Non-parametric Community Change-points Detection in Streaming Graph Signals

A. Ferrari, C. Richard

Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, France

Abstract—Detecting changes in network-structured time series data is of utmost importance in critical applications as diverse as detecting denial of service attacks against online service providers or monitoring energy and water supplies. The aim of this paper is to address this challenge when anomalies activate unknown groups of nodes in a network. We devise an online change-point detection algorithm that fully benefits from the recent advances in graph signal processing to exploit the characteristics of the data that lie on irregular supports. Built upon the kernel machinery, it performs density ratio estimation in an online way. The algorithm is scalable in the sense that it is spatially distributed over the nodes to monitor large-scale dynamic networks. The detection and localization performances of the algorithm are illustrated with simulated data.

Index Terms—Graph signal processing, streaming graph signals, non-parametric change-point detection, graph filtering.

I. Introduction

The problem of change-point detection is defined as finding a switching time instant from which the statistical properties of a signal change, namely, the observations are drawn from a distribution up to the point before it, and are drawn from another distribution after this point. Detecting anomalous events in network-structured time series data has become relevant in a variety of applications. Examples can be found in network security such as detection of denial of service attacks or cyber-intrusions in computer networks. Examples can also be found in medical monitoring, such as automatic detection of epileptic seizures in electroencephalogram signals, or detection of mental workload or drowsiness in brain computer interfaces. This problem is of great interest in a wide range of disciplines, such as physics, biology, and social sciences, where graphs provide a powerful machinery for effectively capturing correlations among inter-dependent time series data.

The anomalies in network-structured time series data often tend to activate communities of nodes in the graphs. The problem then consists of deciding, based on noisy measurements at each node of a graph, whether the underlying unknown graph signal is in a nominal state over the whole graph, or if there exist one or more clusters of nodes with anomalous activation. Over the last years, graph signal processing (GSP) has emerged as a new research field aiming at extending the well-developed tools for processing time-domain signals to signals defined over the nodes of a graph while exploiting the underlying connectivity information. Beyond its use for sampling [1], filtering [2] or modeling [3] signals on graphs, to cite a few key research directions, GSP was recently used

for change-points detection over graph signals [4], [5]. These works address critical issues raised in [6], [7]. Indeed, if the generalized likelihood ratio (GLR) test is a natural solution for detecting anomalous clusters of activity in graphs, it has high computational complexity because it requires scanning all well-connected clusters of nodes and testing them individually. In [4], the authors first address this issue by incorporating, into the detection problem, the properties of the graph topology through its spectrum. They analyze the test statistics, which is based on spectral patterns of the combinatorial Laplacian, and show how it is indeed related to the problem of finding graph cuts. Following the same line of reasoning, they derive another detector in [5], called Graph Fourier Scan Statistic (GFSS). Based on a relaxation of its initial form in [5], it consists of a low-pass filter built upon the graph Fourier transform. Because of its particular form, the performance of this detector can be precisely characterized. None of these works address the online non-parametric detection problem from streaming graph signals, nor do they implement the detectors in a distributed manner.

The aim of this paper is to propose a flexible strategy built upon the (reproducing) kernel machinery to perform density ratio estimation in an online and distributed manner across nodes, while using GSP framework to enhance the contrast produced by possible anomalous activities over unknown groups of nodes and then make their detection easier.

II. Problem Formulation

We consider an undirected graph $\mathcal{G} = \{V, E, \boldsymbol{W}\}$ with p vertices $V = \{1, \ldots, p\}$, m edges $(i, j) \in E \subset V \times V$, endowed with a $p \times p$ weighted adjacency matrix $\boldsymbol{W}$ where $\boldsymbol{W}_{i,j} = \boldsymbol{W}_{j,i} \geq 0$ is the connection strength of $(i, j) \in E$.

