Urban Seismic Monitoring

Sébastien Meyer, Adrienne Tuynman

April 22, 2023

1 Introduction and contributions

Seismic data, in addition to describing earthquakes, can also be used to measure human activity. Behaviour such as walking and driving induce vibrations that can be detected. Therefore, applying change point detection to seismic data can give us information about sudden changes in traffic: sudden increases in noise (riots, cultural events), or sudden decreases in noise (evacuations, pandemic lockdowns).

The application of change point detection algorithms to urban seismic data was already tested out in Brasilia [1], and our aim is to replicate and extend those results to other places. In order to run our experiments, we decided to create a new dataset of seismic data. Indeed, available data was either incomplete or the pre-processing steps were not clearly identified. Sébastien wrote routines adapted from [2] which allow to download and pre-process seismic data. Contrary to [1], where the queried station is located 30km away from the city centre, ours is located directly in the city centre of Strasbourg, France. Moreover, Sébastien implemented wrappers to easily run experiments on processed data. Adrienne ran several experiments on the data and she generated the results by varying the parameters. Both of us formatted the notebooks and wrote the report.

Compared to the example presented in [1], our project uses seismic data from the French RÉSIF network of sensors. More specifically, we studied seismic data from a sensor located in the city centre of Strasbourg, France. This allowed us to have a refined detection of urban noise, while the sensor used in the original paper is located 30km away from the city centre of Brasilia. Besides, we ran experiments on both the power spectrum (multivariate, not presented in the report) and the root mean squared displacement (univariate).

2 Method

All our experiments were based on the *ruptures* package for change detection in Python, introduced in [3]. We tried both penalized models such as Pelt and models for which we provide the expected number of breakpoints such as Dynp.

2.1 Dynamic programming

Detecting change points means finding spots where some properties of the signal change: mean, median, etc. For that, one defines a cost function c that quantifies how regular a segment of signal is, and the goal is to minimize the sum of costs over the cut up segments. In other words, we are looking for:

$$C_k((y_t)_{0\leq t<n}) = \min_{0=n_0<...<n_{k+1}=n} \sum_{i=1}^{k+1} c\left((y_t)_{n_{i-1}\leq t<n_i}\right)$$

where k is the number of breakpoints. Dynamic programming, implemented in the *ruptures* package under `Dynp`, aims to find an exact solution of this problem when provided with the number of breakpoints. For that, it makes use of the property $C_k((y_t)_{0\leq t<n}) = \min_{0\leq n'<n-k+1} C_{k-1}((y_t)_{0\leq t<n'}) + c((y_t)_{n'\leq t<n})$, and keeps the results in memory so as not to compute the same $C_k((y_t)_{n'\leq t<n''})$ twice.

2.2 PELT

A more general setting is when the number of breakpoints to find is unknown. One then introduces a penalty function p, and looks for:

$$F(n) = min_k C_k((y_t)_{0\leq t<n}) + p(k)$$

In the linear penalty setting, p is a linear function, we will write $p(k) = \beta k$. PELT [4] was introduced in order to solve this problem. PELT works in a similar way to Dynamic programming. Indeed, it uses the fact that $F(n) = min_k F(k) + c((y_t)_{k\leq t<n}) + \beta$, where $F(0) = -\beta$. On top of that, it uses pruning to skip unnecessary calculations. Indeed, it works off the assumption that there exists a constant K such that cutting up a signal in two decreases the total cost of at least K: $\forall n', n''$, $c((y_t)_{0\leq t<n'}) + c((y_t)_{n'\leq t<n''}) + K \leq c((y_t)_{0\leq t<n''})$. In this situation, then if $F(n') + c((y_t)_{n'\leq t<n''}) + K \geq F(n'')$, there is no need to even consider n' as a possible last changepoint before $n > n''$, as any segmentation with n' as a last changepoint would be less beneficial than having one with n'' as a last changepoint.

