Detecting Changes in Slope With an L_0 penalty

Sacha HAKIM sacha.hakim@dauphine.eu
Quentin MOAYEDPOUR qmoayedpour@ensae.fr

January 2024

1 Introduction and contributions

In many fields, we are led to observe non-homogeneous time series that necessitate to be analyzed. *Change Point Detection (CPD)* makes this task much easier by dividing a series into several homogeneous segments, which is why it has attracted so much interest. CPD also involves choosing the right number of change points since, in many real-world applications, this is not known a priori. The solutions proposed in the literature generally consist of minimizing a cost function either with a penalty or under a constraint of a maximum number of segments (or both).

State-of-the-art algorithms such as PELT and FPOP can be applied to a wide range of cost functions, enabling the detection of various types of change points. In particular, they have proved highly effective for mean change detection (the most common task of CPD). However, such modeling may not be very relevant for certain types of time series. In particular, we quite often observe series whose mean evolves continuously rather than by jumps, which motivates us to try fitting a continuous-piecewise-linear mean function instead. Maidstone et al. (2019) proposed a *Continuous-piecewise-linear Pruned Optimal Partitioning (CPOP)* algorithm. This algorithm uses dynamic programming and two pruning processes, to efficiently solve the problem. It allows to find optimal change points under a ℓ_0 penalty, while being computationally feasible on fairly large series.

Our contribution includes a Python implementation of the **Continuous-piecewise-linear Pruned Optimal Partitioning (CPOP)** algorithm introduced by the authors. The authors provided very limited details regarding the selection of the residuals and the penalty of the model which, however, play a crucial role in the performance of the algorithm. In this work, we propose a simple and effective method to select and compute these two parameters efficiently. Our approach is evaluated on synthetic datasets and real-world datasets without utilizing their annotations, allowing for a qualitative analysis of the results. To the best of our knowledge, the only python's implementation of the CPOP algorithm can be found here (From master MVA's student alumni). We used it to verify whether our calculations and results were correct. Otherwise, most of the functions were developed based on the article, except for the calculation of the coefficients once the change points were determined.

Both members of the group worked together on nearly all parts of the project. MOAYEDPOUR focused more on the code while HAKIM on the theoritical comprehension of the algorithm. We designed jointly the experimental setup.

2 Method

2.1 Model and Criterion

First, for $t \geq 1$, let's define the family of change points set over $[\![1, t]\!]$ as:

$$\mathcal{T}_t = \{\tau^m = \{\tau_1 < \cdots < \tau_m\} \subset [\![1, t-1]\!], m \geq 0\}.$$

For convenience, we also define $\mathcal{T}_0 = \{\tau^{-1} = \emptyset\}$. Moreover, for all $\tau^m \in \mathcal{T}_t$, we set $\tau_0 = 0$ and $\tau_{m+1} = t$. A continuous piecewise linear function on $[0, t]$ can be parameterized by its points of slope changes $\tau^m \in \mathcal{T}_t$ and the values $\phi_{0:m+1} = (\phi_0, \ldots, \phi_{m+1})$ of the function at these change points, as long as changes only occur at $[\![1, t-1]\!]$. We consider a time series $y = \{y_1, \ldots, y_n\} \in \mathbb{R}^n$, with $n >> 1$. We want to fit a piecewise linear continuous function to y. This can be formalized as a statistical parametric estimation problem. To do this, we model the time series y as the noisy discrete observation of a continuous piecewise linear function. Thus, we assume that there exists $\tau^m \in \mathcal{T}_n$ and $\phi_{0:m+1} \in \mathbb{R}^{m+2}$ such that, for all $t \in [\![\tau_i + 1, \tau_{i+1}]\!]$, y_t is the observation of the following random variable: $Y_t = \phi_i + \frac{\phi_{i+1}-\phi_i}{\tau_{i+1}-\tau_i}(t - \tau_i) + Z_t$, where Z is an independent and centered noise with variance $\sigma^2 > 0$. In the following, we assume that we know σ^2 (in practice we can estimate it beforehand with Median Absolute Deviation for example) while τ^m and $\phi_{0:m+1}$ are not, and we want to estimate it. To do it, we have to define a criterion to be minimized, making a compromise between a good fit and a fit that is not too complex. First, for all $0 \leq s < t \leq n$ and $(\phi, \psi) \in \mathbb{R}^2$, we set:

$$\mathcal{C}(y_{s+1:t}, \phi, \psi) = \frac{1}{\sigma^2} \sum_{i=s+1}^{t} (y_i - \phi - \frac{\psi - \phi}{t - s}(j - s).$$

We will try to estimate the parameters by minimizing the following criterion over $\tau^m \in \mathcal{T}_n$ and $\phi \in \mathbb{R}^{m+2}$:

$$\mathcal{V}(\tau^m, \phi; y) = \sum_{i=0}^{m} \left(\mathcal{C}(y_{\tau_i+1:\tau_i}, \phi_i, \phi_{i+1}) + h(\tau_{i+1} - \tau_i) + \beta \right),$$

where $\beta > 0$ is a L_0-penalty constant and h is a segment-length penalty.

