Mini-Project (ML for Time Series) - MVA 2021/2022

Loïc MAGNE loic.magne@ens-paris-saclay.fr
Tanguy MAGNE tanguy.magne@minesparis.psl.eu

March 30, 2022

What is expected for these mini-projects? The goal of the exercise is to read (and understand) a research article, implement it (or find an implementation), test it on real data and comment on the results obtained Depending on the articles, the task will not always be the same: some articles are more theoretical or complex, others are in the direct line of the course, etc... It is therefore important to balance the exercise according to the article. For example, if you have reused an existing implementation, it is obvious that you will have to develop in a more detailed way the analysis of the results, the influence of the parameters etc... Do not hesitate to contact us by email if you wish to be guided.

The report The report must be at most FIVE pages and use this template (excluding references). If needed, additional images and tables can be put in Appendix, but must be discussed in the main document. The report must contain a precise description of the work done, a description of the method, and the results of your tests. Please do not include source code! The report must clearly show the elements that you have done yourself and those that you have reused only, as well as the distribution of tasks within the team (see detailed plan below.)

The source code In addition to this report, you will have to send us a Python notebook allowing to launch the code and to test it on data. For the data, you can find it on standard sites like Kaggle, or the site https://timeseriesclassification.com/ which contains a lot of signals!

The oral presentations They will last 10 minutes followed by 5 minutes of questions. The plan of the defense is the same as the one of the report: presentation of the work done, description of the method and analysis of the results.

Deadlines Two sessions will be available :

- **Session 1**
  - Deadline for report: March, 23th (23:59)
  - Oral presentations: March, 25th (morning + afternoon)
- **Session 2**
  - Deadline for report: March, 30th (23:59)
  - Oral presentations: March, 31th and April, 1st (morning + afternoon)

1 Introduction and contributions

In this mini-project, we are interested in the article *Offline detection of change-points in the mean for stationary graph signals* [de la Concha et al., 2020] that tackles the problem of finding ruptures in a stream of graph signals, which is a graph signal that evolves through time. The goal is thus to find change points in a multivariate time series, taking into account the graph structure of this time series, and therefore the links between its different components. Finding ruptures on a multivariate time series is a well-known problem, and many methods exist to tackle this problem. It is for instance addressed in [Arlot et al., 2019].

However, no change point detectors that take into account the graph structure of the problem have been developed. The goal of the article is therefore to detect ruptures in the mean of such a stream of graph signals, taking into account the structures of the graph. It does it in an offline way, which means that the complete stream is known when performing the detection. The method is based on the recently developed framework of graph signal processing. To address the problem, the stream of graph signal is transposed to the Fourier domain. It does not requires the number of change point in the signal to be known, and adopt a model selection approach to determine the number of change points.

We split the work in the following way. Loïc worked more on the algorithms themselves, understanding and implementing them. Tanguy on his side worked on the algorithm that allows approximating the power spectral density (PSD) of the stream of graph signals, which is required for the method. We both work on the applications, from different perspectives. The article came along with an implementation of the method. We partially took inspiration from this implementation, adapting it to fit the philosophy of the ruptures package [Truong et al., 2020], but re-coding everything, as the implementation was not well designed and not everything was coded. We repeated the synthetic scenario presented in the paper and also applied the method to detect change points on the Molene dataset [Girault, 2015], which is composed of temperature acquisition in Brittany in January 2014.

2 Method

Formally the problem is defined as follow: we have a graph $G = (E, V, S)$ and of stream of noisy graph signals (SGS) over this graph $Y = \{y_t\}_{t=0}^{T}$ where $\forall t, y_t \in \mathbb{R}^p$. At each timestep we associate the mean of the signal $\mu_t = \mathbb{E}(y_t) \in \mathbb{R}^p$. We want to find timestep intervals where μ_t remains constant, i.e. find $\tau = \{\tau_0 = 0, \tau_1, \dots, \tau_d = T\}$ where $\forall i \in \{1, d\}, \mu_{\tau_{i-1}+1} = \mu_{\tau_{i-1}+2} = \cdots = \mu_{\tau_i}$. Note that we are interested in recovering both the timesteps τ and the mean μ. The overall idea of the studied article [de la Concha et al., 2020] is to apply change-point detection methods in the graph frequency domain where samples are independent, instead of the time domain where samples are correlated depending on the graph structure.