With this pruning method, PELT provides an exact solution to the changepoint detection problem, and has a better complexity than a naive search over all possible segmentations.

2.3 Cost functions

As noted above, the aim is to minimize a sum of costs. For that, we need to have a cost function that quantifies how regular a certain segment is. There are multiple ways to define this cost, each aiming to detect different types of change.

First, the L2 cost is meant to detect changes on the mean of the signal. It measures the distance from the signal to its empirical mean. More formally, with $\overline{y} = \frac{1}{n}\sum_{t=0}^{n-1} y_t$ the empirical mean of the signal, $c((y_t)_{0\leq t<n}) = \sum_{t=0}^{n-1} \|y_t - \overline{y}\|_2^2$

This cost will be impacted by the span of the signal, which could lead to trouble down the line. However, that is not the case for the kernelized mean change. This cost function uses Φ, the feature map of the radial basis function kernel $x, y \mapsto \exp\left(-\gamma \|x-y\|^2\right)$, where γ depends on the data.

3 Data

Our data consists in seismic data queried from the French RÉSIF network. Then, we extracted both the power spectrum and the root mean squared displacement. In order to compute these, we based our implementation on the *obspy* package, introduced in [5].

3.1 The raw data

The raw data simply consists in the waveforms as measured by the broad band triaxial sensor under service in Strasbourg, France. We downloaded data between January 1st, 2020 and January 1st, 2022. In some cases, for unknown reasons, the query does not return any data for very small periods. Therefore, we might end up with missing values.

3.2 Pre-processing

From the raw data, we compute the power spectrum using the `PPSD` routine from *obspy*. The power spectral densities are computed using a window of 30 minutes with an overlap of 15 minutes, following [2]. In addition, the method proposed in McNamara et al. [6] and implemented in *obspy* performs a smoothing of the power spectral densities in frequency bins. Finally, root mean squared displacement is computed over the band of frequencies between $f_{\min} = 4$ and $f_{\max} = 14Hz$, as this corresponds to usual frequencies for urban noise [1]. The formula, deduced from Parseval's theorem, is as follows:

$$d_{amp}^2 = \int_{f_{\min}}^{f_{\max}} \frac{10^{\mathrm{psd}_{dB}(f)/10}}{(2\pi f)^4} df.$$

The resulting power spectral densities and root mean squared displacements contain both missing values and outliers, as shown in **Figure 1**. On the one hand, missing values can span a few hours in a day. On the other hand, outliers are quite difficult to detect as some outliers correspond to true events. To this end, we replace value outside $[-500, 0]$ by the median value for the power spectrum and values outside $[20nm, 100nm]$ by the median value for the root mean squared displacement.

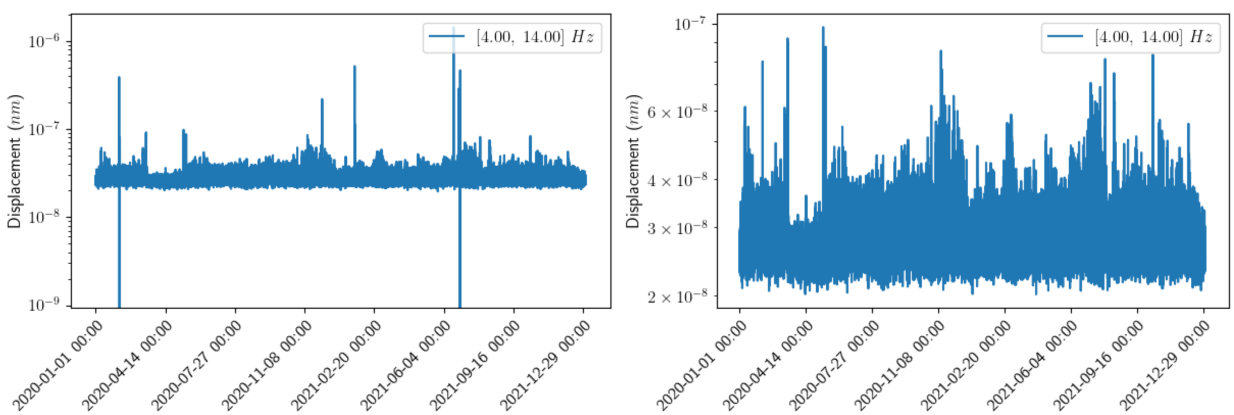


Figure 1: Root mean squared displacement values before (left) and after (right) pre-processing.