2.2 Dynamic Programming Approach

Let $1 \leq t \leq n$ and $\phi \in \mathbb{R}$. We define $f^t(\phi)$ as the minimal cost of $y_{1:t}$ when the last point of the fitting continuous piecewise linear function is ϕ. More formally:

$$f^t(\phi) = \min_{\tau^k \in \mathcal{T}_t, \phi_{0:k} \in \mathbb{R}^{k+1}} \sum_{i=0}^{k-1} \left[\mathcal{C}(y_{\tau_i+1:\tau_{i+1}}, \phi_i, \phi_{i+1}) + h(\tau_{i+1} - \tau_i) \right] + \mathcal{C}(y_{\tau_k+1:t}, \phi_k, \phi) + h(t - \tau_k) + \beta(k+1).$$

Posing $f^0(\cdot) \equiv 0$, one gets the following Bellman equation:

$$f^t(\phi) = \min_{0 \leq s < t, \phi' \in \mathbb{R}} f^s(\phi') + \mathcal{C}(y_{s+1:t}, \phi', \phi) + h(t - s) + \beta.$$

Now let's also fix the change points $\tau^k \in \mathcal{T}_t$. We define $f^t_{\tau^k}(\phi)$ as the minimal cost of $y_{1:t}$ for fitting a function with change points τ^k and last value $\phi_{m+1} = \phi$, that is:

$$f^t_{\tau^k}(\phi) = \min_{\phi_{0:k} \in \mathbb{R}^{k+1}} \sum_{i=0}^{k-1} \left[\mathcal{C}(y_{\tau_i+1:\tau_{i+1}}, \phi_i, \phi_{i+1}) + h(\tau_{i+1} - \tau_i) \right] + \mathcal{C}(y_{\tau_k:t}, \phi_k, \phi) + h(t - \tau_k) + \beta(k+1).$$

Posing again $f^0_{\tau^{-1}}(\cdot) \equiv 0$, we get another Bellman equation:

$$f^t_{\tau^m}(\phi) = \min_{\phi' \in \mathbb{R}} f^{\tau_k}_{\tau_{1:k-1}}(\phi') + \mathcal{C}(y_{\tau_k+1:t}, \phi', \phi) + h(t - \tau_k) + \beta.$$

Using the formula above, we can prove that the function $f^t_{\tau^m}$ are strictly convex quadratic. Hence we can represent this function with a vector $(a^t_{\tau^m}, b^t_{\tau^m}, c^t_{\tau^m}) \in \mathbb{R}^2 \times \mathbb{R}_{++}$. Moreover, we can compute those coefficients with a recurrence formula. We give all the details about it in the Appendices. This result is very important as it tells us that, for any change points set $\tau^m \in \mathcal{T}_n$, we can find optimal values $\phi^*_{\tau^m}$ by recursively minimizing strictly convex quadratic functions $f^{\tau_k}_{\tau_{1:k-1}}$. Hopefully, we have simple analytic formula for such minimization problems, hence the complexity for this task is only to compute recursively the quadratic coefficients. Then, we can find the optimal change points by minimizing the function

$\mathcal{V}(\tau^m, \phi^*_{\tau^m}; y)$. In particular, it ensures that the minimization problem $\min \mathcal{V}(\tau^m, \phi; y)$ as a solution.

2.3 Pruning Techniques

The authors proposed two methods to reduce the number of admissible change points sets. The first one is functional-based, while the second one is inequality-based.

2.3.1 Functionnal Pruning

For all $1 \leq t \leq n$, we let $\mathcal{T}_t^* = \left\{\tau \in \mathcal{T}_t \,\middle|\, \exists \phi \in \mathbb{R}, f^t(\phi) = f^t_\tau(\phi)\right\}$ be the family of change points set that are optimal for some terminal value ϕ. Using the dynamic programming formulas stated above, the authors proved the following theorem:

Theorem 1. *For all $1 \leq s < t \leq n$ such that $\tau \notin \mathcal{T}_s^*$, we have $\tau \cup \{s\} \notin \mathcal{T}_t^*$.*

Let $\hat{\mathcal{T}}_t$ be recursively defined by:

- $\hat{\mathcal{T}}_1 = \{\emptyset\}$,
- $\hat{\mathcal{T}}_{t+1} = \hat{\mathcal{T}}_t \cup \{\tau \cup \{t\} : \tau \in \mathcal{T}_t^*\}$ for all $1 \leq t \leq n-1$.

The theorem above actually means that $\mathcal{T}_t^* \subset \hat{\mathcal{T}}_t$ for all $1 \leq t \leq n$.

2.3.2 Inequality Based Pruning

Taking inspiration from Killick et al. (2012), the authors also proved that:

Theorem 2. *Let $K = 2\beta + h(1) + h(n)$. We assume that h in nonnegative and nonincreasing. Let $1 \leq s \leq n-1$ and $\tau \in \mathcal{T}_s$ be such that:*

$$\min_{\phi \in \mathbb{R}} f^s_\tau(\phi) > \min_{\phi \in \mathbb{R}} f^s(\phi) + K.$$

Then, for all $s < t \leq n$, we have:

$$\min_{\phi \in \mathbb{R}} f^t_\tau(\phi) > \min_{\phi \in \mathbb{R}} f^t(\phi).$$

Combining the two previous theorems, provided that h is nonnegative and nonincreasing, we can further reduce the set of candidates by posing the following.