We denote by $\boldsymbol{y}_t(i)$ the p_i-dimensional signal sample observed at time instant t and vertex i. We shall assume that the observations $\boldsymbol{y}_t(i)$ are independent w.r.t. t and i. In this paper, we consider that a change in the distribution of $\boldsymbol{y}_t = \big(\boldsymbol{y}_t(1) \ldots \boldsymbol{y}_t(p)\big)$ may occur in an unknown cluster $\mathcal{C}^*$ at an unknown time instant t_r, that is,

$$t < t_r : \boldsymbol{y}_t \sim \prod_{i=1}^{p} f_i\big(\boldsymbol{y}_t(i)\big), \ t \geq t_r : \boldsymbol{y}_t \sim \prod_{i=1}^{p} f_i^+\big(\boldsymbol{y}_t(i)\big), \quad (1)$$

where $\forall i \in \mathcal{C}^* : f_i^+ = f_i^*$, and $\forall i \notin \mathcal{C}^* : f_i^+ = f_i$.

The objective is to derive an online algorithm to process the streaming data $\boldsymbol{y}_t$ in order to detect a change-point as defined in (1), where $\mathcal{C}^*$, t_r and all f_i and f_i^* are unknown.

A solution to this problem was recently proposed in [8] for *scalar measurements* ($p_i = 1$) and model $\boldsymbol{y}_t(i) = \boldsymbol{m}_t(i) + \boldsymbol{e}_t(i)$ where $\boldsymbol{e}_t(i)$ is a scalar i.i.d zero-mean Gaussian noise. The changes under consideration occur on the mean vector $\boldsymbol{m}_t$ as:

$$t < t_r : \ \boldsymbol{m}_t = \boldsymbol{m}, \qquad t \geq t_r : \ \boldsymbol{m}_t = \boldsymbol{m} + \boldsymbol{\delta}, \quad (2)$$

where $\boldsymbol{\delta}$ is a non-zero vector only on $\mathcal{C}^*$ which does not vary within this cluster. Built upon GSP to extract information about the clusters, the core detection functionality implemented at each node in [8] is ensured by the adaptive strategy proposed in [9]. Two linear models are learned simultaneously using two learning rates, on the one hand to estimate the baseline trend, and on the other hand to adapt to abrupt changes. The resulting test statistics, which consists of the distance between these two models, only applies to specific scenarios such as (2).

Here we address problem (1) with a flexible non-parametric strategy in the case where a) the measurements at each node are multivariate and b) all distributions f_i and f_i^* are unknown.

III. GFSS APPROACH TO CLUSTERED DETECTION

We first recall the principles of GFSS detection introduced in [5], with a focus on the filtering process to perform spectral clustering. Next, we introduce the likelihood ratio used to address problem (1). None of these ingredients address the online detection problem from streaming graph signals, nor do they implement the detectors in a distributed manner.

A. GFSS spectral clustering

Work [5] focuses on a static graph signal consisting of a scalar measurement at each node as in (2) with $\boldsymbol{m} = \boldsymbol{0}$. Let $\boldsymbol{L}$ be the graph Laplacian of $\mathcal{G}$ and let us define by $\{\boldsymbol{u}_i\}_{i=1,\dots,p}$ the set of orthonormal eigenvectors of $\boldsymbol{L}$ with $\{\mu_i\}_{i=1,\dots,p}$ the corresponding eigenvalues. Given $\boldsymbol{y}$, the GFSS is defined as:

$$t_{\text{GFSS}}(\boldsymbol{y}) = \|\boldsymbol{g}_{\boldsymbol{y}}\|_2, \quad (3)$$

where $\boldsymbol{g}_{\boldsymbol{y}}$ is the graph-filtered signal:

$$\boldsymbol{g}_{\boldsymbol{y}} = \sum_{i=2}^{p} h^*(\mu_i)(\boldsymbol{u}_i^\top \boldsymbol{y}) \boldsymbol{u}_i, \quad (4)$$

with $h^*(\mu)$ the frequency response of the filter defined in [5]:

$$h^*(\mu) = \min\left\{1, \sqrt{\frac{\gamma}{\mu}}\right\}, \ \mu > 0, \quad (5)$$

and $\gamma > 0$ a tuning parameter; see [2], [10], [11] for details on graph filtering.

