DISTRIBUTED CHANGE POINT DETECTION IN STREAMING MANIFOLD-VALUED SIGNALS OVER GRAPHS

Xiuheng Wang *, *Ricardo Augusto Borsoi* †, *Cédric Richard* *, *André Ferrari* *

* Université Côte d'Azur, CNRS, OCA, France
† Université de Lorraine, CNRS, CRAN, France
xiuheng.wang@oca.eu, raborsoi@gmail.com, cedric.richard@unice.fr, andre.ferrari@univ-cotedazur.fr

ABSTRACT

Signal processing methods over graphs and networks have recently been proposed to detect change points occurring in localized communities of nodes. Nevertheless, all these methods are mostly limited to time series data in Euclidean spaces. In this paper, we devise a distributed change point detection method for streaming manifold-valued signals over graphs. This framework combines a local test statistic at each node to account for the geometry of the data on a Riemannian manifold, with a fully distributed graph filter that incorporates information on network topology.

Index Terms— Graph signal processing, distributed, change point detection, Riemannian manifold, graph filtering

1. INTRODUCTION

Change point detection (CPD) aims to detect abrupt changes in the distribution of monitored data and is recognized as a central task in streaming data analysis. Recently, one trend is to detect anomalous events in time series data measured over the nodes of a network [1–3]. This problem is of significant interest with wide-ranging applications in the fields of physics, biology, and finance to cite a few.

The change points in these problems often occur in groups of highly connected nodes (i.e., communities) of networks, represented as graphs. Graph signal processing tools, including spectral analysis [4, 5] and filtering [6–9], are indicated to combine such graph topology information with measurements collected at each node. In [1], the authors introduce the Graph Fourier Scan Statistic (GFSS) and a low-pass filter based on graph Fourier transform to detect anomalies over graph signals. The work in [2] proposes an online CPD algorithm with a fully distributed and adaptive GFSS to monitor for change points in large-scale networks. This algorithm was applied for CPD in multi-channel image sequences in [10]. An online and distributed strategy based on likelihood ratio estimation with kernel machinery is also described in [3] to detect change points over graphs with few assumptions on the data distribution.

All the distributed CPD frameworks listed above are limited to real- or vector-valued time series in Euclidean spaces. To the best of our knowledge, there is no generalization of CPD techniques over graphs where the streaming data collected at each node belongs to a Riemannian manifold. A representative application is the detection of change points in videos with a sequence of spatially localized covariance descriptors [11]. In this application, the changes usually affect multiple related regions in the image, however, this information is not taken into account by algorithms that process each region separately. One possibility to leverage this information is to design a graph that describes the relationship between regions, and then perform the detection of change points cooperatively. To devise algorithms for manifold-valued data, it is important to take the geometry of the data space into account by exploiting, e.g., an appropriate Riemannian metric [12,13].

This paper introduces a distributed framework for detecting change points on manifold-valued signals collected over a network. The proposed method is built upon a test statistic derived for streaming data on a Riemannian manifold which accounts for the geometry of the data, and a fully distributed graph filter that exploits the network topology information to enhance the detection of anomalies localized in unknown communities of nodes. Simulation results show that taking manifold geometry and graph topology into account can significantly improve the detection performance.

2. BACKGROUND

We shall now introduce some basic concepts of Riemannian geometry [13], taking as an example the manifold of $d \times d$ symmetric positive definite (SPD) matrices, denoted by $\mathcal{S}_d^{++}$. A *Riemannian manifold* $(\mathcal{M}, g)$ is defined by a constrained set $\mathcal{M}$ equipped with a *Riemannian metric* $g_x(\cdot,\cdot) : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ defined for each $x \in \mathcal{M}$, where $T_x\mathcal{M}$ is called the *tangent space* of $\mathcal{M}$ at x. The *geodesic distance* is defined as $d_{\mathcal{M}}(\cdot,\cdot) : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ and satisfies all conditions to be a metric. Let $f : \mathcal{M} \to \mathbb{R}$ be a smooth real-valued

