Online Non-parametric Change-point Detection via Graph-based Likelihood-ratio Estimation

Alejandro de la Concha Nicolas Vayatis Argyris Kalogeratos
Centre Borelli, ENS Paris-Saclay, 91190 Gif-sur-Yvette, France

***Abstract*—Consider each node of a graph to be generating a data stream that is synchronized and observed at near real-time. At a change-point τ, a change occurs for a subset of nodes C, which affects the probability distribution of their associated node streams. In this paper, we propose the Online Centralized Kernel- and Graph-based (OCKG) detection method to both detect τ and localize C, based on the direct estimation of the likelihood-ratio between the post-change and the pre-change distributions of the node streams. Our main working hypothesis is the smoothness of the likelihood-ratio estimates over the graph, i.e. connected nodes are expected to have similar likelihood-ratios. The proposed method is evaluated in synthetic experiments.**

***Index Terms*—Online change-point detection, non-parametric statistics, likelihood-ratio estimation, graph-signal processing.**

I. Introduction

Online, or sequential, change-point detection methods assume that a stream of data is observed in near real-time, and aim to detect the moment of a change τ as soon as possible, while minimizing the false alarm rate [1]–[3]. Modern challenges include handling larger amounts of complex data streams, e.g. data lying over a graph, or even graph streams. Many real-world systems can be seen as a network in which each node generates a stream of data: e.g. a network of seismic stations monitoring different geological events, the content shared by users of a social network, or a network of financial institutions, etc. A change-point may signify an earthquake, a shift of users' interest, or an early sign of an economic crisis. In these examples, the graph structure provides a priori relevant information about how the streams relate with each other, and perhaps shape their behavior after a change occures.

In this paper, we address a naturally arising question: how can we exploit the graph information in the online change-point detection task? As a response, we present the *Online Centralized Kernel- and Graph-based* (OCKG) detection method that is build over the collaborative likelihood-ratio estimation framework introduced in [4], under the intuitive assumption that the likelihood-ratios of any two connected nodes are expected to have a similar behavior. More precisely, we rely on the collaborative likelihood-ratio estimation framework introduced in [4]. OCKG has the notable advantages that it is: i) it is non-parametric and hence requiring minimum hypotheses about the nature of the data generating process at each graph node, ii) more sensitive, thanks to the integration of the graph structure, compared to methods that aggregate all data streams in a single stream, and iii) more accurate in localizing the affected nodes compared with similar methods (e.g. [5]). A longer version of this work can be found in [6].

☐ The authors acknowledge the support of the Industrial Data Analytics and Machine Learning Chair hosted at ENS Paris-Saclay. Correspondence to: {alejandro.de_la_concha_duarte, argyris.kalogeratos}@ens-paris-saclay.fr.

II. Background and problem statement

General notations. Let a_i be the i-th entry of a vector a; when the vector is itself indexed by an index j, then we refer to its i-th entry by $a_{j,i}$. A_{ij} is the entry at the i-th row and j-th column of a matrix A. $e_{\max}(A)$ denotes the maximum eigenvalue of A. $\mathbf{1}_M$ represents the vector with M ones (resp. $\mathbf{0}_M$), and $\mathbb{I}_M$ is the $M \times M$ identity matrix. We denote by $G = (V, E, W)$ a positive weighted and undirected graph, where V is the set of vertices, E the set of edges, and $W \in \mathbb{R}^{N \times N}$ its weighted adjacency matrix. The graph has no self-loops, i.e. $W_{uu} = 0, \forall u \in V$. The degree of v is $d_v = \sum_{u \in \text{ng}(v)} W_{uv}$, where $u \in \text{ng}(v)$ indicates that u is a neighbor of v. With these elements, we can define the combinatorial Laplacian operator associated with $\mathcal{G}$ as $\mathcal{L} = \text{diag}((d_v)_{v \in V}) - W$, where $\text{diag}(\cdot)$ is a diagonal matrix with the elements of the input vector in its diagonal.

Problem statement. Suppose that we observe N synchronous data streams, each associated to a node of a known connected graph G. Let $x_{v,t}$ be the observation at node v at time t. We suppose a common input space for all nodes, i.e. $x_{v,t} \in \mathcal{X}$, $\forall v \in V$, $t \in \{1, ...\}$. Furthermore, the observations are independent in time, which is a standard hypothesis in kernel-based change-point detection literature [7]–[10].

