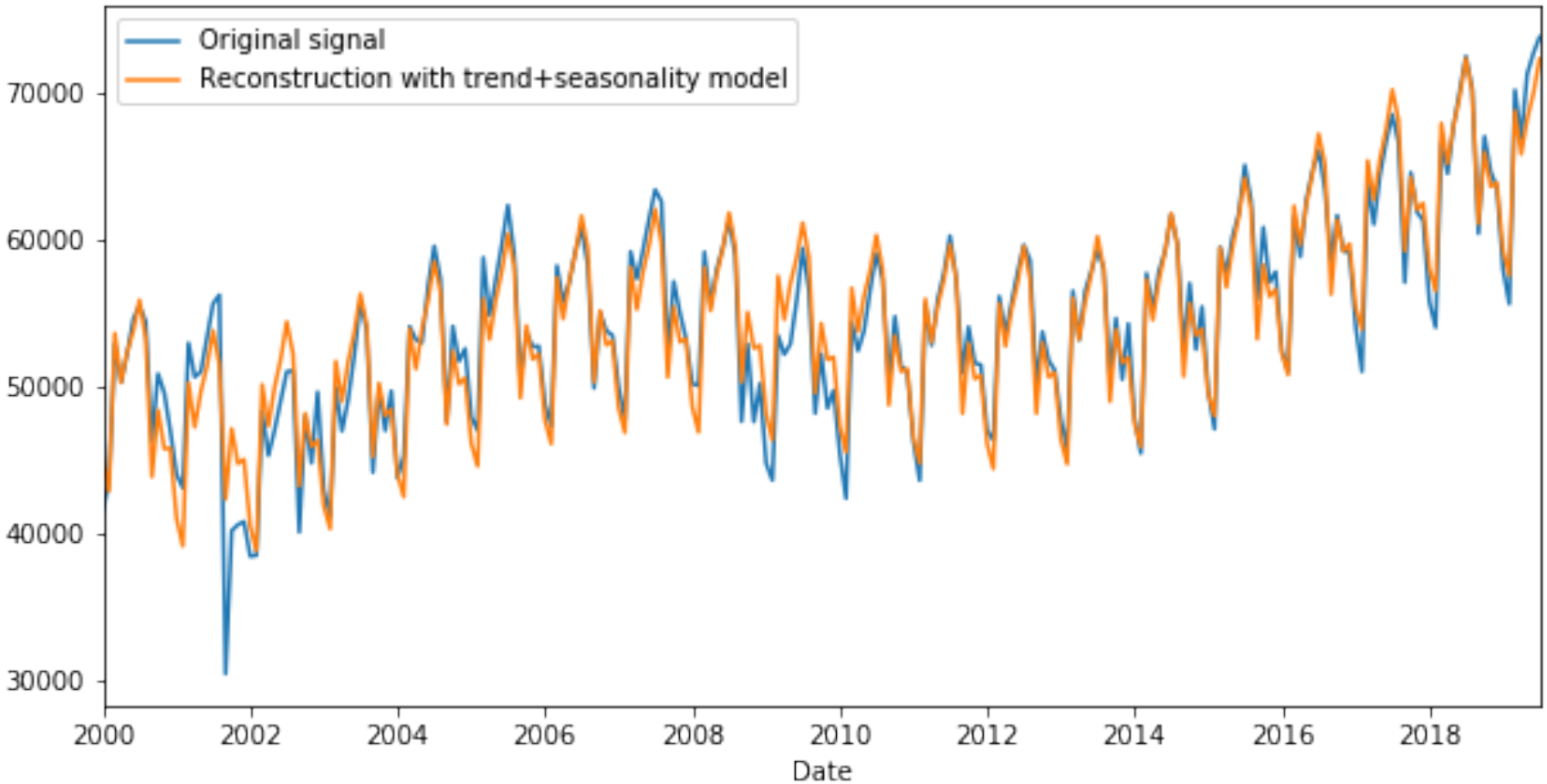


Une seule anomalie détectée

Détection d'anomalie basée sur un modèle

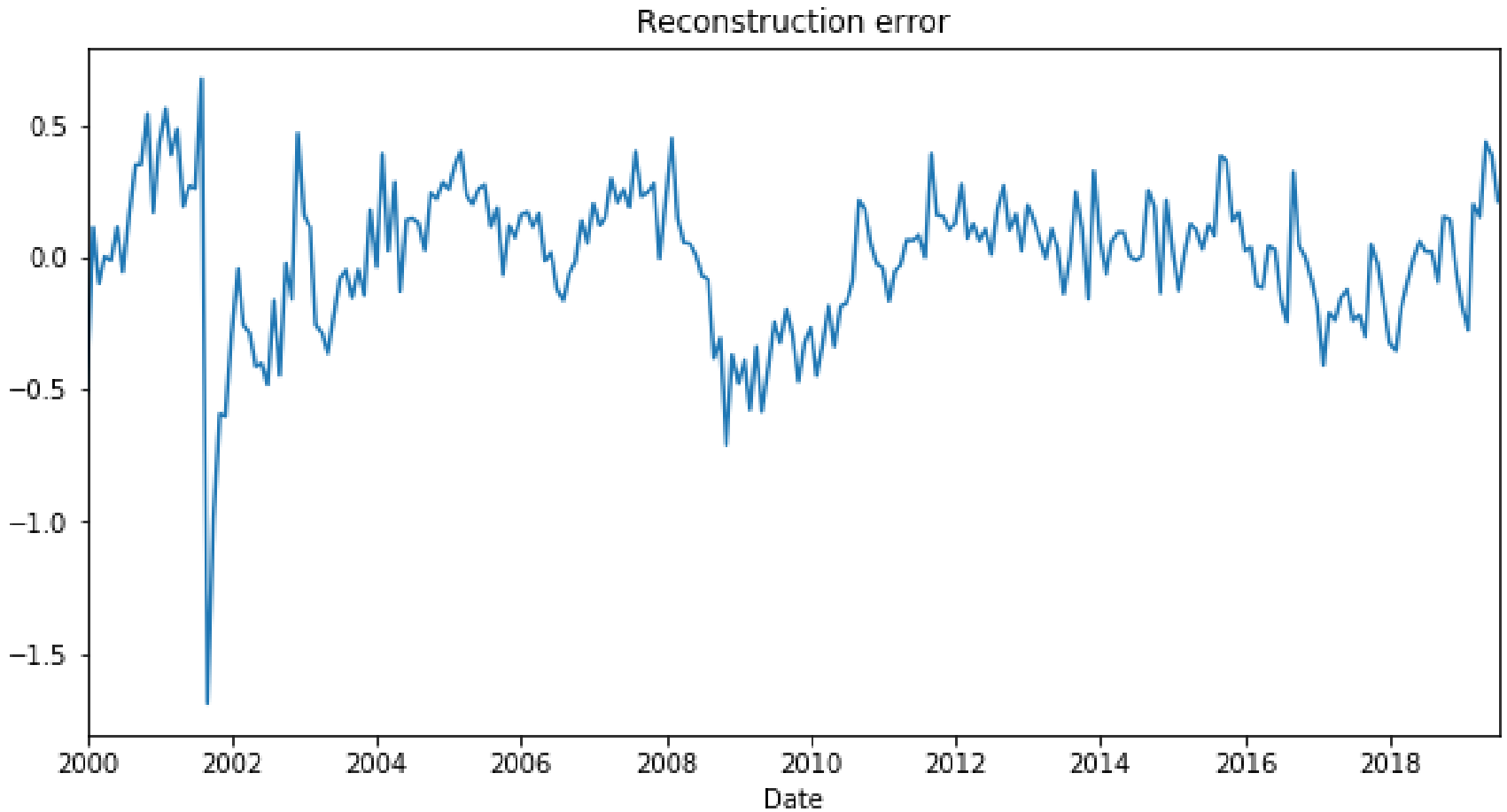
- Idée : utiliser un modèle pour détecter des anomalies [Yamanishi et al., 2002; Hill et al., 2010]
- Avantage : permet de prendre en compte les aspects temporels
- Trois étapes :
  1. Choisir un modèle adéquat et apprendre les paramètres
  2. Calculer la prédiction ou le signal reconstruit avec le modèle
  3. Détecter les anomalies comme les échantillons qui divergent de ce modèle