4 Results

As in [1], we applied change detection methods in order to spot the start and end dates of lockdowns which occurred in France in both 2020 and 2021. A simple example consists in finding the start and end dates of the first nationwide lockdown, which lasted from March 16, 2020 to May 11, 2020. Finally, we made a more complex analysis by trying to predict the different periods of curfews, from October 17, 2020 to June 20, 2021.

4.1 A simple example: nationwide lockdown

The first nationwide lockdown which took place in many countries at the beginning of 2020 led to a very strong reduction of urban noise throughout the world. In **Figure 2**, we show the results of change point detection after removing values between 7pm and 7am as there is no urban noise during these periods and smoothing the values per day using the median in order to discard spikes due to crowd movements. This shows how change point detection can be applied to seismic data.

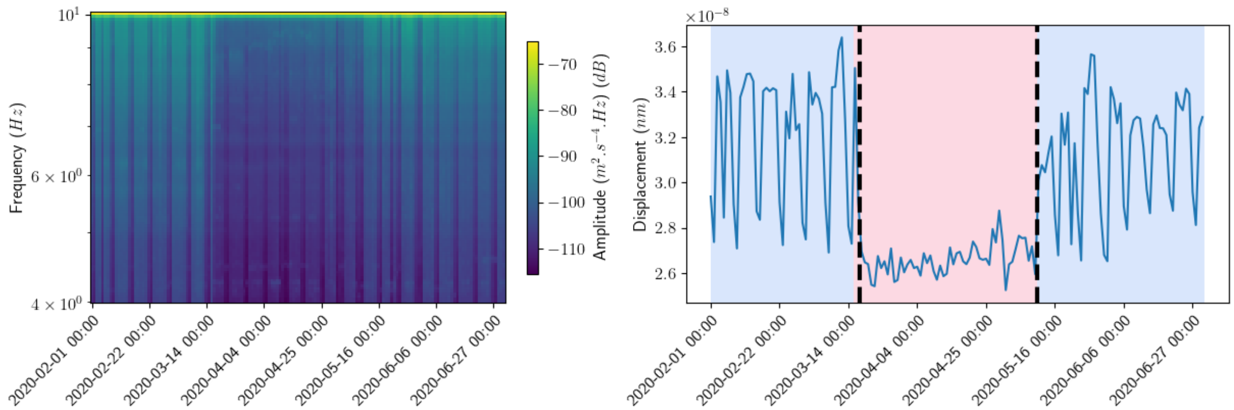


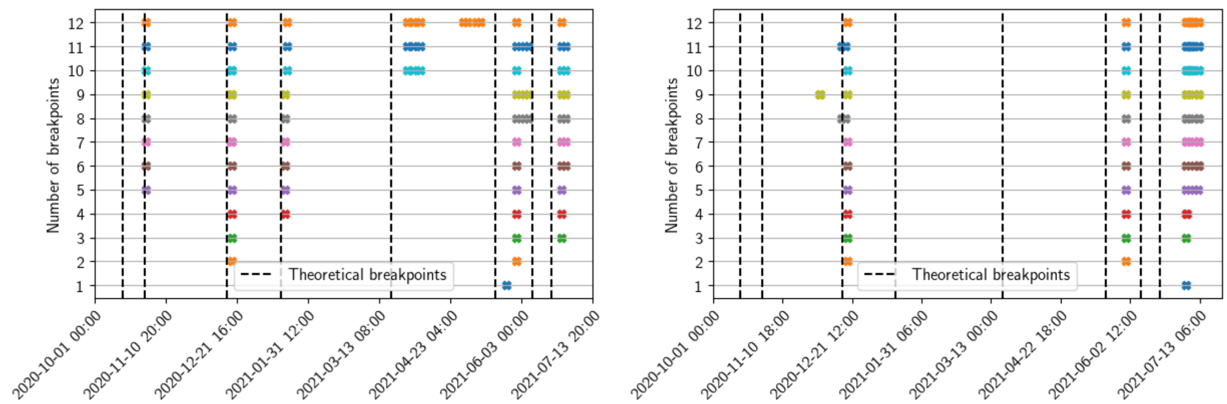
Figure 2: On the left, we observe the overall reduction of urban noise during the lockdown. On the right, dashed lines correspond to predicted breakpoints (Hausdorff score: 2 days).