Consider as change-point the timestamp τ at which the distribution associated with the streams of the nodes belonging to a set C changes:

$$\begin{cases} t < \tau & x_{v,t} \sim p_v; \\ t \geq \tau & x_{v,t} \sim q_v; \end{cases} \tag{1}$$

where $p_v \neq q_v$ if $v \in C$, otherwise $p_v = q_v$. We consider all p_v, q_v, C, τ to be unknown. Moreover, we expect C to depend on the graph structure. A simple example with signals $\mathcal{X} \subset \mathbb{R}^2$ at each node, is shown in Fig. 1. For each node v, let the sample of n consecutive observations, indexed by t, be the set:

$$\mathcal{X}_{v,t} = \{x_{v,t-n}, x_{v,t-(n-1)}, ..., x_{v,t-1}\}. \tag{2}$$

Our approach compares the two subsequent samples (datasets), $\mathcal{X}_{v,t}$ and $\mathcal{X}_{v,t+n}$, for each node to decide whether they follow the same distribution. The dissimilarity between the two samples is quantified via an approximation of the Pearson's divergence (PE-divergence) [11].

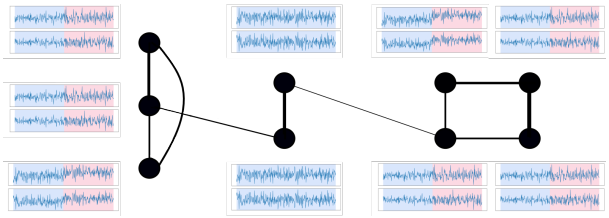

Fig. 1: Example of different two-dimensional data streams observed over the nodes of a weighted graph. A change occurs in a subset of nodes, and the change-point is the moment when the color changes in the time-series. The 3 nodes on the left have a change associated with a shift in the covariance matrix, while the 4 nodes on the right experience a change related to a shift in the mean of their streams.

III. Problem formulation and solution

The proposed *Online Centralized Kernel- and Graph-based* (OCKG) change-point detection method capitalizes over the connection between the approximation of the Pearson's PE-divergence and the likelihood-ratio estimation, which allows the comparison of the samples $\mathcal{X}_{v,t}$ and $\mathcal{X}_{v,t+n}$. Since by definition the PE-divergence is a non-symmetric similarity measure, i.e. generally $PE(p,q) \neq PE(q,p)$, we choose to approximate both those quantities. The OCKG method comprises three main tasks, to which the next subsections are devoted:

- **A.** ***Estimation***: When a new observation arrives at time t, we estimate the vector of relative likelihood-ratios $\vec{\mathbf{r}}^{\alpha}{}_t(\cdot) = (\vec{r}^{\alpha}{}_{1,t}(\cdot), ..., \vec{r}^{\alpha}{}_{N,t}(\cdot))$, between the samples of observations $\mathcal{X}_t$, $\mathcal{X}_{t+n}$, and for the reverse sample order $\overleftarrow{\mathbf{r}}^{\alpha}{}_t(\cdot)$ with samples $\mathcal{X}_{t+n}$, $\mathcal{X}_t$.
- **B.** ***Detection***: The estimated likelihood-ratios $\vec{\mathbf{r}}^{\alpha}{}_t(\cdot)$ and $\overleftarrow{\mathbf{r}}^{\alpha}{}_t(\cdot)$ are used to approximate the respective PE-divergences, $\hat{PE}^{\alpha}_v(\mathcal{X}_t, \mathcal{X}_{t+n})$ and $\hat{PE}^{\alpha}_v(\mathcal{X}_{t+n}, \mathcal{X}_t)$. Then, we define node scores, $\{S_v\}_{v\in V}$ based on the latter approximations. Finally, the node scores are aggregated into a global score indicating whether a change has occurred in the system.
- **C.** ***Identification***: Once a change-point is spotted, we use the node scores $\{S_v\}_{v\in V}$ to identify the nodes at which the change occurred, hence identifying the set C.

A. Estimation

Our graph-related objective is to estimate jointly the N node-level α-*relative likelihood-ratio functions* $r^{\alpha}_v(x)$, $v \in V$, given a user parameter $\alpha \in (0,1)$, each one being associated with the node v's pdfs p_v and q_v and defined as:

$$r^{\alpha}_v(x) = \frac{q_v(x)}{(1-\alpha)p_v(x) + \alpha q_v(x)} = \frac{q_v(x)}{p^{\alpha}_v(x)} \leq \frac{1}{\alpha}, \qquad (3)$$

where $p^{\alpha}_v(x) = (1-\alpha)p_v(x) + \alpha q_v(x)$. The relative likelihood-ratio r^{α}_v has two main advantages compared to the usual ratio $r_v(x) = \frac{q_v(x)}{p_v(x)}$: i) it is well-defined even if p_v is not absolutely continuous with respect to q_v, and ii) it is a bounded function, which is a property that facilitates its numerical estimation.