To get more insight into statistics (3), it is important to recall the role played by the eigenvectors $\boldsymbol{u}_i$ of the graph Laplacian matrix $\boldsymbol{L}$ in spectral clustering. Consider first the *ideal case* of a graph with $k > 1$ connected components or communities, and denote by $\mathcal{C}_i$ with $i = 1, \dots, k$ the set of vertices in component i. Each eigenvector $\boldsymbol{u}_i$ of the Laplacian matrix is proportional (with the factor $1/\sqrt{|\mathcal{C}_i|}$) to the indicator function of the ith connected component $\mathcal{C}_i$, and $\boldsymbol{u}_i^\top \boldsymbol{y}$ is proportional to the sum of $\boldsymbol{y}$ on $\mathcal{C}_i$. Consequently, $(\boldsymbol{u}_i^\top \boldsymbol{y})\boldsymbol{u}_i$

in (4) assigns to each vertex n of $\mathcal{C}_i$, the mean value of the signal samples $\boldsymbol{y}(n)$ at all vertices n within $\mathcal{C}_i$. Finally, as the number of components $\mathcal{C}_i$ is unknown, the frequency response in (5) penalizes large numbers of components in (4). When the k components $\mathcal{C}_i$ are connected by few edges, we can assume that this analysis which relies on the properties of the eigenvectors $\boldsymbol{u}_i$ of the Laplacian matrix is still approximately valid. Note that this kind of approximation is the starting point of spectral clustering methods; see, e.g., [12], [13], [14].

B. Likelihood ratio test

Consider problem (1), and assume that all probability densities are known. We denote by $\boldsymbol{\ell}_t$ the p-dimensional vector containing the log-likelihood ratios of the measurement $\boldsymbol{y}_t(k)$:

$$\boldsymbol{\ell}_t(k) = \log\left(\frac{f_k^+(\boldsymbol{y}_t(k))}{f_k(\boldsymbol{y}_t(k))}\right). \quad (6)$$

The likelihood ratio test is a natural solution for addressing a change-point detection problem such as (1). Nevertheless it suffers from its computational complexity since it consists of scanning all possible well-connected clusters of nodes $\mathcal{C}_k$ and testing them individually.

Substituting $\boldsymbol{y}$ by $\boldsymbol{\ell}_t$ in (4), and considering the resulting graph signal $\boldsymbol{g}_{\boldsymbol{\ell}_t}$, it can be expected that the entries of $\boldsymbol{g}_{\boldsymbol{\ell}_t}$ will be approximately equal to:

$$\boldsymbol{g}_{\boldsymbol{\ell}_t}(k) \approx \frac{1}{|\mathcal{C}_i|} \log\left(\frac{\prod_{n \in \mathcal{C}_i} f_n^+(\boldsymbol{y}_t(n))}{\prod_{n \in \mathcal{C}_i} f_n(\boldsymbol{y}_t(n))}\right), \quad (7)$$

for all node k in $\mathcal{C}_i$. This means that, by exploiting the graph topology, $\boldsymbol{g}_{\boldsymbol{\ell}_t}$ assigns to each node of $\mathcal{G}$ the log-likelihood ratio of the cluster to which it belongs. The clustered structure of the graph is promoted in the log-likelihood ratio $\boldsymbol{g}_{\boldsymbol{\ell}_t}$ by the graph filter (5), which makes its use attractive to address the change-point detection problem (1).

IV. DETECTION ALGORITHM

The approach outlined in Section III relies on the knowledge of the likelihood ratio at each node n and time instant t. We shall now devise a non-parametric detection scheme based on an estimate $\hat{\boldsymbol{\ell}}_t$ of $\boldsymbol{\ell}_t$. Computing $\boldsymbol{g}_{\boldsymbol{\ell}_t}$ from $\boldsymbol{\ell}_t$ using (4) needs to be carried out in a centralized manner. In order to make the detection algorithm scalable, we shall investigate a distributed implementation across the nodes of the network.