- $\hat{\mathcal{T}}_1 = \{\emptyset\}$,
- $\hat{\mathcal{T}}_{t+1} = \left\{\tau \in \hat{\mathcal{T}}_t \cup \{\overline{\tau} \cup \{t\} : \overline{\tau} \in \mathcal{T}_t^*\} \,\middle|\, \min_{\phi \in \mathbb{R}} f^t_\tau(\phi) \leq \min_{\phi \in \mathbb{R}} f^t(\phi) + K\right\}$ for all $1 \leq t \leq n-1$.

2.4 CPOP Algorithm

From now on, we assume that h is nonnegative and nonincreasing. Combining the previous results, and provided we know how to calculate $\mathcal{T}_t^*$ from $\hat{\mathcal{T}}_t$, we should be able to implement a dynamic programming algorithm for minimizing $\mathcal{V}(\tau^m, \phi; y)$ that is fairly efficient. Indeed, we could then recursively compute the set $\mathcal{T}_n^*$, which would allow us to simply deduce the optimal change points by minimizing the quadratic functions $f^n_\tau(\cdot)$ for all $\tau \in \mathcal{T}_n^*$ and then selecting the candidate $\tau \in \mathcal{T}_n^*$ giving the smallest value of $\min_{\phi} f^n_\tau(\phi)$. This is the idea behind the CPOP algorithm (see **Algorithm 1, Appendix E** of Maidstone et al. (2019)). It still remains to find how to compute $\mathcal{T}_t^*$ from $\hat{\mathcal{T}}_t$. In fact, to solve this task, we just need to find the subset of $\mathbb{R}$ over which $f^t_\tau(\phi) = f^t(\phi)$ for all $\tau \in \hat{\mathcal{T}}_t$: if this set is empty for some τ, then we can remove this candidate from $hat\mathcal{T}_t$. And to solve this second task, we can use a line search, as proposed the authors (see **Algorithm 2, Appendix E** of Maidstone et al. (2019)). To simply understand this approach, remember that every function f^t_{tau} is strictly convex quadratic and therefore has a parabolic form. We can then visualize quite intuitively that the set $\mathbb{R}$ is divided into closed intervals over which a τ realizes $f^t_\tau(\phi) = f^t(\phi)$.

Both pruning techniques allows us to considerably reduce the computation time and space for the algorithm. For instance, the run time on a simple example is less than 0.1 seconds when using the pruning techniques, while more than 5 minutes when we don't prune the suboptimal τ vectors.

3 Parameters of the algorithm

The algorithm has three parameters: The penalty constant β, associated with the penalty ℓ_0, the cost functions of the lengths of the segments h and the standard deviation of the residuals σ^2. In this report, we will focus on the parameters β and σ. We will set $h(x) = \gamma log(x)$ with $\gamma = 1$.

3.1 Statistical Criterion to select β

In the original paper, the authors demonstrate that the algorithm asymptotically converges to an optimal solution when $\beta \geq C_1 \log(n)$, where C_1 is a constant that cannot be computed explicitly. Consequently, when searching for the optimal value of β, rather than using a traditional grid search, one can multiply all candidate values by $\log(n)$, which helps to narrow the search space and potentially improves computational efficiency. Furthermore, since β is a parameter of the model, we aim to have some selection criterion to choose it. For instance, if we suppose that the residuals of the model Z_t follows a normal distribution, we can deduce the likelihood of the model:

$$\log \mathcal{L}(y; \hat{y}, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2$$

$$\text{where} \quad \hat{y}_t = \hat{\phi}_{\tau_i} + \frac{\hat{\phi}_{\tau_{i+1}} - \hat{\phi}_{\tau_i}}{\tau_{i+1} - \tau_i}(t - \tau_i)$$

And the vector τ is estimated by the model. From the likelihood, we can compute some information criterion such as the AIC, BIC, or modified BIC (Zhang et al. 2007, Truong et al. 2020). However, that estimation highly depends on σ and its estimation.

3.2 Estimation of σ

Figure 2 illustrates that the algorithm's results are highly sensitive to the parameter σ, which also plays a critical role in the potential statistical criterion. If the estimation of σ using the median absolute deviation (MAD) method (Hampel 1974) is relatively robust to outliers and generally provides reliable estimates, it can occasionally exhibit bias, leading to inaccuracies. Since the true value of τ in the model is required for an accurate estimation, it can be dynamically refined during the process. Specifically, after estimating the coefficients of the model for a given τ, $\hat{\sigma}$ can be updated using the formula[1] the average squared error of the model.

Our experiments demonstrate promising results when initializing σ with an estimation based on the median absolute deviation and a low value of β. Dynamically updating σ while progressively increasing the penalty β improves performance. The final result is selected by minimizing an information criterion,

[1] Since the model minimise the squared errors at each segment, the residuals means is often close to 0 and has an expectation of 0

such as BIC or mBIC. We saw that starting from a low value of β to a large one shows better results than starting from a large one (Figures 3 & 4).