The PE-divergence between p^{α}_v and q_v is given in terms of the relative likelihood-ratio as follows:

$$\begin{aligned} PE(p^{\alpha}_v \| q_v) &= \frac{1}{2}\mathbb{E}_{p^{\alpha}_v(y)}[(r^{\alpha}_v(y)-1)^2] \\ &= \mathbb{E}_{q_v(x')}[r^{\alpha}_v(x')] - \frac{(1-\alpha)}{2}\mathbb{E}_{p_v(x)}[r^{\alpha}_v(x)^2] \\ &\quad - \frac{\alpha}{2}\mathbb{E}_{q_v(x')}[r^{\alpha}_v(x')^2] - \frac{1}{2}, \end{aligned} \qquad (4)$$

where the expectations are taken w.r.t. the probability measure appearing as subindex. The joint estimation of the vector-valued function $\mathbf{r}^{\alpha} = (r^{\alpha}_1, ..., r^{\alpha}_N)$ is based on the collaborative likelihood-ratio estimation framework called GRULSIF [4]. The main hypotheses of that framework are that: each r^{α}_v is an element of a Reproducing Kernel Hilbert Space (RKHS), and that the graph signal $\mathbf{r}^{\alpha}(x) = (r^{\alpha}_1(x), ..., r^{\alpha}_N(x))$ is expected to be graph-smooth at any moment. It is easy to verify that the latter holds for all nodes when there is no change since $p_v = q_v$, and $\mathbf{r}^{\alpha}(x)$ is the constant vector $\mathbf{1}_M$, which is a perfectly smooth graph signal. Then, our graph-smooth assumption for $\mathbf{r}^{\alpha}(x)$ implies also that a change in the nodes would essentially respect the graph structure.

1) Cost function: Let us introduce the RKHS $\mathbb{H}$ equipped with a reproducing kernel $\mathrm{K}: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, with the associated inner-product $\langle \cdot, \cdot \rangle_{\mathbb{H}}$ and feature map $\phi(\cdot) : \mathcal{X} \to \mathbb{H}$. Let $\hat{r}_v(\cdot)$ be the function approximating $r^{\alpha}_v(\cdot)$; we suppose $\hat{r}_v(\cdot)$ is a linear model of the form $\hat{r}_v(x) = \langle \theta_v, \phi(x) \rangle_{\mathbb{H}}$, where $\theta_v \in \mathbb{H}$. In practice, we will have access to L elements of a global dictionary $D = \{x_1, ..., x_L\}$ shared by all nodes. Therefore, the vectorized form of the kernel feature map is $\phi(x) = (\mathrm{K}(x, x_1), ..., \mathrm{K}(x, x_L))$, $\forall x \in \mathcal{X}$ and $\theta_v \in \mathbb{R}^L$. The linear model can now be expressed as:

$$\hat{r}_v(x) = \textstyle\sum_{l=1}^{L} \theta_{v,l} \mathrm{K}(x, x_l), \qquad (5)$$

and by definition, it holds: $|\hat{r}_v(x) - \hat{r}_u(x)| \leq \|\phi(x)\| \, \|\theta_u - \theta_v\|$. If $\sup_{x\in\mathcal{X}} \mathrm{K}(x,x) \leq \kappa$, then we can guarantee that $\hat{r}_v(x)$ and $\hat{r}_u(x)$ are close if the parameters θ_u and θ_v are close as well. Suppose that, at time t, for each node v we have access to observations coming from the two probabilistic models p_v and q_v, which we denote by $\mathcal{X}_{v,t}$ and $\mathcal{X}'_{v,t}$. We define the elements:

$$\begin{aligned} &H_{v,t} = \frac{1}{n}\sum_{x\in\mathcal{X}_{v,t}} \phi(x)\phi(x)^{\mathsf{T}}, \quad H'_{v,t} = \frac{1}{n}\sum_{x\in\mathcal{X}'_{v,t}} \phi(x)\phi(x)^{\mathsf{T}}, \\ &h'_{v,t} = \frac{1}{n}\sum_{x\in\mathcal{X}'_{v,t}} \phi(x), \end{aligned} \qquad (6)$$

with which we define the optimization Problem 7 that is made of two terms: the first term is a least square cost function aiming to approximate the relative likelihood-ratio at the node-level; the second term induces smoothness to the functions $\hat{r}_u$ and $\hat{r}_v$ by making their associated parameters, θ_u and θ_v of any

two connected nodes u and v, to be similar, while controlling the risk of overfitting via the penalization term $\|\theta_u\|^2$ [12]:

$$\min_{\Theta\in\mathbb{R}^{NL}} \Phi_t(\theta) = \min_{\Theta\in\mathbb{R}^{NL}} \frac{1}{N}\sum_{v\in V} \frac{(1-\alpha)\theta_v^\mathsf{T} H_{v,t}\theta_v}{2} + \frac{\alpha\theta_v^\mathsf{T} H'_{v,t}\theta_v}{2} - \frac{1}{N}\sum_{v\in V}\theta_v^\mathsf{T} h'_{v,t} + \frac{\lambda}{4}\sum_{u,v\in V} W_{uv}\|\theta_u-\theta_v\|^2 + \frac{\lambda\gamma}{2}\sum_{v\in V}\|\theta_v\|^2. \tag{7}$$