A. Online likelihood ratio estimation

Non-parametric change-point detection algorithms were introduced to handle scenarios where no prior information on the data distribution and the nature of the change is available. They are usually classified into two categories, supervised and unsupervised methods, depending if training data are available. This paper focuses on unsupervised algorithms as their flexibility allows them to handle many practical situations. Some of them are based on subspace or manifold tracking methods [9], [15], [16], and aim to detect abrupt changes in the sequence of residuals. Their parametric form makes them pre-designed for tracking specific statistics. Others are model-free as they do not

rely on particular parametric assumption [17]. They however tend to be less accurate in high-dimensional problems because of the curse of dimensionality. To overcome this difficulty, recent contributions have focused on estimating the ratio of probability densities directly without going through density estimation [18], [19], [20]. In this paper, we are interested in an algorithm, denoted as NOUGAT (Nonparametric Online chanGepoint detection AlgoriThm) which implements this strategy [21], [22]. It estimates the density ratio on two consecutive intervals of the time series: a set of recent samples, say, the test set, and the set of samples that came before, say, the reference set. Unlike the algorithm considered in [8], it naturally manipulates multivariate observations at each node.

NOUGAT aims to detect changes in the distribution of $\boldsymbol{y}_t(k)$ at each vertex k by estimating a model for $\boldsymbol{r}_t(k) - 1$, where $\boldsymbol{r}_t(k) = f_k^+(\boldsymbol{y}_t(k))/f_k(\boldsymbol{y}_t(k))$ is the density ratio (with some abuse of notation) between the probability densities $f_k^+(\boldsymbol{y})$ and $f_k(\boldsymbol{y})$ of the data on a test and reference interval, respectively:

$$\left(\boldsymbol{y}_{t-(N_{\text{test}}-1)}(k), \ldots, \boldsymbol{y}_t(k)\right) \in \mathbb{R}^{p_k \times N_{\text{test}}}, \quad (8)$$

$$\left(\boldsymbol{y}_{t-(N_{\text{ref}}+N_{\text{test}}-1)}(k), \ldots, \boldsymbol{y}_{t-N_{\text{test}}}(k)\right) \in \mathbb{R}^{p_k \times N_{\text{ref}}}, \quad (9)$$

with N_{test} and N_{ref} the number of samples in the test and reference intervals, respectively. It is shown in [22] how $\boldsymbol{r}_t(k) - 1$ can be estimated, in a reproducing kernel Hilbert space $\mathcal{K}$ endowed with kernel $\kappa(\cdot, \cdot)$, by a kernel model. Note that $\boldsymbol{r}_t(k) - 1$ is preferred to $\boldsymbol{r}_t(k)$ because it leads to an unbiased estimator under the no change-point hypothesis. Using this model, we arrive at the estimation of $\boldsymbol{\ell}_t(k)$ in (6):

$$\hat{\boldsymbol{\ell}}_t(k) = \log\left(\boldsymbol{\theta}_{t-1}(k)^\top \boldsymbol{\kappa}_{\boldsymbol{\omega}}^k(\boldsymbol{y}_t(k)) + 1\right), \quad (10)$$

$$\boldsymbol{\kappa}_{\boldsymbol{\omega}}^k(\cdot) = \left(\kappa_{\omega_1}^k(\cdot), \ldots, \kappa_{\omega_L}^k(\cdot)\right)^\top, \quad (11)$$

where $\kappa_{\omega_q}^k(\cdot) = \kappa(\cdot, \boldsymbol{y}_{\omega_q}(k))$, with $q \in \{1, \ldots, L\}$, are the elements of a dictionary of size L which ensures the executability of the algorithm. This dictionary can be designed in an online or offline way using a standard strategy as in [23]. For the sake of clarity, we shall assume in the sequel that the dictionary has been designed beforehand.