The work of C. Richard was funded in part by the ANR under grant ANR-19-CE48-0002, and by the 3IA Côte d'Azur Senior Chair program.

function. The *Riemannian gradient* of f at $x \in \mathcal{M}$ is defined as the unique tangent vector $\nabla f(x) \in T_x\mathcal{M}$ satisfying $\frac{d}{dt}\big|_{t=0} f(\exp_x(tv)) = \langle \nabla f(x), v\rangle_x$ for all $v \in T_x\mathcal{M}$. The *exponential map* $w = exp_x(v)$ defines the point w of $\mathcal{M}$ located on the unique geodesic $\gamma_v(t)$ such that $\gamma_v(0) = x$, $\gamma_v'(0) = v$ and $\gamma_v(1) = w$. For instance, the geodesic distance between two SPD matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ can be computed [12] as:

$$d_{\mathcal{S}_d^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = \left\| \log(\boldsymbol{\Sigma}_2^{-\frac{1}{2}}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_2^{-\frac{1}{2}}) \right\|_F, \tag{1}$$

with $\| \, \|_F$ the Frobenius norm. The Riemannian gradient at $\boldsymbol{\Sigma}$ of the loss $d^2_{\mathcal{S}_d^{++}}(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_t)$ can be obtained by applying:

$$\frac{1}{2}\boldsymbol{\Sigma}\,(\boldsymbol{G}^\top + \boldsymbol{G})\boldsymbol{\Sigma}$$

to its Euclidean gradient $\boldsymbol{G}$. This gives us:

$$H(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_t) = 2\log(\boldsymbol{\Sigma}\boldsymbol{\Sigma}_t^{-1})\boldsymbol{\Sigma}\,. \tag{2}$$

Let $\boldsymbol{\xi} \in T_{\boldsymbol{\Sigma}}\mathcal{S}_d^{++}$. A retraction $R_{\boldsymbol{\Sigma},\mathcal{S}_d^{++}} : T_{\boldsymbol{\Sigma}}\mathcal{S}_d^{++} \to \mathcal{S}_d^{++}$ is:

$$R_{\boldsymbol{\Sigma},\mathcal{S}_d^{++}}(\boldsymbol{\xi}) = \boldsymbol{\Sigma} + \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\xi}\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}\,. \tag{3}$$

This retraction is a second-order approximation of the exponential mapping on $\mathcal{S}_d^{++}$.

3. PROBLEM FORMULATION

We consider an undirected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ with N vertices in $\mathcal{N} = \{1, \ldots, N\}$ and M edges in $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ such that $(i, j) \in \mathcal{E}$ iff nodes i and j are connected. With the graph $\mathcal{G}$ is associated a $N \times N$ weighted adjacency matrix $\boldsymbol{W}$. Each entry $W_{i,j} \geqslant 0$ is the connection strength between nodes i and j, with non-zero value iff $(i, j) \in \mathcal{E}$. A community $\mathcal{C} \subset \mathcal{N}$ in $\mathcal{G}$ is a subset of nodes that are densely connected.

At each time instant $t \in \mathbb{N}$, we observe a signal over the graph $\mathcal{X}_t = \{\boldsymbol{x}_t(n)\}_{n=1}^N$, where $\boldsymbol{x}_t(n) \in \mathcal{M}$ denotes the measurement collected at node n, that lies on a Riemannian manifold $(\mathcal{M}, g)$. In this paper, the objective is to detect an abrupt change in the graph signal $\mathcal{X}_t$ that might occur at an unknown time t_r, called the *change point*. In particular, we assume that the changes occur in an unknown community $\mathcal{C}^*$ of $\mathcal{G}$, which means that:

$$\begin{aligned} t < t_r &: \ \boldsymbol{x}_t(n) \sim P_{0,n}\,, \\ t \geqslant t_r &: \ \boldsymbol{x}_t(n) \sim P_{1,n}\,, \end{aligned} \tag{4}$$

with

$$\begin{aligned} \forall n \in \mathcal{C}^*, &\quad P_{0,n} \neq P_{1,n}\,, \\ \forall n \notin \mathcal{C}^*, &\quad P_{0,n} = P_{1,n}\,. \end{aligned} \tag{5}$$

where $P_{0,n}$ and $P_{1,n}$ denote probability measures on $\mathcal{M}$ that represent the distribution of the signal $\boldsymbol{x}_t(n)$ before and after the change point t_r. For ease of notation, (4) considers only a single change point. However, the algorithm presented hereafter can handle multiple change points.