Let $\hat{\Theta}_t = (\hat{\theta}_1^\mathsf{T}, \hat{\theta}_2^\mathsf{T}, ..., \hat{\theta}_N^\mathsf{T})$ be the solution of Problem 7, and $\{\hat{r}_{v,t}(\cdot)\}_{v\in V}$ be the estimated relative likelihood-ratio. Then we approximate the PE-divergence $PE(p_v^\alpha\|q_v)$ using an empirical approximation of Eq. 4 and the estimation $\hat{r}_{v,t}(\cdot)$:

$$\hat{PE}_v^\alpha(\mathcal{X}_{v,t}\|\mathcal{X}'_{v,t}) = -(1-\alpha)\frac{\hat{\theta}_{v,t}^\mathsf{T} H_{v,t}\hat{\theta}_{v,t}}{2} - \alpha\frac{\hat{\theta}_{v,t}^\mathsf{T} H'_{v,t}\hat{\theta}_{v,t}}{2} + \hat{\theta}_{v,t}^\mathsf{T} h'_{v,t} - \frac{1}{2}. \tag{8}$$

The lack of symmetry of Pearson's PE-divergence is important for the change-point detection task. At every time t, we need to compare the two samples associated with the probabilistic models described by $\{p_v\}_{v\in V}$ and $\{q_v\}_{v\in V}$, respectively. Depending on the which pdf is taken as numerator in $r_v^\alpha(\cdot)$, the associated PE-divergence may lead to different detection sensitivity. For this reason, we estimate the parameters $\vec{\Theta}_t$ and $\overleftarrow{\Theta}_t$ to approximate both $\hat{PE}_v^\alpha(\mathcal{X}_{v,t}\|\mathcal{X}_{v,t+n})$ and $\hat{PE}_v^\alpha(\mathcal{X}_{v,t+n}\|\mathcal{X}_{v,t})$.

2) Optimization procedure: The quadratic Problem 7 admits a closed-form solution. Nevertheless, in real applications the size of the graph and the dictionary may render this solution infeasible. For this reason, we propose to solve the problem via the Cyclic Block Coordinate Gradient Descent (CBCGD) method [13], [14]. In our formulation, each block of variables is associated with a node v, thus it contains θ_v. Before applying CBCGD, we need first to define a fixed order for updating the variables, which in our case is arbitrary as it is not important for the convergence. Let $\theta_{<v}$ be the set of variables that were updated before v, and $\theta_{\geq v}$ be the complement of that set. Then, the i-th update of the parameter $\hat{\theta}_{v,t}$ is performed according to the schema:

$$\hat{\theta}_{v,t}^{(i)} = \frac{1}{\eta_{v,t}+\lambda\gamma}\Bigg[\eta_{v,t}\hat{\theta}_{v,t}^{(i-1)} \overbrace{-\left(\frac{(1-\alpha)H_{v,t}+\alpha H'_{v,t}}{N}\hat{\theta}_{v,t}^{(i-1)} - \frac{h'_{v,t}}{N}\right)}^{\text{component depending on node } v} \overbrace{-\lambda\left(d_v\hat{\theta}_{v,t}^{(i-1)} - \sum_{u\in \text{ng}(v)} W_{uv}\left(\hat{\theta}_{u,t}^{(i)}\mathbf{1}_{u<v} + \hat{\theta}_{u,t}^{(i-1)}\mathbf{1}_{u\geq v}\right)\right)}^{\text{component depending on the graph}}\Bigg]. \tag{9}$$

When no change has occurred, we expect the problem instances Φ_t and Φ_{t+1} to be similar. In that case, we can initialize the problem for finding Φ_{t+1} with the solution $\hat{\Theta}_t$. In fact, our problem being quadratic, we can prove that by initializing with $\hat{\Theta}_t$ the problem at time $t+1$ can be solved with a manageable number of $O(\log^2(N))$ cycles. This is a consequence of Theorem 3 in [4].

B. Detection and identification

A well-known property of $PE(p,q)$, is that it becomes zero if and only if $p=q$, which makes it a good candidate as a score to validate whether a change exists [15]. Then, the definition of a node-level score comes naturally:

$$S_{v,t} = \max\{\hat{PE}_v^\alpha(\mathcal{X}_{v,t},\mathcal{X}_{v,t+n}) + \hat{PE}_v^\alpha(\mathcal{X}_{v,t+n},\mathcal{X}_{v,t}), 0\}. \tag{10}$$

The maximum is taken as the approximations can be negative. Next we define the global score as $S_t = \sum_{v\in V} S_{v,t}$, which triggers a global alarm when $S_t \geq \eta$, with $\eta > 0$ being a threshold parameter fixed by the user. The moment τ at which this global alarm fires, is also the estimated occurrence time of the associated change-point. Once a change-point has been detected, we need to identify the affected node subset C. For this, we identify the nodes that satisfy $S_{v,t} > \eta_v$, where $\{\eta_v\}_{v\in V}$ is a set of positive constants given by the user. Alternatively, the set of parameters $\{\eta_v\}_{v\in V}$ could be selected via a permutation test, as described in [16], although that would be computationally expensive for a detection method designed to operate in near real-time.