Exemple : tendance et saisonnalité



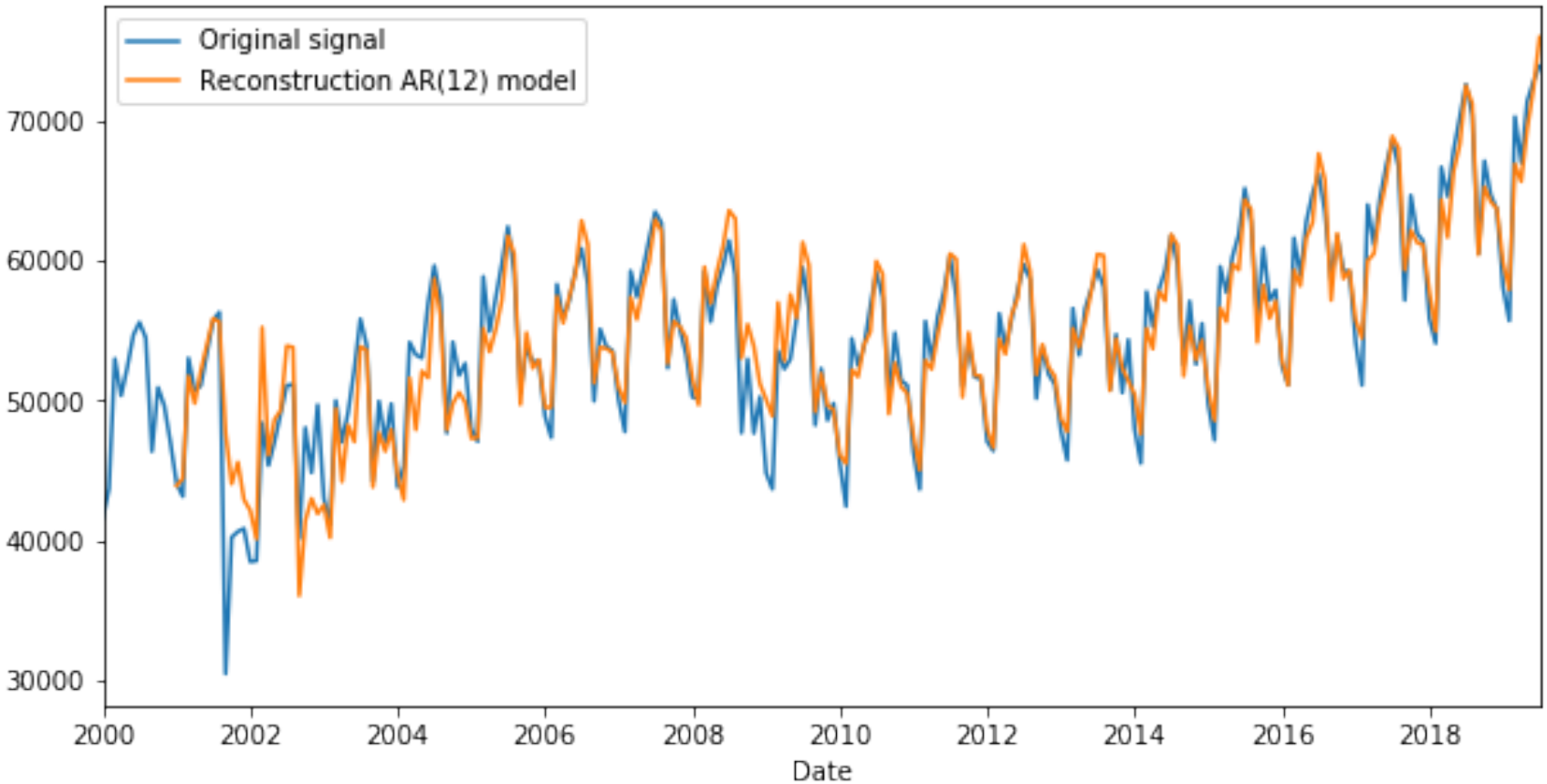
- Tendance : polynôme de degré 4
- Saisonnalité : fonction cosinus/sinus avec des fréquences fondamentales multiples de $\frac{1}{12}$