Similarly to the derivation of the KLMS in [23], the parameter vector $\boldsymbol{\theta}_t(k)$ in (10) can be estimated recursively at each vertex k by a stochastic gradient descent algorithm:

$$\boldsymbol{\theta}_t(k) = \boldsymbol{\theta}_{t-1}(k) - \mu\left((\boldsymbol{H}_t(k) + \nu\boldsymbol{I})\boldsymbol{\theta}_{t-1}(k) + \boldsymbol{e}_t(k)\right), \quad (12)$$

where μ is the adaptation step and ν is a ridge regularization parameter. Quantities $\boldsymbol{H}_t(k)$ and $\boldsymbol{e}_t(k)$ are defined as:

$$\boldsymbol{e}_t(k) = \boldsymbol{h}_t^{\text{ref}}(k) - \boldsymbol{h}_t^{\text{test}}(k), \quad (13)$$

$$\boldsymbol{h}_t^{\text{test}}(k) = \frac{1}{N_{\text{test}}} \sum_{i=t-(N_{\text{test}}-1)}^{t} \boldsymbol{\kappa}_{\boldsymbol{\omega}}^k(\boldsymbol{y}_i(k)), \quad (14)$$

$$\boldsymbol{h}_t^{\text{ref}}(k) = \frac{1}{N_{\text{ref}}} \sum_{i=t-(N_{\text{test}}+N_{\text{ref}}-1)}^{t-N_{\text{test}}} \boldsymbol{\kappa}_{\boldsymbol{\omega}}^k(\boldsymbol{y}_i(k)), \quad (15)$$

$$\boldsymbol{H}_t(k) = \frac{1}{N_{\text{ref}}} \sum_{i=t-(N_{\text{test}}+N_{\text{ref}}-1)}^{t-N_{\text{test}}} \boldsymbol{\kappa}_{\boldsymbol{\omega}}^k(\boldsymbol{y}_i(k))\boldsymbol{\kappa}_{\boldsymbol{\omega}}^{k\top}(\boldsymbol{y}_i(k)). \quad (16)$$

B. Distributed implementation

The test statistics $\boldsymbol{\ell}_t$ defined in (6), and estimated in (10), does not take into account the graph topology. The aim of Section III was to show how graph-filtering $\boldsymbol{\ell}_t$ to $\boldsymbol{g}_{\boldsymbol{\ell}_t}$ improves the localisation of the cluster containing the abrupt change. The filtering operation defined in (4) requires centralized computational resources and hence cannot be scaled to large networks. Distributed implementations of graph filters have received much attention in the literature [24], [25], [26], [27], [28]. They are based on the approximation of the filtering operator by a polynomial in $\boldsymbol{L}$ for which, similarly to a FIR filter, the computation of the n-th entry of $\boldsymbol{L}\boldsymbol{\ell}_t$ depends on $\boldsymbol{\ell}_t(n)$ and on $\boldsymbol{\ell}_t(m)$ where $m \sim n$. Recently, [27], [28] proposed novel filter approximation strategies where, as in an IIR filter, each node recursively aggregates the previous outputs of its neighbors, allowing for the computation of a larger family of frequency responses with a limited memory footprint.

These filters iterate on a static graph signal. In the context of online change point detection, we propose to apply the ARMA_K filter defined in [27] to the streaming signal $\hat{\boldsymbol{\ell}}_t$, $t \geq 0$, defined in (10):

$$\boldsymbol{x}_{q,t} = \psi_q \boldsymbol{L}\boldsymbol{x}_{q,t-1} + \varphi_\ell \hat{\boldsymbol{\ell}}_t, \ \boldsymbol{x}_{q,-1} = \boldsymbol{0}, \ \forall q = 1 \ldots K, \quad (17)$$

$$\hat{\boldsymbol{g}}_{\hat{\boldsymbol{\ell}}_t} = \sum_{q=1}^{K} \boldsymbol{x}_{q,t} + c\hat{\boldsymbol{\ell}}_t. \quad (18)$$

If we assume that: $i)$. there is no change point, i.e. $\boldsymbol{\ell}_t = \boldsymbol{\ell}$, $ii)$ $\hat{\boldsymbol{\ell}}_t$ is an unbiased estimate of $\boldsymbol{\ell}_t$, see [22] and $iii)$ $\max_\ell\{|\psi_\ell|\}\rho(\boldsymbol{L}) < 1$ with $\rho(\cdot)$ the spectral radius of its matrix argument, it can be shown that $\lim_{t\to\infty} \mathsf{E}\{\hat{\boldsymbol{g}}_{\hat{\boldsymbol{\ell}}_t}\} = \hat{\boldsymbol{g}}_{\boldsymbol{\ell}}$ where $\hat{\boldsymbol{g}}_{\boldsymbol{\ell}}$ is $\boldsymbol{\ell}$ graph filtered by

$$h(\mu) = c + \sum_{q=1}^{K} \frac{\varphi_q}{1 - \psi_q\mu}. \quad (19)$$

The proof is omitted in this paper due to lack of space.