The OCKG pseudocode is detailed in Alg. 1. Notice that we expect by design $\hat{PE}_v(\cdot,\cdot)$ to get its maximum value when it compares $\mathcal{X}_{v,\tau}$ and $\mathcal{X}_{v,\tau+n}$, which means there is always a time lag of length n (observations) in the detection of τ. We desire n to be as small as possible, yet guarantying a good identification of the nodes of interest.

Dictionary. Earlier, we made the implicit hypothesis that we have access to a predefined dictionary D. To build a dictionary in an online fashion, we follow the approach of [17]: at each time t a *coherence* measure assesses the linear dependency of the incoming observations with the current elements of the dictionary. The new datapoint is added into the dictionary only if the coherence is smaller than a given threshold μ_0.

IV. Experiments

In this section, we use two synthetic scenarios featuring different change-points, graph structures, and window sizes, to compare the performance of the proposed OCKG detector with alternative non-parametric methods. First, OCKG-POOL is a variant of the proposed OCKG method that ignores the graph structure (i.e. $W = \mathbf{0}_{M\times M}$), and serves as a baseline for assessing the benefit of using the graph. Second, Nougat [5], [18] is a closely related non-parametric method of the literature, which detects a change in a cluster of nodes; it estimates the node-level likelihood-ratio via kernel methods and a stochastic gradient descent. At each time t, a single step of stochastic gradient descend is performed, and the updated function is evaluated at time $t+1$ with the new incoming observation. This is done independently for each node. The resulting evaluation of the estimated function is used to construct a graph signal. Finally, Nougat filters the signal with the Graph Fourier Scan Statistic (GFSS), a graph-based statistical test that has been used for detecting nodes with anomalous activity [19].

Algorithm 1 The OCKG detector

Input: $\alpha \in (0,1)$: parameter of the relative likelihood-ratio (see Eq. 3); n: the size of the sample to use; D_1, D_2: precomputed dictionaries with L_1 and L_2 elements, respectively; $(\sigma_1^*, \lambda_1^*, \gamma_1^*), (\sigma_2^*, \lambda_2^*, \gamma_2^*)$: optimal hyperparemeters (see Alg. 1 in [4]); μ_0: coherence threshold controlling the dictionary creation; L: the maximum dictionary size; tol: tolerated relative error for the optimization process; $\eta, \{\eta_v\}$: threshold to raise a global alarm, η_v threshold to raise an alarm at node v.
Output: $\hat{\tau}$: detected change-point; $\hat{C}$: set of nodes where the change is observed.

■ **Initialization of parameters**
1: $\vec{\Theta}_n^{(0)} = \breve{\Theta}_n^{(0)} = \mathbf{0}_{LN}$
■ **Online estimation and detection**
2: **for** $t \in \{n, \dots, \}$ **do**
3: **for** $v \in \{1, \dots, N\}$ **do**
4: Observe $x_{v,t+n-1}$ and update the sliding windows $\mathcal{X}_t, \mathcal{X}_{t+n}$ (Eq. 2)
□ **Dictionary update**
5: **if** $\max_{l \in \{1,\dots,L_1\}} k(x_{v,l}, x_{v,t+n-1}) \leq \mu_0$ **then**
6: Add $x_{v,t+n-1}$ to the dictionary D_1
7: If the maximum dictionary size is reached, delete the element with the highest coherence
8: **end if**
9: **end for**
□ **Parameters update**
10: Define $\vartheta_v = [\theta_{v,t-1}^\intercal, \mathbf{0}_{d_1}]$, ($d_1$ the number of new elements added to the dictionary)
11: Initialize $\vec{\theta}_{v,t-1}^{(0)} = \vartheta_v$
12: Fix $\mathcal{X} = \mathcal{X}_t$ and $\mathcal{X}' = \mathcal{X}_{t+n}$
13: **for** $v \in \{1, \dots, N\}$ **do**
14: Compute the quantities H_v, H'_v, h'_v. (see Eq. 6)
15: Fix $\eta_{v,t} = e_{\max}\left(\frac{(1-\alpha)H_{v,t} + \alpha H'_{v,t}}{N} + \lambda d_v \mathbb{I}_{L_1}\right)$
16: **end for**
17: **while** $\left\|\vec{\Theta}_t^{(i)} - \vec{\Theta}_t^{(i-1)}\right\| > \epsilon$ **do**
18: **for** $v \in \{1, \dots, N\}$ **do**
19: Update $\vec{\theta}_v^{(i)}$ (see Eq. 9)
20: **end for**
21: **end while**
22: **for** $v \in \{1, \dots, N\}$ **do**
23: Estimate $\hat{PE}_v^\alpha(\mathcal{X}_{v,t}, \mathcal{X}'_{v,t})$ (see Eq. 8)
24: **end for**
25: Fix $\mathcal{X} = \mathcal{X}_{t+n}$ and $\mathcal{X}' = \mathcal{X}_t$
26: Repeat steps 6 – 22 to compute $\breve{\Theta}_t$ and $\hat{PE}_v^\alpha(\mathcal{X}'_{v,t}, \mathcal{X}_{v,t})$
27: □ **Online detection and Identification**
28: Compute the node scores $S_{v,t}$ (see Eq. 10)
29: Compute the global score $S_t = \sum_{v \in V} S_{v,t}$
30: **if** $S_t > \eta$ **then**
31: A change-point is detected at $\hat{\tau} = t$
32: **if** $S_{v,t} > \eta_v$ **then**
33: Add v to $\hat{C}$
34: **end if**
35: **end if**
36: **end for**
37: Return $\hat{\tau}$ and $\hat{C}$