Exemple : tendance et saisonnalité



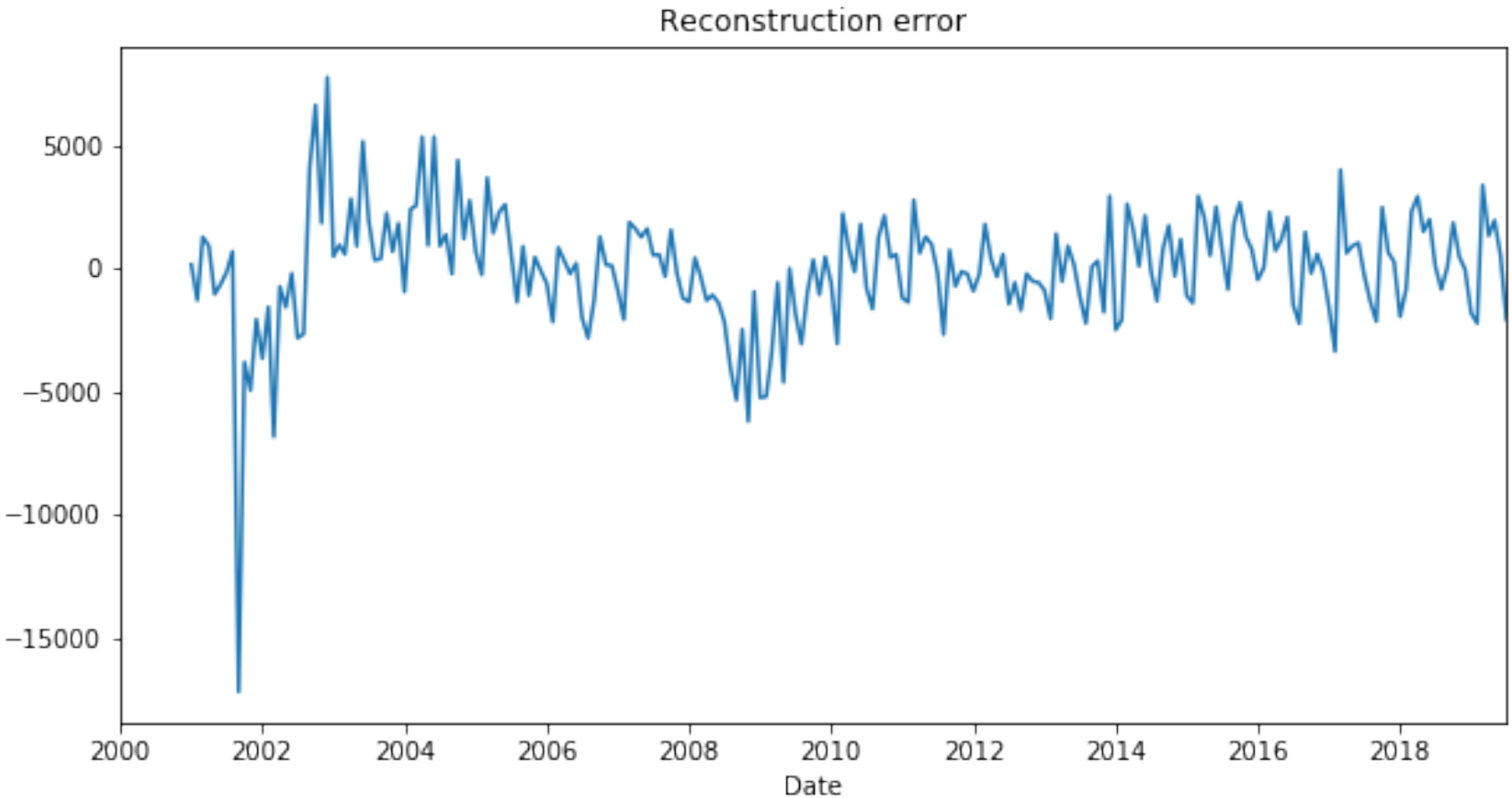
On ne détecte par seulement les valeurs grandes/petites, mais celles qui sont incohérentes du point de vue de l'évolution temporelle

Exemple : modèle AR



Modèle AR avec $p = 12$

Exemple : modèle AR



L'anomalie change également la prédiction sur les p échantillons suivants

Détection d'événements inusuels

- Certaines anomalies peuvent être plus complexes à détecter car elles ne sont pas caractérisées par des valeurs aberrantes, mais plutôt comme un nouveau comportement qui n'avait pas été observé jusque là
- Dans ce cas, une anomalie peut être définie comme une divergence par rapport à un comportement normal
- En réalité, cette tâche est la tâche duale de l'extraction de motifs : au lieu de rechercher des motifs répétitifs, on va chercher des événements qui ne se répètent pas !