4.2 Detailed analysis: curfews

In order to evaluate change point detection on more complex situations than the first nationwide lockdown in France, which appears quite clearly in the spectrogram, we applied `Dynp` and `Pelt` to predict the different changes in curfew hours in France[1]. For instance, the curfew started on October 17, 2020 before the second nationwide lockdown of October 30, 2020 and it was reinstated back in a stronger fashion on December 16, 2020. Later breakpoints correspond to changes in the curfew hours. We kept all samples and smoothed the values per hour using the median.

We can see in **Figure 3** how the chosen changepoints vary, and we can note the difference depending on the cost function chosen. The kernelized mean change function yields better results, while the linear cost does not detect quite a few actual breakpoints, and prefers to stick all change points close to one another, as small segments have lower costs.

[1] `https://fr.wikipedia.org/wiki/Couvre-feux_de_2020-2021_en_France` (in French)

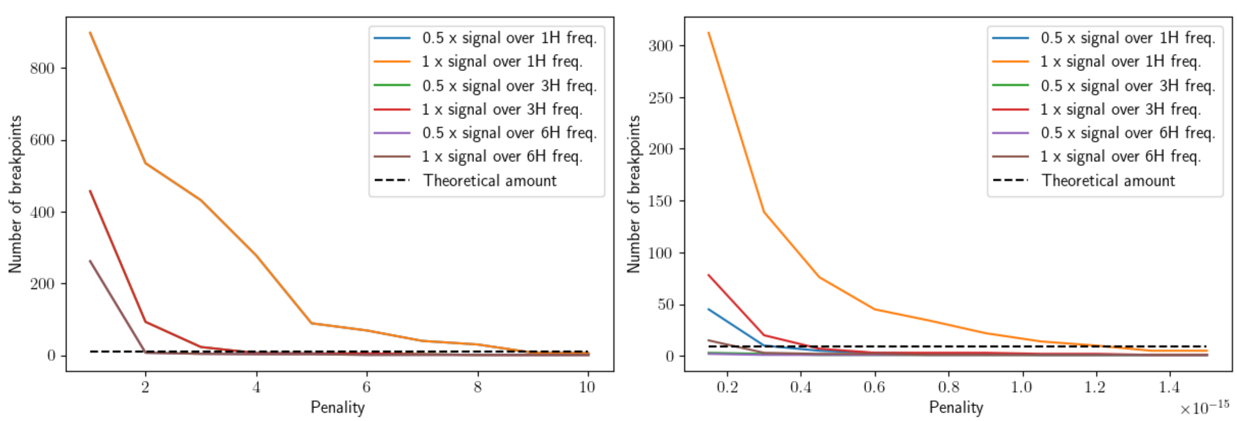


(a) With a kernelized mean change cost function

(b) With an L2 cost function

Figure 3: The different breakpoints selected depending on how many `Dynp` was asked to find.