Scenario I: Changes in node clusters. *(Bivariate Gaussian distribution → Gaussian copula with uniform marginals.)* We sample a Stochastic Block Model with 4 clusters, $C1$,...,$C4$, each containing 20 nodes. In this experiment, all nodes initially follow a bivariate Gaussian model with the same covariance matrix and mean vector. Then we pick a cluster C at time $t = 2000$. From that moment, nodes of C start generating observations from a Gaussian copula ($\sim GC$) whose marginals follow uniform distributions ($\sim U(-c, c)$):

$$\begin{aligned} &(x,y) \sim N(\mu, \Sigma), \quad \mu = (0,0), \quad \Sigma_{x,x} = 1, \Sigma_{x,y} = \frac{4}{5} \\ &\qquad \downarrow \\ &(x,y) \sim GC, \quad \Sigma_{x,x} = 1, \Sigma_{x,y} = \frac{4}{5}. \end{aligned} \tag{11}$$

The parameter c is chosen so as the mean vector and covariance matrix before and after the change-point are the same.

Scenario II: Changes in a subset of connected nodes. *(Shift in the mean of one cluster.)* We generate a Barabási-Albert graph with 100 nodes. For each instance of the experiments, we generate C by selecting a node at random with probability proportional to its degree, and then by picking all the nodes that are within a distance of 4 in the graph. These nodes suffer from a change in the probability model generating their associated streams, and the change is set to occur at time $t = 1000$. In each cluster, the observed streams before and after the change are drawn from a different multivariate Gaussian distribution of dimension 3:

$$\begin{aligned} &x_v \sim N(\mu, \Sigma), \mu = \mathbf{0}_3, \Sigma_{i,i} = 1, \Sigma_{1,2} = \frac{4}{5}, \Sigma_{3,1} = 0 \\ &\qquad \downarrow \\ &x_v \sim N(\mu, \Sigma), \mu = (1,0,0), \Sigma_{i,i} = 1, \Sigma_{1,2} = \frac{4}{5}, \Sigma_{3,1} = 0. \end{aligned} \tag{12}$$

Experimental Scenario I	Detector	Detection Delay (std)	AUC (std)	Precision
n =125	OCKG $\alpha = 0.1$	126.26 (11.95)	**0.89 (0.05)**	**1.00**
	OCKG $\alpha = 0.5$	129.67 (11.37)	0.85 (0.06)	0.98
	OCKG-POOL $\alpha = 0.1$	123.72 (24.08)	0.82 (0.05)	0.58
	OCKG-POOL $\alpha = 0.5$	131.90 (21.02)	0.80 (0.05)	0.80
	Nougat	146.50 (70.74)	0.56 (0.22)	0.12
n =250	OCKG $\alpha = 0.1$	252.72 (14.82)	**0.93 (0.03)**	**1.00**
	OCKG $\alpha = 0.5$	251.31 (21.82)	0.89 (0.04)	0.98
	OCKG-POOL $\alpha = 0.1$	249.54 (25.20)	0.86 (0.04)	0.92
	OCKG-POOL $\alpha = 0.5$	245.32 (22.20)	0.87 (0.04)	0.94
	Nougat	273.50 (96.59)	0.67 (0.19)	0.20
n =500	OCKG $\alpha = 0.1$	502.70 (6.49)	**0.99 (0.00)**	**1.00**
	OCKG $\alpha = 0.5$	500.84 (5.15)	**0.99 (0.00)**	**1.00**
	OCKG-POOL $\alpha = 0.1$	506.20 (18.31)	**0.99 (0.00)**	**1.00**
	OCKG-POOL $\alpha = 0.5$	501.90 (7.86)	**0.99 (0.00)**	**1.00**
	Nougat	576.86 (129.27)	0.66 (0.20)	0.74