This suggests to estimate $\boldsymbol{g}_{\boldsymbol{\ell}_t}$ by $\hat{\boldsymbol{g}}_{\hat{\boldsymbol{\ell}}_t}$ where c and $\{(\phi_q, \psi_q)\}_{q=1,\ldots,L}$ are compute in order that $h(\mu)$ approximates $h^*(\mu)$ in (5). The corresponding distributed change point detection algorithm is described in Alg. 1.

Algorithm 1: Distributed graph change point detection

1: **for** $t = 1, 2, \ldots$ **do**
2:     // Each node updates separately its likelihood ratio
3:     **for** $n = 1, \ldots, k$ **do**
4:         Update $\boldsymbol{\theta}_t(n)$ using (12)–(16)
5:     **end for**
6:     // One iteration of distributed ARMA_K graph-filter
7:     Compute $\hat{\boldsymbol{g}}_{\hat{\boldsymbol{\ell}}_t}$ using (17)–(18)
8:     // Test each node
9:     **if** $\exists n \in V: |\hat{\boldsymbol{g}}_{\hat{\boldsymbol{\ell}}_t}(n)| > \xi$ **then**
10:        flag t as a change point on the graph at vertex n
11:    **end if**
12: **end for**

(a) Graph topology.　(b) Test statistic $|\hat{\ell}_t(n)|$ for $t = t^*$.　(c) Test statistic $|\hat{\boldsymbol{g}}_{\hat{\ell}_t}(n)|$ for $t = t^*$.
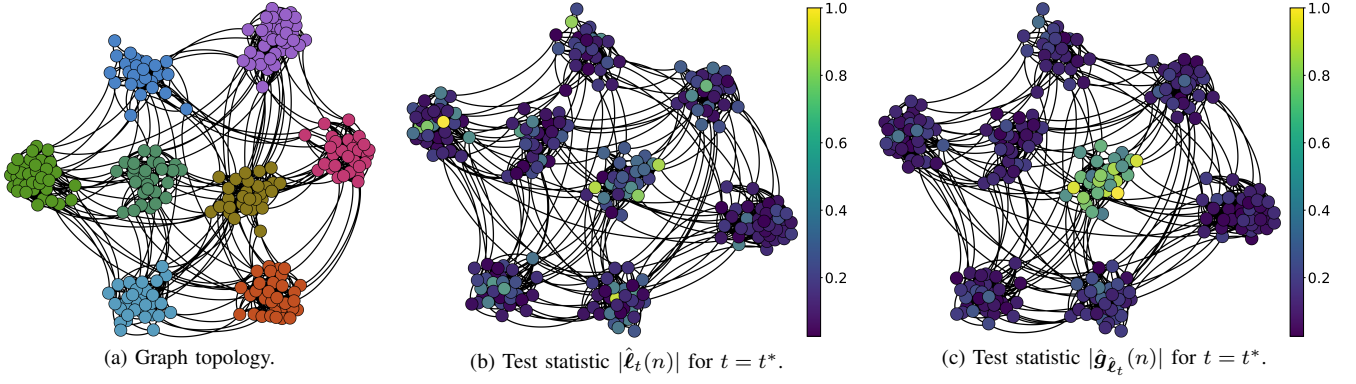
Fig. 1: Graph topology with coloured clusters C_i and test statistics over the graph.