As shown in **Figure 5** of the appendix, the repartition of detected changepoints when using PELT also depends on the cost function. To have an idea about how to determine the penalty, we test the algorithm on different representations of the same period. For that, we sample the signal on three different frequencies, and compute the number of detected points on those signals. We also see what happens if we apply the algorithm to the signals multiplied by a scalar. Results are shown in **Figure 4**. We can see that even this simple operation will disrupt the results. However, the kernelized mean change is slightly more robust: multiplying the signal by a scalar has no effect, since the distance is adapted to the data through the automated adaptation of the parameter γ. This suggests that the kernelized mean change is more robust than the others, and that a good penalty for one dataset is more likely to remain correct for other ones.

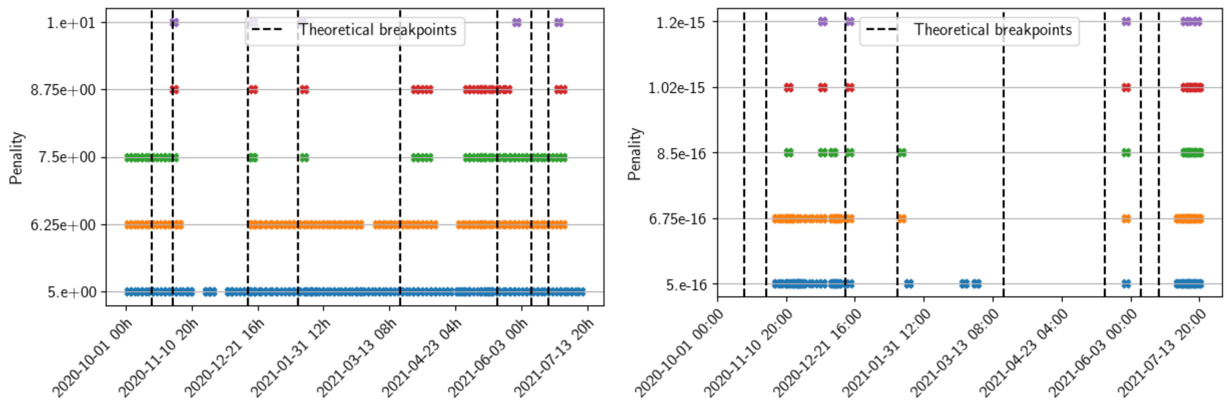


(a) With a kernelized mean change cost function

(b) With an L2 cost function

Figure 4: Amount of detected changepoints as functions of the penalty.

5 Appendix: additional figure



(a) With a kernelized mean change cost function

(b) With an L2 cost function

Figure 5: The different breakpoints selected depending on the penalty given to `Pelt`.

References

[1] Susanne Taina Ramalho Maciel, Marcelo Peres Rocha and Martin Schimmel. *Urban seismic monitoring in Brasilia, Brazil.* PLoS ONE, 16(8 August), 1–14.

[2] Thomas Lecocq, Frédérick Massin, Claudio Satriano, Mark Vanstone and Tobias Megies. *SeismoRMS - A simple python/jupyter notebook package for studying seismic noise changes.* 2020.

[3] Charles Truong, Laurent Oudre and Nicolas Vayatis. *Selective review of offline change point detection methods.* Signal Processing, 167:107299, 2020.

[4] R. Killick, P. Fearnhead and I. A. Eckley. *Optimal Detection of Changepoints With a Linear Computational Cost.* Journal of the American Statistical Association, vol. 107, no. 500, 2012, pp. 1590–1598.

[5] Moritz Beyreuther, Robert Barsch, Lion Krischer, Tobias Megies, Yannik Behr and Joachim Wassermann. *ObsPy: A Python Toolbox for Seismology*. June 2010.

[6] Daniel E. McNamara and Raymond P. Buland. *Ambient Noise Levels in the Continental United States*. Bulletin of the Seismological Society of America, 94 (4), 1517-1527. 2004.