Experimental Scenario II	Detector	Detection Delay (std)	AUC (std)	Precision
n =25	OCKG $\alpha = 0.1$	25.44 (1.96)	**0.97 (0.02)**	**1.00**
	OCKG $\alpha = 0.5$	25.06 (1.34)	**0.97 (0.02)**	0.96
	OCKG-POOL $\alpha = 0.1$	24.51 (1.68)	0.91 (0.03)	0.82
	OCKG-POOL $\alpha = 0.5$	24.44 (2.03)	0.93 (0.02)	0.86
	Nougat	34.25 (13.80)	0.64 (0.17)	0.08
n =50	OCKG $\alpha = 0.1$	50.38 (1.21)	**0.99 (0.01)**	**1.00**
	OCKG $\alpha = 0.5$	50.67 (1.48)	0.96 (0.04)	0.98
	OCKG-POOL $\alpha = 0.1$	48.55 (5.14)	0.91 (0.04)	0.98
	OCKG-POOL $\alpha = 0.5$	49.63 (2.38)	**0.99 (0.01)**	0.98
	Nougat	77.84 (10.24)	0.75 (0.17)	0.76
n =100	OCKG $\alpha = 0.1$	100.52 (1.25)	0.99 (0.00)	**1.00**
	OCKG $\alpha = 0.5$	100.16 (0.64)	**1.00 (0.00)**	**1.00**
	OCKG-POOL $\alpha = 0.1$	99.86 (1.23)	0.99 (0.00)	**1.00**
	OCKG-POOL $\alpha = 0.5$	100.38 (1.01)	0.99 (0.00)	**1.00**
	Nougat	127.52 (13.87)	0.77 (0.16)	0.88

TABLE I: Performance comparison between change-point detectors in the two synthetic experimental scenarios. Three window sizes (n) are considered in each scenario. The mean and standard deviation of the score is based on 50 instances of the experiments.

Discussion on the results. Tab. I and Fig. 2 report the average AUC of the ROC curves and their standard deviation, as well as the percentage of times that the change-point τ was successfully detected. As expected, in both experiments, all methods perform generally better as the number of observations increases. Nougat requires the largest amount of observations to detect τ and identify the set C. We believe that is due to the stochastic gradient descent step, which produces more noisy detection scores compared with other methods. In most cases, the comparative advantage of OCKG variants is best seen

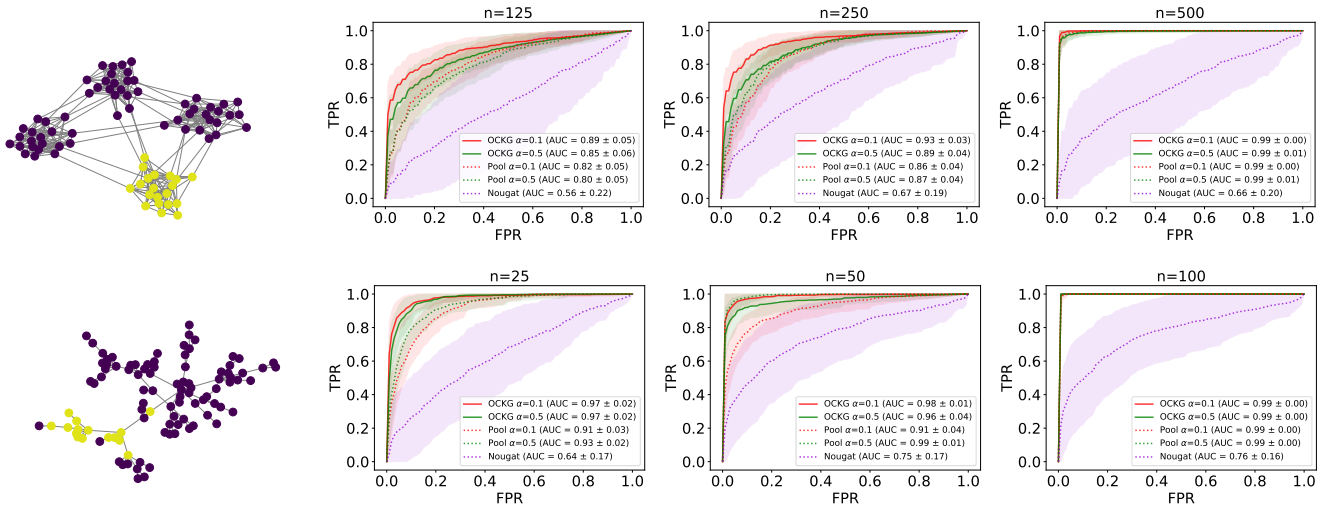

Fig. 2: Each row presents results for the two synthetic scenarios. In the first column, an instance of the simulated nodes that suffer a change is shown (yellow nodes). The rest of the columns show the mean ROC curves along with their standard deviations for the different considered window sizes n. The mean and standard deviations are estimated based on 50 random instances of the experiments.