V. EXPERIMENTS

The graph $\mathcal{G}$ used for simulations is provided in Fig. 1a. It is unweighted and contains $p = 250$ vertices and $m = 2508$ edges. In order to generate $\boldsymbol{y}_t$ as in (1) with a change on $\mathcal{C}^*$, the communities $\mathcal{C}_i$ of $\mathcal{G}$ were unfolded using [29]. Cluster $\mathcal{C}^*$ is colored in green khaki color at the center right of the graph.

The change-point (1) was set to $t_r = 1400$ and defined as:

$$\forall i \in \mathcal{C}^* : f_i^+ \equiv \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_i^+), \text{ and } \forall i \notin \mathcal{C}^* : f_i \equiv \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}),$$

with:

$$\boldsymbol{C}_i^+ = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}. \tag{20}$$

The parameters of NOUGAT were set to $N_{\text{ref}} = N_{\text{test}} = 128$, $\mu = 0.01$ and $\lambda = 0.01$. Note that given these windows lengths, see (8)–(9), NOUGAT detection is expected at $t^* = t_r + 128$. A Gaussian kernel κ with bandwidth $\sqrt{0.3}$ was used. The kernel dictionary size was set to $L = 80$ with 40 elements sampled from f_i and 40 elements sampled from f_i^+. Parameter γ of the GFSS filter (5) was set to $\gamma = 0.3$, and K in ARMA$_K$ approximation to $K = 4$ [8].

Figure 2 evaluates the detection performance of Algorithm 1 by comparing $\|\hat{\ell}_t\|_2^2$ and $\|\hat{\boldsymbol{g}}_{\hat{\ell}_t}\|_2^2$, for illustration purposes only since these two statistics are global. As expected, $\|\hat{\boldsymbol{g}}_{\hat{\ell}_t}\|_2^2$ clearly benefits from the graph topology through GFSS filter. In order to show that this improvement comes from the ability of $\hat{\boldsymbol{g}}_{\hat{\ell}_t}$ to better localize the nodes where a change occurs, Figs. 1b and 1c represents the two test statistics on $\mathcal{G}$ at time $t = t^*$. To confirm this result Monte Carlo simulation were performed to estimate ROC curves in both cases. Probabilities of detection and false alarm are defined (e.g. for $|\hat{\ell}_{t^*}(n)|$) as:

$$\text{P}_\text{d} = \text{Prob}\left(|\hat{\ell}_{t^*}(k)| > \xi \,|\, k \in \mathcal{C}^*\right), \tag{21}$$

$$\text{P}_\text{fa} = \text{Prob}\left(|\hat{\ell}_{t^*}(k)| > \xi \,|\, k \notin \mathcal{C}^*\right). \tag{22}$$

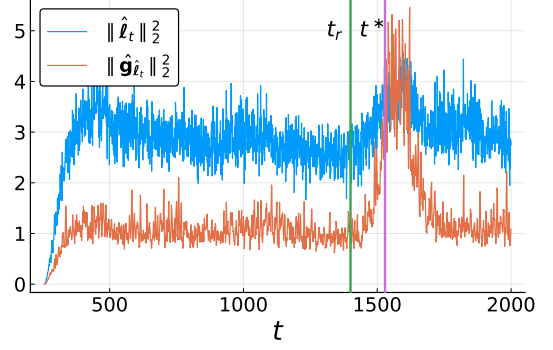Figure 3 clearly shows the benefit of using graph filtering in the proposed algorithm.

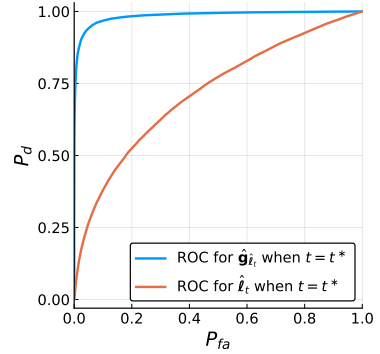Fig. 2: Global change point detection statistics over $\mathcal{G}$.