4. METHODOLOGY

Distributed CPD strategies [1–3] initially designed to handle time series signals in an Euclidean space cannot handle streaming data that lies on a manifold. In this work, we aim to design a new framework to detect change points in streaming manifold-valued signals over graphs. First, we consider an online CPD strategy on Riemannian manifolds to take the data geometry into account. Second, we leverage the graph topology by graph-filtering test statistics computed at each node, without compromising the manifold interpretation of the signals. Finally, the centralized graph filter is implemented in a fully distributed way, to provide an efficient CPD method for large-scale networks.

4.1. CPD in streaming manifold-valued signals

Some recent algorithms have been investigated for detecting change points in streaming manifold-valued data. For instance, an online and parametric CPD algorithm in [14] was specifically designed for the compound Gaussian distribution. An offline and non-parametric technique can be found in [15]. In this paper, we consider an online and non-parametric algorithm proposed in [11], which detects change points by monitoring for abrupt changes in the so-called *Karcher means* [16] of the streaming data. Consider a random signal $\boldsymbol{x}$ in $\mathcal{M}$ distributed according to P. Its Karcher mean is defined as:

$$\boldsymbol{m}^* = \underset{\boldsymbol{m} \in \mathcal{M}}{\arg\min} \left\{ f(\boldsymbol{m}) \triangleq \int d^2_{\mathcal{M}}(\boldsymbol{m}, \boldsymbol{x})\, dP(\boldsymbol{x}) \right\}. \tag{6}$$

The algorithm in [11] allows us to detect change points by comparing two Karcher means estimated using two stochastic Riemannian optimization methods, one rapidly approaching the information of new data, and another steadily progressing to emphasize a long-term trend. This method can be applied to detect change points in the distribution of $\boldsymbol{x}_t(n)$ at each node n by estimating its Karcher mean. Specifically, for each node n, using two Riemannian stochastic gradient descent (SGD) algorithms [17] with two distinct step sizes $0 < \lambda < \Lambda$, two Karcher means are estimated recursively as:

$$\boldsymbol{m}_{\lambda,t}(n) = R_{\boldsymbol{m}_{\lambda,t-1}(n)}\big(- \lambda H(\boldsymbol{m}_{\lambda,t-1}(n), \boldsymbol{x}_t(n))\big)\,, \tag{7}$$

$$\boldsymbol{m}_{\Lambda,t}(n) = R_{\boldsymbol{m}_{\Lambda,t-1}(n)}\big(- \Lambda H(\boldsymbol{m}_{\Lambda,t-1}(n), \boldsymbol{x}_t(n))\big)\,, \tag{8}$$

where $R_{\boldsymbol{m}}$ is a retraction map at $\boldsymbol{m}$, and $H(\boldsymbol{m}, \boldsymbol{x})$ denotes the Riemannian gradient of the loss function $f(\boldsymbol{m})$. The convergence rates of (7) and (8) are directly affected by λ and Λ. Constraint $\lambda < \Lambda$ implies that $\boldsymbol{m}_{\Lambda,t}(n)$ is more adaptive to new data while $\boldsymbol{m}_{\lambda,t}(n)$ has a longer memory.