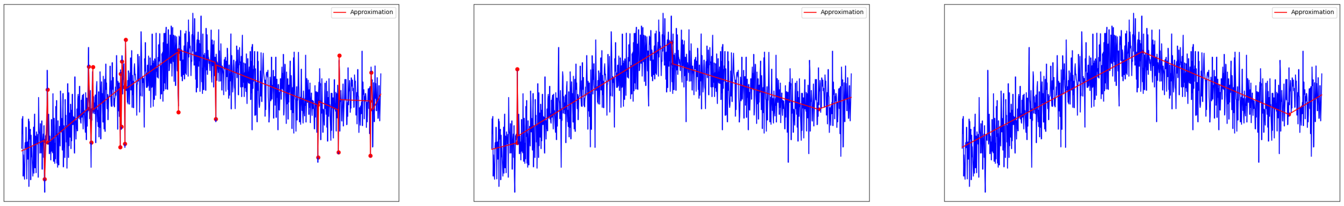
4 Experiments

4.1 Datasets

We conducted experiments using both synthetic and real-world datasets. For the synthetic data, we generated processes based on the statistical model defined in Equation (1). For the real-world data, we selected time series from the Turing Change Point Dataset (TCPD) Burg et al. 2022. However, we opted to use only the time series from this dataset, excluding their associated labels and metrics. The primary objective was to evaluate the algorithm's performance on real-world data to observe how the model performs on arbitrary datasets, without the intention of assessing or comparing its accuracy. Given that the model assumes the data follows a specific generative process, testing it on data that does not adhere to this assumption would be irrelevant. In this context, we specifically chose time series from the TCPD that exhibit linear trends. For example, time series depicting a country's GDP over time generally display linear trends over extended periods, with breaks that can be attributed to significant economic events. Additionally, the data from the TCPD are already well-prepared, meaning no pre-processing was required.

4.2 Results

To select the model's hyperparameters (β and σ), we empirically found that starting with a low value for β and gradually increasing it, while dynamically recalculating σ at each step, led to satisfactory results, on generated and real world data (Figures 5 & 6). These results were obtained by selecting the pair of β and σ that minimizes an information criterion, such as BIC or AIC. When the time series contains a small number of data points, the three criteria (AIC, BIC, and mBIC) tend to select the same model. However, as the number of data points increases, AIC and BIC tend to favor a model that overfits, while mBIC selects more parsimonious solutions.



(a) AIC

(b) BIC

(c) mBIC

Figure 1: Selected solution on synthetic data for different criterions.

The previous figure shows that the algorithm is highly sensitive to outliers, which can lead to an overestimation of the number of changepoints. However, since we assume that the distribution of the time series changes over time, traditional outlier removal models cannot be applied, as they rely on the assumption of a "stable" distribution over the serie. It is mainly due to the fact that the model aim to minimize a square error cost, the presence of extreme values can have a big impact on the estimations.

Removing the noise from the serie using noise removal techniques (SSA approximation for instance) doesn't improve the results. Since the serie will have a low amount of noise, the model will tend to overfit and the selections criterions will not be efficients (see Figure 7).

The empirical selection of hyperparameters β and σ using information criteria (AIC, BIC, mBIC) effectively balances model fit and complexity. However the model rely on strong assumptions about the serie and may not fit everytime.

References

G. J. J. van den Burg et al. (2022). *An Evaluation of Change Point Detection Algorithms*. arXiv: 2003.06222 [stat.ML]. URL: https://arxiv.org/abs/2003.06222.

C. Truong et al. (2020). "Selective review of offline change point detection methods". In: *Signal Processing* 167.

R. Maidstone et al. (2019). "Detecting Changes in Slope With an L_0 Penalty". In: *Journal of Computational and Graphical Statistics* 28.2, pp. 265–275.

R. Killick et al. (2012). "Optimal Detection of Changepoints with a Linear Computational Cost". In: *Journal of the American Statistical Association* 107.500, pp. 1590–1598.

N. Zhang et al. (2007). "A Modified Bayes Information Criterion with Applications to the Analysis of Comparative Genomic Hybridization Data". In: *Biometrics* 63, pp. 22–32.

F. R. Hampel (1974). "The Influence Curve and its Role in Robust Estimation". In: *Journal of the American Statistical Association* 69.346, pp. 383–393. DOI: 10.1080/01621459.1974.10482962.

5 Appendices

5.1 Quadratic Coefficients Calculation

We found an MVA alumni project dealing with the same article. They indicated that there was an error in the calculations of the quadratic coefficients, so we decided to redo them. We won't go into detail about all the computations, but we will state the results we found.