The article develops a general theory and set of algorithms which make use of a Graph Shift Operator (GSO). As in practice the GSO used is always the graph Laplacian, we will describe the methods with the graph Laplacian which makes things more intuitive to understand.

The article proposes two algorithms to solve the change-point detection problem: the Lasso-based Graph Signal change-point detector (LGS), and the Variable Selection-based GS change-point detector (VSGS), the second one being basically a version of the first one which doesn't require parameters tuning thanks to variable selection. In the following, we describe both methods, and

explain hypothesis and analysis made in the article to make those algorithms work.

As mentioned before, the idea of both algorithms is to translate the problem to the graph frequency domain to make samples de-correlated. If y denotes a graph signal, $\tilde{y}$ is the graph Fourier transform defined by $\tilde{y} = U^T y$ where U are the eigenvectors of the Laplacian $L = U\Theta U^T$. The following hypotheses are made over the SGS:

- Graph signals are independent with respect to time
- Graph signals follow a multivariate Gaussian distribution with the same parameters in each segment
- If $t \in \{\tau_{l-1}+1, \dots, \tau_l\}$, $y_t - \mu_{\tau_l}$ is stationary with respect to the Laplacian L (check the article for a detailed definition of stationarity for the spatial domain)
- The mean of the graph signal admits a sparse representation in the frequency domain
- L has all its eigenvalues different and remains the same through time

Using those hypotheses and using maximum-likelihood principles and sparsity principles, the two algorithms are derived. The LGS method sets the problem as a typical cost-function based change-point detection problem, the objective function is:

$$\begin{aligned}\hat{d}, \hat{\tau}, \hat{\mu} &:= \arg\min_{d,\tau,\mu} C(\tau, \tilde{\mu}, \tilde{Y}) + pen(d) \\ &= \arg\min_{d,\tau,\mu} \sum_{l=1}^{d} \sum_{t=\tau_{l-1}+1}^{\tau_l} \left[\sum_{i=1}^{p} \frac{(\tilde{y_t}^{(i)} - \tilde{\mu_{\tau_l}}^{(i)})^2}{TP_y^{(i)}} + \lambda \frac{|\tilde{\mu_{\tau_l}}|_1}{T} \right] + \frac{d}{T}(c_1 + c_2 log \frac{d}{T})\end{aligned}$$

where λ, c_1, c_2 are constants that need to be set. Notice that the $|\mu|_1$ term encodes the sparsity hypothesis, while the $pen(d)$ term is here to tune automatically the number of change-point. While Table 1 shows that this method can get good results, it requires tuning three parameters. The VSGS algorithm is thus introduced to perform variable selection and not having parameters to tune. The optimized cost function is a variant of the LGS one:

$$\hat{d}, \hat{\tau}, \hat{\mu} = \arg\min_{d,\tau,\mu} \sum_{l=1}^{d} \sum_{t=\tau_{l-1}+1}^{\tau_l} \left[\sum_{i=1}^{p} \frac{(\tilde{y_t}^{(i)} - \tilde{\mu_{\tau_l}}^{(i)})^2}{TP_y^{(i)}} \right] + K_1 \frac{D_m}{T} + \frac{d}{T}(K_2 + K_3 log \frac{d}{T})$$

where K_1, K_2, K_3 are constants, and D_m is a quantity encoding the sparsity. Variables are selected in two ways: for λ, a grid of value is used which defines several levels of sparsity D_m, and the cost function problem is solved for multiple levels of sparsity. For K_1, K_2 and K_3, slope heuristic [Arlot, 2019] is used to find the best values. The article provides theoretical analysis which explains why both approaches are valid.