By assessing the disparity between $\boldsymbol{m}_{\lambda,t}(n)$ and $\boldsymbol{m}_{\Lambda,t}(n)$ through the geodesic distance on $\mathcal{M}$, an adaptive CPD statistic $d_t(n)$ for each node n can be computed as:

$$d_t(n) = d_{\mathcal{M}}\big(\boldsymbol{m}_{\lambda,t}(n), \boldsymbol{m}_{\Lambda,t}(n)\big)\,. \tag{9}$$

Algorithm 1 CPD in streaming manifold-valued signals

```
Input: ${\boldsymbol{x}_t(n)}_{n=1}^N$, step sizes $\lambda, \Lambda$, threshold $\xi$.
  Initialize $\boldsymbol{m}_{\lambda,0}(n) = \boldsymbol{m}_{\Lambda,0}(n) = \boldsymbol{x}_0(n)$
  for $t = 1, 2, 3, \ldots$ do
      for $n = 1, \ldots, N$ do
          Update $\boldsymbol{m}_{\lambda,t}(n)$ and $\boldsymbol{m}_{\Lambda,t}(n)$ using (7) and (8)
          Compute $d_t(n)$ using (9)
      end for
      if $\exists\, n \in \mathcal{N}$: $d_t(n) > \xi$ then
          Flag $t$ as a change point
      end if
  end for
```

CPD at each node can then be performed by comparing $d_t(n)$ to a threshold ξ. The corresponding CPD procedure on manifolds is summarized in Algorithm 1.

4.2. Community CPD over graphs

Consider $\boldsymbol{d}_t = [d_t(1), \ldots, d_t(N)]^\top$. The CPD statistic computed in (1) at each node does not take into account the graph topology. To improve the localization of the communities that might contain a change point, we consider the graph filter $\boldsymbol{g}$ introduced in [1] for computing the GFSS. The GFSS aims to test if a graph signal with scalar measurements at each node is zero-mean against the hypothesis that there is a community of well-connected nodes where signals have a mean that differs from zero. We propose to apply the GFSS to the node-level test statistics $\boldsymbol{d}_t$ defined in (9) rather than original signals $\boldsymbol{x}_t$ to avoid loss of the manifold interpretation of problem (4).

Let us denote the normalized graph Laplacian of $\mathcal{G}$ by $\boldsymbol{L}$. Let $\boldsymbol{u}_n$ for $n = 1, \ldots, N$ be the set of orthonormal eigenvectors of $\boldsymbol{L}$ with μ_n being the associated eigenvalues. Given the node-level test statistics $\boldsymbol{d}_t$, the GFSS is defined as:

$$t_{\text{GFSS}}(\boldsymbol{d}_t) = \|\boldsymbol{g}_{\boldsymbol{d}_t}\|_2 \,, \tag{10}$$

$$\boldsymbol{g}_{\boldsymbol{d}_t} = \sum_{n=2}^{N} h^*(\mu_n)(\boldsymbol{u}_n^\top \boldsymbol{d}_t)\boldsymbol{u}_n \,, \tag{11}$$

where $\boldsymbol{g}_{\boldsymbol{d}_t}$ is the graph-filtered statistics, and $h^*(\mu)$ is the frequency response of the filter defined as [1]:

$$h^*(\mu) = \min\left\{1, \sqrt{\frac{\gamma}{\mu}}\right\}, \ \mu > 0 \,, \tag{12}$$

where $\gamma > 0$ being a tuning parameter.

To get more insight into the filtering procedure in (11), let us recall the role of the eigenvectors $\boldsymbol{u}_n$ of the graph Laplacian matrix $\boldsymbol{L}$ in spectral clustering [4, 5]. Consider the ideal case of a graph with $1 < k < N$ disconnected clusters of densely connected nodes. We denote by $\mathcal{C}_n$ the set of nodes in cluster n, with $n = 1, \cdots, k$. Each $\boldsymbol{u}_n$ is proportional to the indicator function of $\mathcal{C}_n$, and $\boldsymbol{u}_n^\top \boldsymbol{d}_t$ is therefore proportional to the sum of the $d_t(n)$'s in $\mathcal{C}_n$. This means that $(\boldsymbol{u}_n^\top \boldsymbol{d}_t)\boldsymbol{u}_n$ in (11) assigns the average value of the $d_t(n)$'s in $\mathcal{C}_n$ to each node in $\mathcal{C}_n$. As the number of communities k is unknown, the filter response in (12) is designed to penalize large numbers of clusters in (11). This assumption is also a cornerstone of spectral clustering methods [4, 5].