Let $1 \leq t \leq n$, $(\phi, \phi') \in \mathbb{R}^2$ and $\tau^k \in \mathcal{T}_t$. Posing $s = t - \tau^k$, we get:

$$\forall 1 \leq t \leq n, \quad (\phi, \phi') \in \mathbb{R}^2, \quad \tau_k \in \mathcal{T}_t, \quad \text{we have:}$$

$$\begin{aligned}
\mathcal{C}(y_{\tau_k+1:t}, \phi', \phi) &= \left[\frac{(s+1)(2s-1)}{6s\sigma^2}\right]\phi^2 + \left[\left(\frac{s+1}{\sigma^2} - \frac{(s+1)(2s+1)}{3s\sigma^2}\right)\right]\phi'\phi \\
&+ \left[-\frac{2}{6s\sigma^2}\sum_{j=\tau_k+1}^{t} y_j(j-\tau_k)\right]\phi + \left[\frac{1}{\sigma^2}\sum_{j=\tau_k+1}^{t} y_j^2\right] \\
&+ \left[2\left(\frac{1}{s\sigma^2}\sum_{j=\tau_k+1}^{t} y_j(j-\tau_k) - \frac{1}{\sigma^2}\sum_{j=\tau_k+1}^{t} y_j\right)\right]\phi' \\
&+ \left[\frac{(s-1)(2s-1)}{6s\sigma^2}\right]\phi \\
&= A\phi^2 + B\phi'\phi + C\phi + D + E\phi' + F\phi'^2
\end{aligned}$$

We want to show that $f^{\tau_i}_{\tau_0,\cdots\tau_{i-1}}$ is quadratic in ϕ.

$$\begin{aligned}
f^{\tau_1}_{\tau_0} &= \min_{\phi'\in\mathbb{R}}\left(A\phi^2 + B\phi'\phi + C\phi + D + E\phi' + F\phi'^2 + h(\tau_1) + \beta\right) \\
&= A\phi^2 + C\phi + D + h(\tau_1) + \beta + \min_{\phi'\in\mathbb{R}}\left(B\phi\phi' + E\phi' + F\phi'^2\right) \\
&= A\phi^2 + C\phi + D + h(\tau_1) + \beta - \frac{(E+B\phi)^2}{4F}\mathbb{I}_{\{F>0\}} \\
&= \left(A - \frac{B^2}{4F}\right)\phi^2 + \left(C - \frac{2EB}{4F}\right)\phi + \left(D + h(\tau_1) + \beta - \frac{E^2}{4F}\right) \\
&= c^{\tau_1}_{\tau_0}\phi^2 + b^{\tau_1}_{\tau_0}\phi + a^{\tau_1}_{\tau_0}
\end{aligned}$$

So we can compute recursively

$$\begin{aligned}
f^{\tau_{i+1}}_{\tau_0,\cdots\tau_i} &= min_{\phi'\in\mathbb{R}}\left(a^{\tau_i}_{\tau_0,\cdots\tau_{i-1}} + b^{\tau_i}_{\tau_0,\cdots\tau_{i-1}}\phi' + c^{\tau_i}_{\tau_0,\cdots\tau_{i-1}}\phi'^2 + B\phi'\phi + E\phi' + F\phi'^2\right) \\
&+ A\phi^2 + C\phi + D + h(\tau_{i+1} - \tau_i) + \beta \\
&= \left(A - \frac{B^2}{4(c^{\tau_i}_{\tau_0,\cdots\tau_{i-1}} + F)}\right)\phi^2 + \left(C - \frac{(b^{\tau_i}_{\tau_0,\cdots\tau_{i-1}} + E)B}{2(c^{\tau_i}_{\tau_0,\cdots\tau_{i-1}} + F)}\right)\phi \\
&+ \left(D + a^{\tau_i}_{\tau_0,\cdots\tau_{i-1}} + h(\tau_{i+1} - \tau_i) + \beta - \frac{(E + b^{\tau_i}_{\tau_0,\cdots\tau_{i-1}})^2}{4(c^{\tau_i}_{\tau_0,\cdots\tau_{i-1}} + F)}\right)
\end{aligned}$$

Where we can deduce the coefficients of the quadratic form of $f^{\tau_{i+1}}_{\tau_0,\cdots\tau_i}$. At each step of the algorithm, we compute and store those coefficients and will use them

5.2 Figures

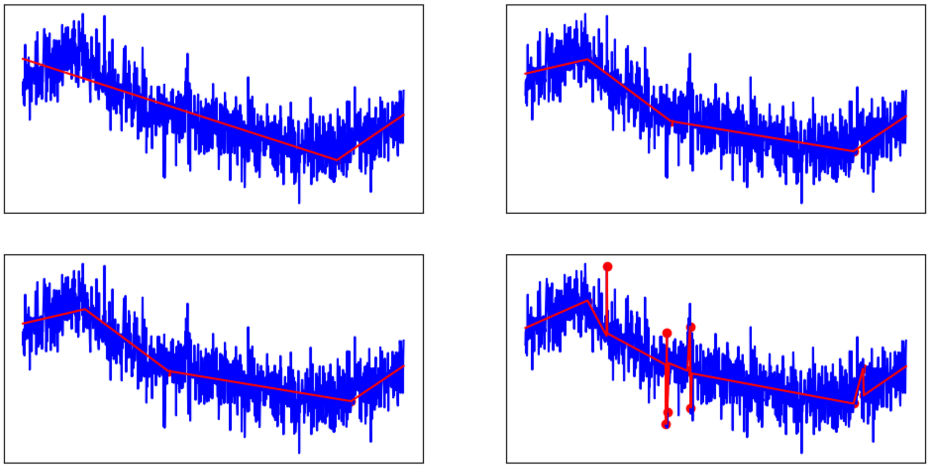


Figure 2: Estimation for different values of σ (all other parameters are fixed)