when fewer observations are available, as the Precision and AUC scores suggest. Exploiting the graph structure improves the precision of the change-point detection and reduces the detection delay. Finally, the results suggest using small α parameter values (e.g. $\alpha = 0.1$) to keep the OCKG detector sensitive to situations in which Pearson's divergence between the pre- and post-change pdfs is rather small.

V. Conclusions

In this paper, we introduced the OCKG change-point detection and localization method for multivariate streams over the nodes of a graph. Among its appealing properties, there is its non-parametric formulation that integrates the a priori provided information of the graph structure, and its capacity to spot and localize in a graph different types of change-point with minimum hypotheses. Future work may relax the expectation of the framework that the nodes experience changes at the same moment, which may currently lead to a long detection delay in the case of escalating phenomena that affect gradually more nodes over time. Combining online likelihood-ratio estimation (e.g. [20]) with graph diffusion effects, is a promising direction to address this limitation.

References

[1] A. Tartakovsky, I. Nikiforov, and M. Basseville, *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. Taylor & Francis, CRC Press, 2014.

[2] A. Tartakovsky, *Sequential change detection and hypothesis testing: general non-i.i.d. stochastic models and asymptotically optimal rules*. Chapman & Hall/CRC, 2021.

[3] L. Xie, S. Zou, Y. Xie, and V. V. Veeravalli, "Sequential (quickest) change detection: Classical results and new directions," *IEEE Journal on Selected Areas in Information Theory*, 2021.

[4] A. de la Concha, N. Vayatis, and A. Kalogeratos, "Collaborative likelihood-ratio estimation over graphs," *arXiv:2205.14461*, 2024.

[5] A. Ferrari and C. Richard, "Non-parametric community change-points detection in streaming graph signals," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2020, pp. 5545–5549.

[6] A. de la Concha, N. Vayatis, and A. Kalogeratos, "Online centralized non-parametric change-point detection via graph-based likelihood-ratio estimation," *arXiv:2301.03011*, 2023.

[7] S. Arlot, A. Celisse, and Z. Harchaoui, "A kernel multiple change-point algorithm via model selection," *Journal of Machine Learning Research*, vol. 20, no. 162, pp. 1–56, 2019.

[8] S. Li, Y. Xie, H. Dai, and L. Song, "Scan B-statistic for kernel change-point detection," *Sequential Analysis*, vol. 38, no. 4, pp. 503–544, 2019.

[9] Z. Harchaoui, E. Moulines, and F. Bach, "Kernel change-point analysis," in *Advances in Neural Information Processing Systems*, vol. 21, 2008.

[10] I. Bouchikhi, A. Ferrari, C. Richard, A. Bourrier, and M. Bernot, "Kernel based online change point detection," in *European Signal Processing Conf.*, 2019, pp. 1–5.

[11] K. Pearson, "X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *Lond. Edinb. Dubl. Phil. Mag.*, vol. 50, no. 302, pp. 157–175, 1900.

[12] D. Sheldon, "Graphical Multi-Task Learning," Cornell University, Tech. Rep., 2008.

[13] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM Journal on Optimization*, vol. 23, pp. 2037–2060, 2013.

[14] X. Li, T. Zhao, R. Arora, H. Liu, and M. Hong, "On faster convergence of cyclic block coordinate descent-type methods for strongly convex minimization," *Journal of Machine Learning Research*, vol. 18, no. 184, pp. 1–24, 2018.

[15] Y. Kawahara and M. Sugiyama, "Sequential change-point detection based on direct density-ratio estimation," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 2, pp. 114–127, 2012.

[16] A. de la Concha, N. Vayatis, and A. Kalogeratos, "Collaborative non-parametric two-sample testing," in *Int. Conf. on Artificial Intelligence and Statistics*, 2025.

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

[18] A. Ferrari, C. Richard, A. Bourrier, and I. Bouchikhi, "Online change-point detection with kernels," 2021.

[19] J. Sharpnack, A. Rinaldo, and A. Singh, "Detecting anomalous activity on networks with the graph fourier scan statistic," *IEEE Trans. on Signal Processing*, vol. 64, no. 2, pp. 364–379, 2016.

[20] A. D. de la Concha Duarte, N. Vayatis, and A. Kalogeratos, " Online non-parametric likelihood-ratio estimation by Pearson-divergence functional minimization ," in *Int. Conf. on Artificial Intelligence and Statistics*, 2024.