Algorithm 2 Distributed CPD in streaming manifold-valued signals over graphs

```
Input: $\{\mathcal{X}_t\}$, step sizes $\lambda, \Lambda$, threshold $\xi$
  Initialize $\boldsymbol{y}_{\ell,-1} = \boldsymbol{0}$
  for $t = 1, 2, 3, \ldots$ do
      $\forall\, n \in \mathcal{N}$, compute $d_t(n)$ as in Algorithm 1
      Set $\boldsymbol{d}_t = [d_t(1), \ldots, d_t(N)]^\top$
      for $\ell = 1, \ldots, K$ do
          $\boldsymbol{y}_{\ell,t} = \psi_\ell \boldsymbol{L}\boldsymbol{y}_{\ell,t-1} + \varphi_\ell \boldsymbol{d}_t$
          $\hat{\boldsymbol{g}}_{\boldsymbol{d}_t} = \sum_{\ell=1}^K \boldsymbol{y}_{\ell,t} + c\boldsymbol{d}_t$
      end for
      if $\exists\, n \in \mathcal{N}$ : $\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}(n) > \xi$ then
          Flag $t$ as a change point
      end if
  end for
```

4.3. Distributed implementation

The filtering operation as defined in (11) requires the eigendecomposition of the normalized graph Laplacian matrix $\boldsymbol{L}$. This is computationally expensive and hence cannot be scaled to large networks. A strategy to make our community CPD algorithm scalable is to substitute the filter in (11)–(12) with a distributed filter that can be implemented locally at each graph vertex [6, 7]. In contrast to these finite impulse response filters, an autoregressive moving average (ARMA) graph filter has been proposed in [8, 9]. This filter recursively aggregates signals in the neighborhood of each node, which therefore requires low computation and memory costs.

In the context of online community CPD, we propose to apply the parallel ARMA$_K$ graph filter [9], an approximation of the GFSS filter $\boldsymbol{g}$ defined in (11), to the streaming statistics $\boldsymbol{d}_t$ in (9), which leads to:

$$\boldsymbol{y}_{\ell,t} = \psi_\ell \boldsymbol{L}\boldsymbol{y}_{\ell,t-1} + \varphi_\ell \boldsymbol{d}_t, \ \boldsymbol{y}_{\ell,-1} = \boldsymbol{0}, \ \forall \ell = 1 \ldots K \,, \tag{13}$$

$$\hat{\boldsymbol{g}}_{\boldsymbol{d}_t} = \sum_{\ell=1}^{K} \boldsymbol{y}_{\ell,t} + c\boldsymbol{d}_t \,. \tag{14}$$

The operation $\boldsymbol{L}\boldsymbol{y}_{\ell,t-1}$ is a graph-shift which is performed locally at each node n by linearly combining the statistics as follows: $\sum_{k\in\mathcal{N}_p} L_{p,k}\, y_{\ell,t-1,k}$, where $\mathcal{N}_p$ is the neighborhood of node p, including p itself, $y_{\ell,t-1,k}$ is the k-th entry of $\boldsymbol{y}_{\ell,t-1}$ and $L_{p,k}$ the (p,k)-th entry of $\boldsymbol{L}$. This operation plays a central role in the fully distributed graph-filtering procedure of streaming statistics $\boldsymbol{d}_t$ as it only involves the values of the neighboring nodes over graphs. Note that there exists a series of appropriate parameters c and $\{(\psi_\ell, \varphi_\ell)\}_{\ell=1\ldots K}$ so that $h(\mu)$ closely approximates $h^*(\mu)$ in (12). The fully dis-

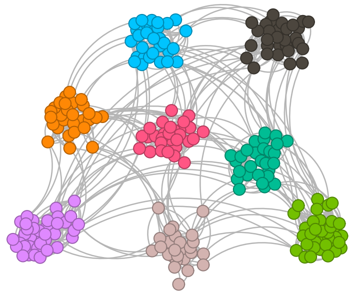

Fig. 1. Graph topology with colored communities $\mathcal{C}_i$.

tributed CPD procedure for streaming manifold-valued signals over graphs is described in Algorithm 2.