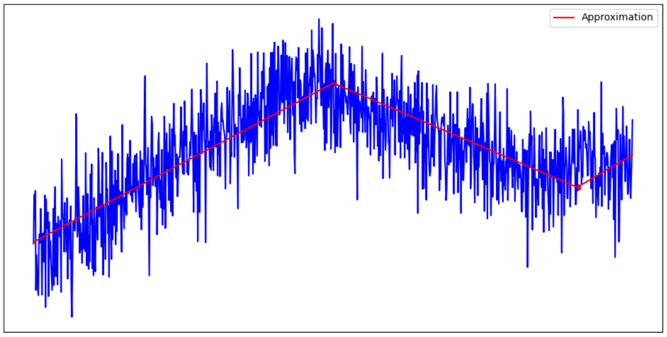


Figure 3: Model that minimize the mBIC when starting from a large value of β to a low one

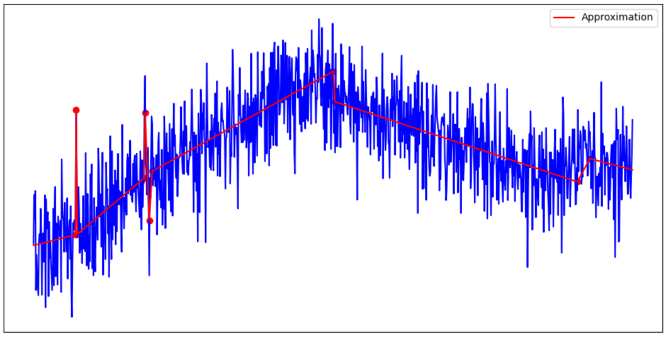