Fig. 3: ROC curves for $\hat{\ell}_{t^*}$ and $\hat{\boldsymbol{g}}_{\hat{\ell}_{t^*}}$

VI. CONCLUSION

This paper introduces a novel algorithm to detect a change point on a streaming graph signal which is located on an unknown cluster of the graph. The algorithm is fully non-parametric: it does not require any model of the change point and applies to multivariate measurements on the nodes of the graph. It also achieves scalability to large-scale dynamic networks through: *i)* a fully distributed implementation across the nodes, *ii)* an online processing of the measurements. The detection and localization performances of the algorithm are illustrated using simulated data.

REFERENCES

[1] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3775–3789, 2016.

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[3] R. Nassif, C. Richard, J. Chen, and A. H. Sayed, "Distributed diffusion adaptation over graph signals," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE ICASSP)*, 2018, pp. 4129–4133.

[4] J. Sharpnack, A. Singh, and A. Rinaldo, "Changepoint detection over graphs with the spectral scan statistic," in *Proc. International Conference on Artificial Intelligence and Statistics*, vol. 31, 2013, pp. 545–553.

[5] J. Sharpnack, A. Rinaldo, and A. Singh, "Detecting anomalous activity on networks with the graph Fourier Scan Statistic," *IEEE Transactions on Signal Processing*, vol. 64, no. 2, pp. 364–379, 2016.

[6] L. Addario-Berry, N. Broutin, L. Devroye, and G. Lugosi, "On combinatorial testing problems," *The Annals of Statistics*, vol. 38, no. 5, pp. 3063–3092, 2010.

[7] E. Arias-Castro, E. J. Candès, and A. Durand, "Detection of an anomalous cluster in a network," *The Annals of Statistics*, vol. 39, no. 1, pp. 278–304, 2011.

[8] A. Ferrari, C. Richard, and L. Verduci, "Distributed change detection in streaming graph signals," in *Proc. IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2019.

[9] N. Keriven, D. Garreau, and I. Poli, "NEWMA: a new method for scalable model-free online change-point detection," *arXiv.org*, p. arXiv:1805.08061, 2018.

[10] N. Tremblay, P. Gonçalves, and P. Borgnat, "Design of graph filters and filterbanks," in *Cooperative and Graph Signal Processing*. Academic Press, 2018, pp. 299–324.

[11] A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[12] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[13] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*. University of California, Berkeley, Berkeley, United States, 2002.

[14] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[15] H. Chen, "Sequential change-point detection based on nearest neighbors," *The Annals of Statistics*, vol. 47, no. 3, pp. 1381–1407, Jun. 2019.

[16] Y. Xie, J. Huang, and R. Willett, "Change-point detection for high-dimensional time series with missing data," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 12–27, 2013.

[17] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *Proc. International Conference on Very Large Data Bases*, 2004, pp. 180–191.

[18] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density Ratio Estimation in Machine Learning*, 1st ed. New York, NY, USA: Cambridge University Press, 2012.

[19] S. Aminikhanghahi, T. Wang, and D. J. Cook, "Real-time change point detection with application to smart home time series data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 1010–1023, 2018.

[20] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Neural Networks*, vol. 43, pp. 72 – 83, 2013.

[21] I. Bouchikhi, A. Ferrari, C. Richard, A. Bourrier, and M. Bernot, "Non-paramametric online change-point detection with kernel lms by relative density ratio estimation," in *Proc. IEEE Statistical Signal Processing Workshop (SSP)*, 2018.

[22] ——, "Kernel based online change point detection," in *Proc. European Conference on Signal Processing (EUSIPCO)*, 2019.

[23] C. Richard, J.-C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, 2009.

[24] D. I. Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev polynomial approximation for distributed signal processing," in *Proc. IEEE International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, 2011.

[25] A. Sandryhaila, S. Kar, and J. M. F. Moura, "Finite-time distributed consensus through graph filters," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[26] S. Segarra, A. G. Marques, and A. Ribeiro, "Distributed implementation of linear network operators using graph filters," in *Proc. Annual Allerton Conference on Communication, Control, and Computing*, 2015.

[27] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 274–288, 2017.

[28] M. A. Coutino Minguez, E. Isufi, and G. Leus, "Advances in distributed graph filtering," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2320–2333, 2019.

[29] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.