5. SIMULATIONS

We shall now illustrate the performance of the proposed approach using graph signals $\boldsymbol{x}_t(n)$ over manifold $\mathcal{S}_d^{++}$. The topology of the graph $\mathcal{G}$ used for simulations is illustrated in Fig. 1. It contains $p = 250$ nodes and $m = 2508$ edges, and 8 communities. These communities $\mathcal{C}_i$ have been unfolded using [18] and colored for visualization. We generated $\mathcal{X}_t$ as in (4) with a change point in community $\mathcal{C}^*$. Its nodes are colored in orange in Fig. 1. The synthetic matrices $\boldsymbol{\Sigma}_t \in \mathcal{S}_d^{++}$ with $d = 6$ were sampled from a Wishart distribution with the scaling matrix $\boldsymbol{V}$ and degrees of freedom d. We generated 800 samples and inserted the change point at $t_r = 500$ in (4) where we reset $\boldsymbol{V}$.

With $\{\boldsymbol{\Sigma}_t\}_{t\in\mathbb{N}}$ lying on $\mathcal{S}_d^{++}$ and the metric defined in (1), the Karcher means were estimated by minimizing

$$f(\boldsymbol{\Sigma}) = \mathbb{E}_{\boldsymbol{\Sigma}_t \sim P(\boldsymbol{\Sigma})}\left\{\left\| \log(\boldsymbol{\Sigma}_t^{-\frac{1}{2}}\boldsymbol{\Sigma}\boldsymbol{\Sigma}_t^{-\frac{1}{2}})\right\|_F^2\right\}, \tag{15}$$

using the Riemannian SGD algorithms in (7) and (8) with the stochastic gradient (2) and the retraction (3). Step sizes were set to $\lambda = 0.01$ and $\Lambda = 0.02$ and used to compute the online statistic in (9). Parameter γ of the GFSS filter (12) was set to 0.03 and K in the ARMA$_K$ filter to 4. The practical computation problems of parameters c and $(\psi_\ell, \varphi_\ell)$ for $\ell = 1, \ldots, K$ are discussed in [2].

To compare the detection performance of these algorithms, Monte Carlo simulations were performed to estimate the mean detection delay, average run length, and false alarm rate for daGFSS, $\boldsymbol{d}_t$ and $\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}$. Considering $\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}$ for illustration purposes, these metrics are defined as follows:

$$T_{\text{mdd}} = \inf\{t - t_r : \hat{\boldsymbol{g}}_{\boldsymbol{d}_t}(n) > \xi \,|\, n \in \mathcal{C}^*\}, \tag{16}$$

$$T_{\text{arl}} = \inf\{t : \hat{\boldsymbol{g}}_{\boldsymbol{d}_t}(n) > \xi \,|\, n \notin \mathcal{C}^*\}, \tag{17}$$

$$P_{\text{fa}} = P\big(\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}(n) > \xi \,|\, t > t_r, n \notin \mathcal{C}^*\big). \tag{18}$$

To illustrate the advantage of exploiting both manifold geometry and graph topology, we compared our Algorithm 2 to two baselines. The first baseline is daGFSS [2], originally designed for Euclidean data. We applied daGFSS to the vectorization of the lower triangular and diagonal parts of $\boldsymbol{\Sigma}_t$. The second baseline is the Karcher means-based CPD method on manifolds detailed in Algorithm 1, performed node-by-node without cooperation.

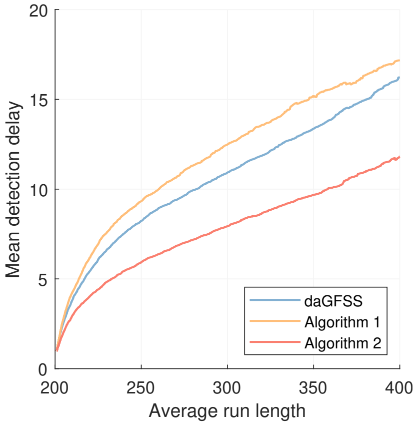

Fig. 2. Average run lengths versus mean detection delays for all compared algorithms. Algorithm 1: $\boldsymbol{d}_t$, Algorithm 2: $\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}$.