Figure 4: Model that minimize the mBIC when starting from a low value of β to a large one

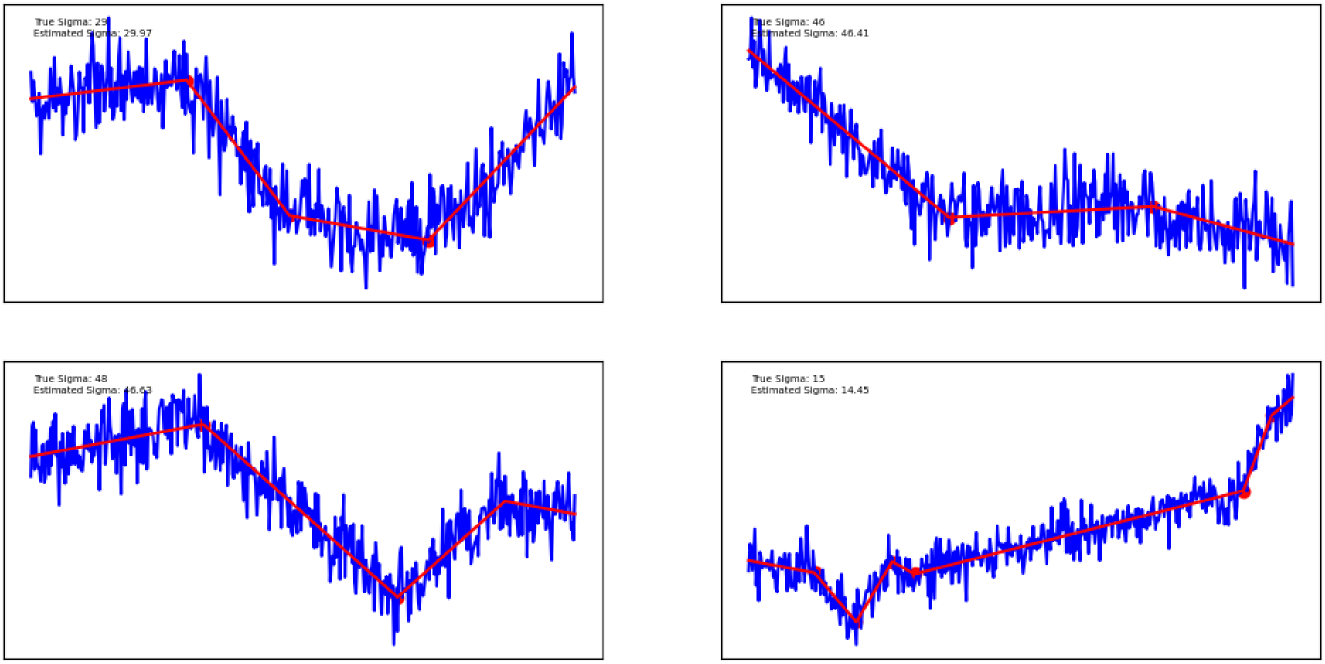
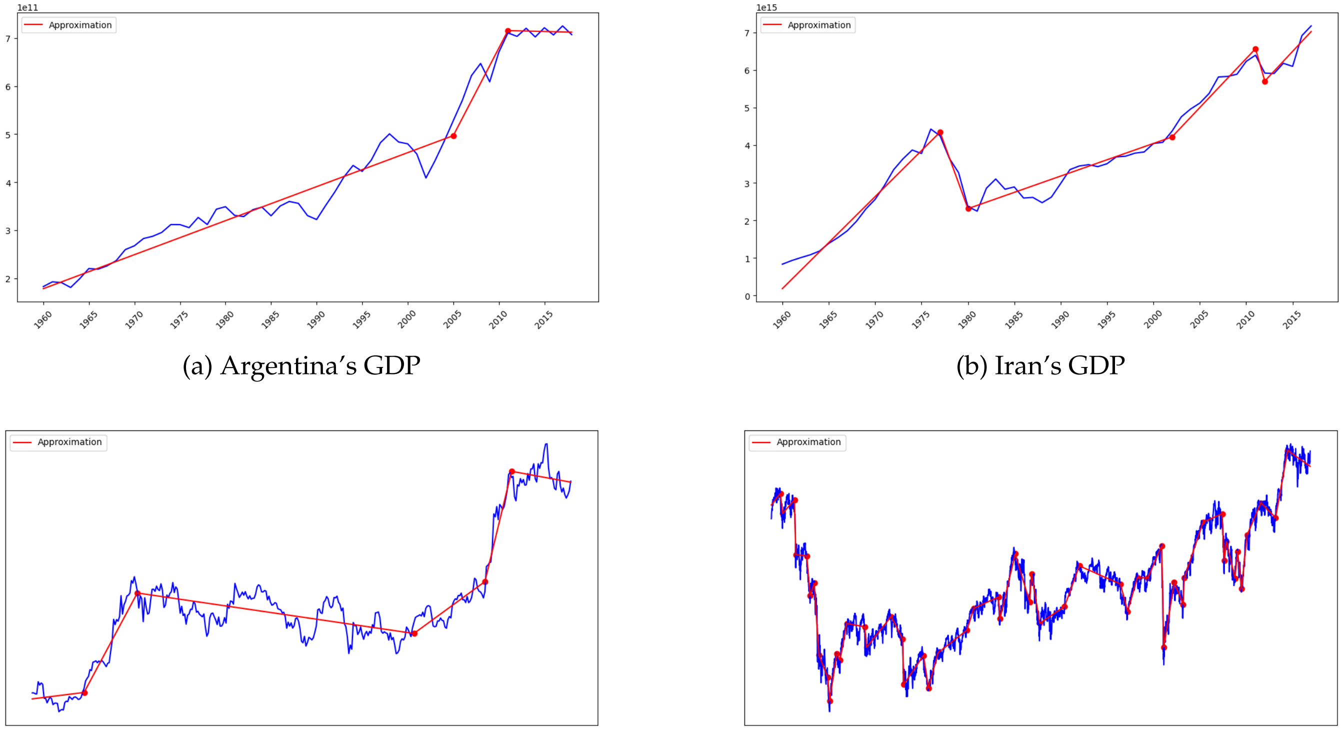