Results of both methods can be found in Table 1. As a baseline, we compared those graph methods with a traditional Linear Kernel method which doesn't make use of the graph structure.

As we have seen, both graph methods require the graph power spectral density (PSD) of the signal. In experimental cases, this quantity is unknown and we have to estimate it. To do so, many methods exist. Let us first recall the definition of the PSD: $\text{diag}(P_y) = U^T \Sigma_y U$ where P_y is

the PSD, Σ_y is the covariance of the graph signal and U the matrix that contains the eigenvectors of the GSO.

Using this, the first obvious method is just to estimate the covariance of the signal using the realization that we have, and then compute the PSD. However, this method doesn't scale well with the number of nodes as it requires computing the eigenvectors of the GSO. Moreover, it requires to have enough timesteps on which to estimate accurately the covariance matrix. Therefore a more sophisticated method was developed by [Perraudin and Vandergheynst, 2017, Section 4]. This method is based on the use of the windowed Fourier transform.

3 Data

For our experiments, we used both synthetic and real data. This allows us to get both theoretical results to understand the behavior of the method on data that fit perfectly the hypothesis, but also more practical results when we're not completely sure that the hypotheses are verified.

Synthetic data The synthetic data we used are the same as the one presented in the paper. Using synthetic data in the first time is really useful. Indeed, when generating synthetic data, we also generate the ground truth precisely. Therefore it is easy to assess the performance of the model on the data. Moreover, using synthetic data allows to control the way they are generated, and thus be sure that they satisfy the hypothesis required by the method. In our case, another interest in using such synthetic data is to be able to compare the results of our implementation with the ones presented in the article.

The data are generated to verify the hypothesis. Therefore, since we are using the Laplacian of the graph as the GSO, we want the graph to be connected, to have distinct eigenvalues. To respect the fact that the mean of the graph signals admits a sparse representation with respect to the basis defined by the eigenvectors, we construct the means as the inverse graphs Fourier Transform of a sparse vector. Finally, to ensure that all hypotheses concerning stationary are satisfied, the signal is generated as the output of a filter on white noise. This is indeed an equivalent definition of stationary to the one given in the article.

Different scenarios are created, all following the conditions mentioned above. They are based on different distributions for the white noise used, but also for the number of change points, and their relative distance. Also, different filters are used.

Real data Then, we applied the method to real data. For that, we choose to use the Molene dataset, which is composed of temperature acquisition in Brittany in January 2014 [Girault, 2015]. Because we know the real geographical position of the station that measured the temperatures, we can construct a graph based on the geodesic distance between the station. We only kept the station that had no missing values and constructed the graph using the distance matrix and exponential smoothing, with the minimum threshold that keeps the graph connected. This allows us to satisfy the last hypothesis, the Laplacian being normal with distinct eigenvalues.

About the first hypothesis, the fact that the graph signals must be independent with respect to the temporal domain, this seems not clear why it should be the case here because the temperature at a given time depends on the temperature at the previous time. The second hypothesis of the method which states that the graph signals follow a multivariate Gaussian distribution sharing the same expectation parameter if they belong to the same segment seems reasonable. Indeed, on

a short period of time and in the same location, external factors stay the same and the distribution remains unchanged. Other hypotheses will be discussed later, in the result section, once we have the segmentation.