Rappel : le matrix profile

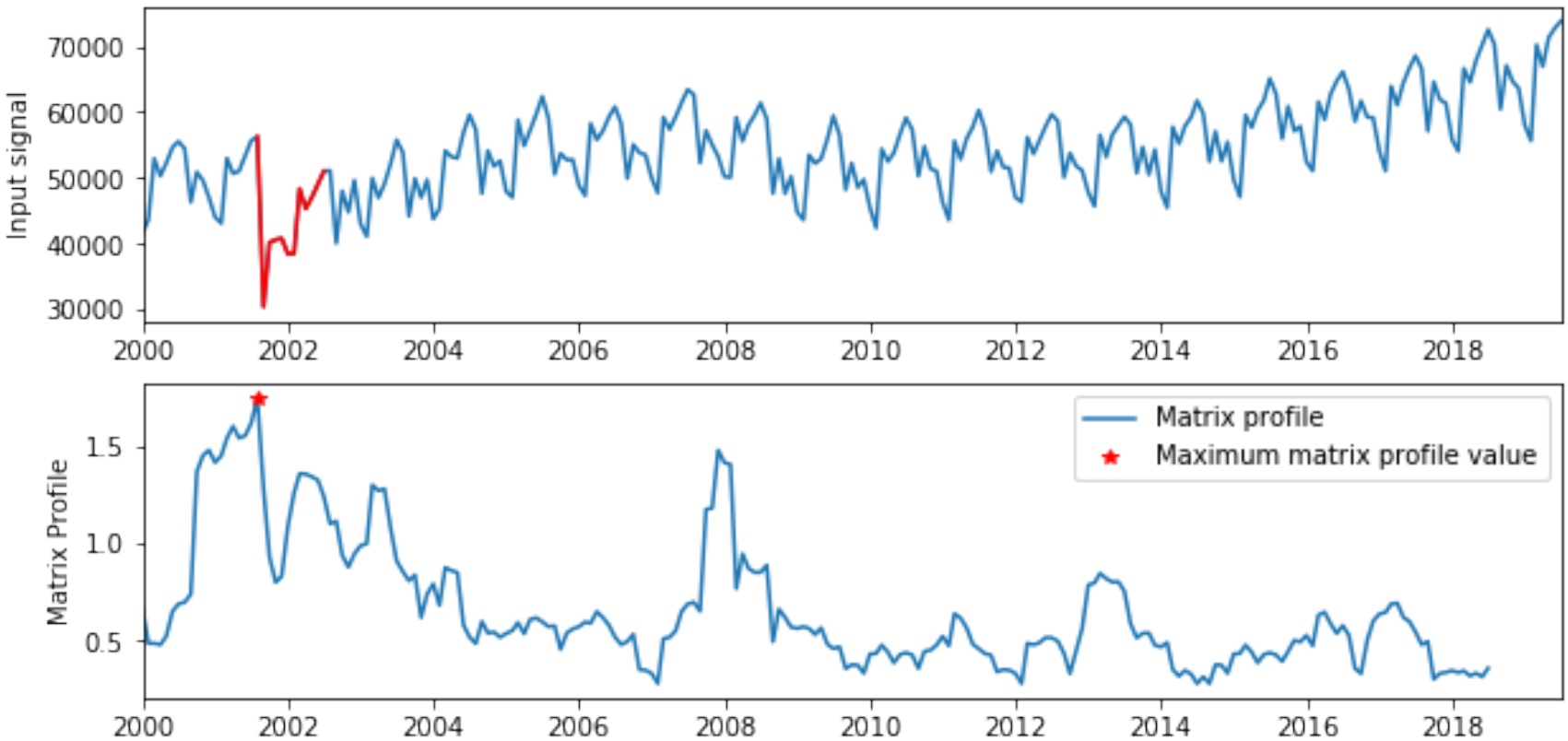
- Rappel : **Matrix profile** [Yeh et al., 2016] : étant donnée une taille de motifs L, on calcule

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

- Les petites valeurs de matrix profile indiquent que la sous-séquence a été trouvée ailleurs dans la séries temporelle, suggérant ainsi qu'il peut s'agir d'un motif
- Le matrix profile peut être calculé de façon très efficace avec la distance euclidienne renormalisée

Que peut-on dire des grandes valeurs du matrix profile ?

Exemple : matrix profile



Matrix profile avec une taille de fenêtre de $L = 12$ mois

Matrix profile

- En observant les grandes valeurs du matrix profile, on peut détecter des anomalies
- Les sous-séquences qui sont *éloignées* de toutes les autres sous-séquences du signal sont susceptibles de correspondre à de nouveaux comportements
- Avantage : pas besoin d'un modèle paramétrique
- Nécessite en revanche d'avoir une idée au moins approximative de l'échelle des anomalies (paramètre L)

Références

- 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.
- 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.
- 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.
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection : A survey. ACM computing surveys (CSUR), 41(3), 1-58.
- 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.