Figure 5: CPOP when updating σ dynamically and minimizing the modified BIC

5.2.1 Application on real worlds data



(a) Argentina's GDP

(b) Iran's GDP

(c) Bitcoin Price.

(d) EuroStoxx50 price

Figure 6: CPOP when minimizing the modified BIC.

The following figure presents the CPOP algorithm on a simulated serie where we artificially removed the noise.

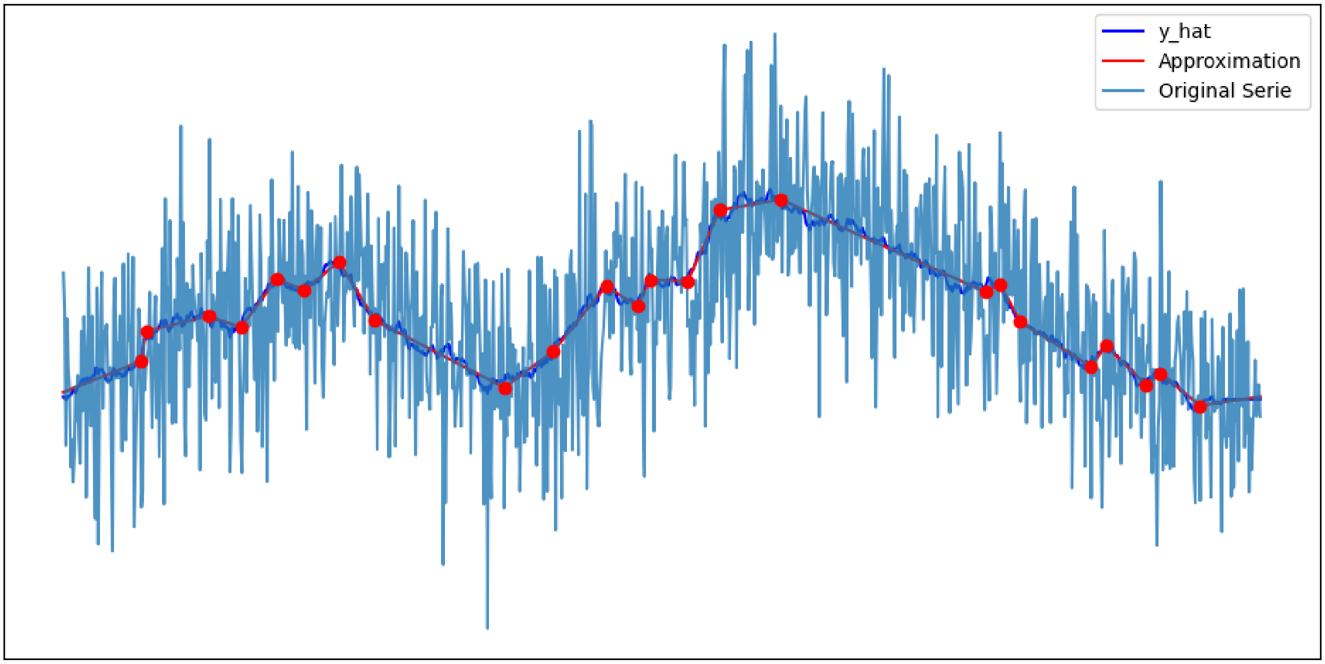


Figure 7: Change point detection on a noised-removed time serie (using SSA approximation)

The following figures shows the change point estimation for the algorithm PELT using the Mean Square Error of the regression of y with respect to time.

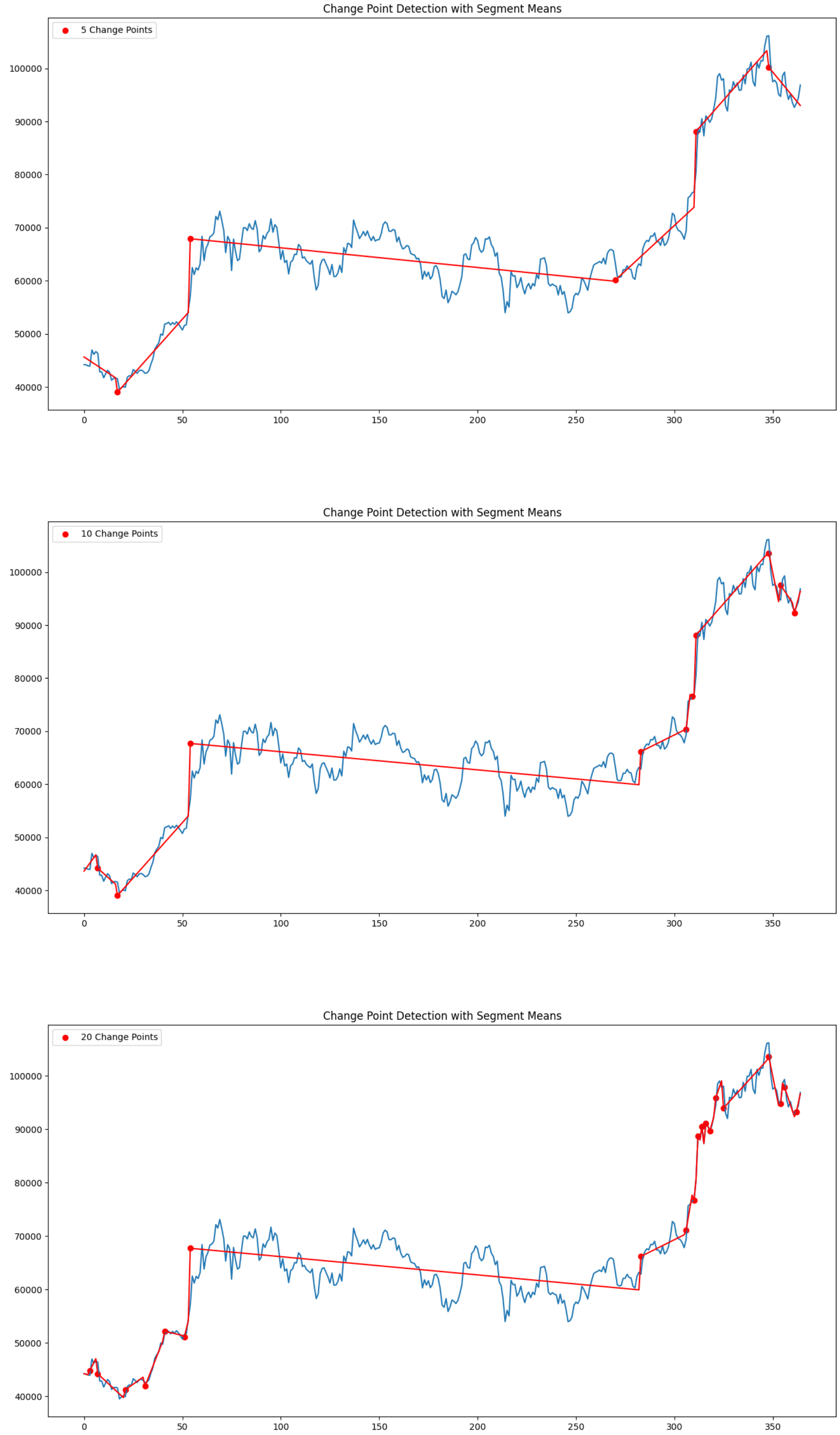


Figure 8: PELT using a linear cost