4 Results

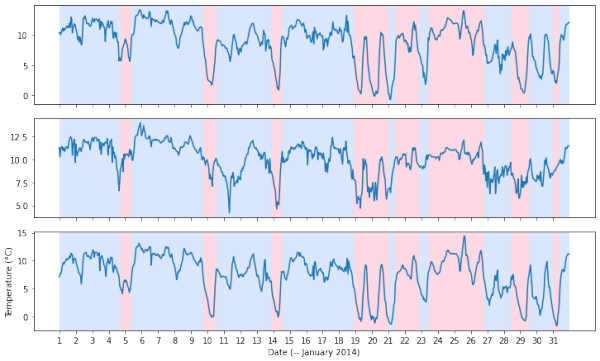
Dataset	Algorithm	PSD Method	Hausdorff ↓	Randindex ↑	Recall ↑	Precision ↑
ER	LGS	True	1.72 (0.53)	0.98 (0.01)	1.00 (0.00)	0.98 (0.02)
ER	LGS	Covariance	1.91 (0.54)	0.98 (0.01)	1.00 (0.00)	0.98 (0.05)
ER	LGS	Perraudin	2.00 (0.00)	0.98 (0.01)	1.00 (0.00)	1.00 (0.00)
ER	VSGS	True	**1.71 (0.52)**	**0.98 (0.01)**	**1.00 (0.00)**	**1.00 (0.00)**
ER	VSGS	Covariance	8.87 (15.2)	0.97 (0.02)	0.96 (0.09)	1.00 (0.00)
ER	VSGS	Perraudin	2.00 (0.00)	0.98 (0.01)	1.00 (0.00)	1.00 (0.00)
BA	LGS	True	**5.74 (11.6)**	**0.97 (0.04)**	**0.96 (0.10)**	**1.00 (0.00)**
BA	LGS	Covariance	8.52 (13.9)	0.96 (0.05)	0.93 (0.13)	1.00 (0.00)
BA	LGS	Perraudin	32.68 (3.18)	0.88 (0.01)	0.67 (0.03)	1.00 (0.00)
BA	VSGS	True	6.96 (13.2)	0.97 (0.04)	0.95 (0.12)	1.00 (0.03)
BA	VSGS	Covariance	10.95 (16.0)	0.95 (0.06)	0.91 (0.15)	1.00 (0.00)
BA	VSGS	Perraudin	37.63 (3.68)	0.87 (0.01)	0.67 (0.03)	1.00 (0.00)
ER	LinKernel	-	**0.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**
BA	LinKernel	-	**2.20 (8.84)**	**0.99 (0.03)**	**0.98 (0.08)**	**1.00 (0.00)**

Table 1: Performance evaluation of presented methods in various scenarios. The mean and standard deviation (in parenthesis) are computed over 100 generated instances of graphs with 500 nodes.

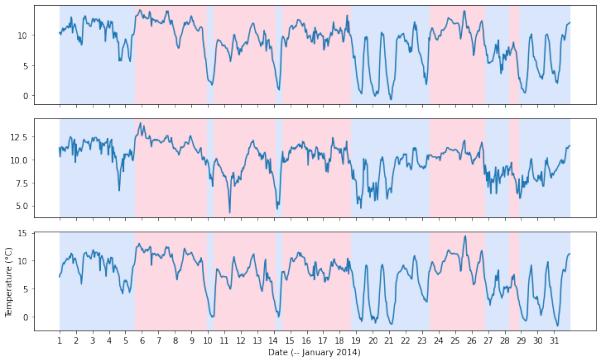
Synthetic data Results for synthetic data are shown in Table 1. Several data generation scenarios are tested (Erdos-Renyi (ER) and Barabasi-Albert (BA)), with each proposed algorithm, and various PSD estimation methods. Both algorithms give comparable results, but one has to keep in mind that VSGS doesn't need parameter tuning. Perraudin method is often worse than Covariance method to estimate the PSD. Although the results look good, it is important to remember that the data were generated to exactly satisfy hypotheses corresponding to the methods. Moreover, LGS/VSGS don't really over-perform the Linear Kernel method which doesn't use the graph structure, which makes the use of those graph methods questionable. The arxiv paper preprint actually has the same results (kernel methods better than graph methods) but authors didn't include those results in the final version of the paper. Overall our results are on par with the article result, slightly worse for the Erdos-Renyi case, slightly better for the Barabasi-Albert case.

Real data The article mention that both methods to approximate the PSD give similar results, therefore we tried both on the real dataset. Moreover, as the VSGS is simpler to use as it doesn't require tuning parameters, we use this algorithm, over the LGS algorithm. Results are presented in Figure 1.

We can notice several things. Firstly, we have to recall that on this dataset, a timestep last 1 hour in real life. Therefore every 24 points correspond to one day. Then we can see that up to the 18^{th} of January, both methods were able to detect two or three small segments. These small segments correspond to nights when the temperature dropped significantly. On the larger segments, the



(a) Using PSD estimated through estimation of the covariance

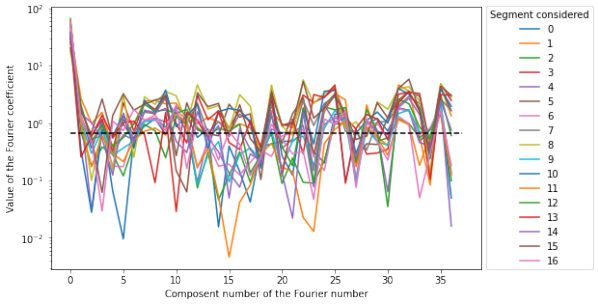


(b) Using PSD estimated through Perraudin method

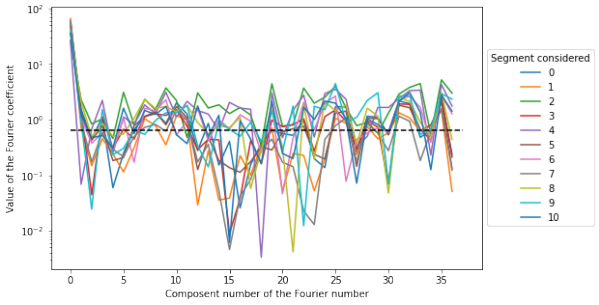
Figure 1: Results of the VSGS algorithm on the Molene dataset. The three lines correpond respectively to the station of Auray, Belle Ile-Le Talut and Bignan

temperature remains mostly constant, even during the nights. After the 18^{th} of January, things are a bit less clear, temperatures are varying much more. There still is another period detected by both methods where the temperatures are stable, between the 24^{th} and the 27^{th}. On the other parts, the mean is moving too much and it seems normal that the methods perform a bit poorly.

Finally, once we have this segmentation we can verify the 4^{th} hypothesis, that state that on each of the segment, the mean of the graph signal admits a sparse representation with respect to the basis defined by the eigenvectors of U, which means that its Fourier representation is sparse. We plotted the Fourier coefficients of each of the segments in Figure 2.



(a) Segments found with the PSD estimated through estimation of the covariance



(b) Segments found with the PSD estimated through Perraudin method

Figure 2: Fourier transform of the mean of each segments

In both cases, we can see that several coefficients are indeed really small, less 1% of the greatest one.

References

[Arlot, 2019] Arlot, S. (2019). Rejoinder on: Minimal penalties and the slope heuristics: a survey. *Journal de la Societe Française de Statistique*, 160(3):158–168.

[Arlot et al., 2019] Arlot, S., Celisse, A., and Harchaoui, Z. (2019). A kernel multiple change-point algorithm via model selection. *Journal of machine learning research*, 20(162).

[de la Concha et al., 2020] de la Concha, A., Vayatis, N., and Kalogeratos, A. (2020). Offline detection of change-points in the mean for stationary graph signals. *arXiv preprint arXiv:2006.10628*.

[Girault, 2015] Girault, B. (2015). Stationary graph signals using an isometric graph translation. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1516–1520.

[Perraudin and Vandergheynst, 2017] Perraudin, N. and Vandergheynst, P. (2017). Stationary signal processing on graphs. *IEEE Transactions on Signal Processing*, 65(13):3462–3477.

[Truong et al., 2020] Truong, C., Oudre, L., and Vayatis, N. (2020). Selective review of offline change point detection methods. *Signal Processing*, 167:107299.