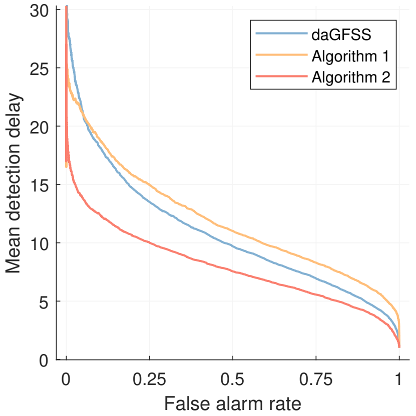

Fig. 3. Mean detection delays versus false alarm rates for all compared algorithms. Algorithm 1: $\boldsymbol{d}_t$, Algorithm 2: $\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}$.

Fig. 2 and Fig. 3 show the mean detection delays versus average run lengths and false alarm rates, respectively, of all detectors considered in this paper. As expected, Algorithm 2 clearly benefits from $\boldsymbol{d}_t$ compared to daGFSS, and $\hat{\boldsymbol{g}}$ in Algorithm 1. Fig. 4 illustrates the test statistics $\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}$, normalized here only for illustration convenience. This figure shows the ability of the proposed algorithm to localize the community where change points occur.

6. CONCLUSION

This paper introduces a distributed algorithm for detecting change points in streaming manifold-valued signals within an unknown community of a graph. The algorithm is online,

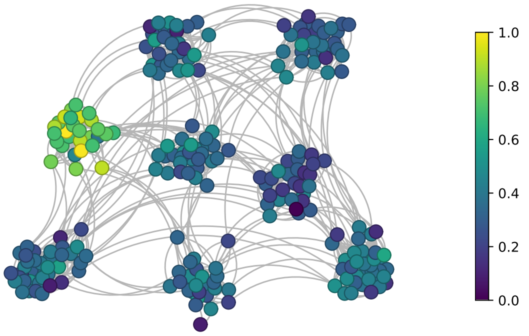

Fig. 4. Normalized $\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}$ at $t_r + 20$.

non-parametric, and fully distributed across the graph nodes. Simulation results validate its effectiveness in leveraging data manifold geometry and graph topology for improved performance.

7. REFERENCES

[1] 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*, 2013, vol. 31, pp. 545–553.

[2] A. Ferrari, C. Richard, and L. Verduci, "Distributed change detection in streaming graph signals," in *Proc. 8th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2019, pp. 166–170.

[3] A. Ferrari and C. Richard, "Non-parametric community change-points detection in streaming graph signals," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5545–5549.

[4] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001.

[5] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, pp. 395–416, 2007.

[6] 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.

[7] 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.

[8] A. Loukas, A. Simonetto, and G. Leus, "Distributed autoregressive moving average graph filters," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1931–1935, 2015.

[9] 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.

[10] R. A. Borsoi, C. Richard, A. Ferrari, J. Chen, and J. C. M. Bermudez, "Online graph-based change point detection in multiband image sequences," in *Proc. 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2020, pp. 850–854.

[11] X. Wang, R. Borsoi, and C. Richard, "Online change point detection on Riemannian manifolds with Karcher mean estimates," in *IEEE European Signal Processing Conference (EUSIPCO)*, 2023.

[12] X. Pennec, P. Fillard, and N. Ayache, "A riemannian framework for tensor computing," *International Journal of computer vision*, vol. 66, no. 1, pp. 41–66, 2006.

[13] N. Boumal, *An introduction to optimization on smooth manifolds*, Cambridge University Press, 2023.

[14] F. Bouchard, A. Mian, J. Zhou, S. Said, G. Ginolhac, and Y. Berthoumieu, "Riemannian geometry for compound gaussian distributions: Application to recursive change detection," *Signal Processing*, vol. 176, pp. 107716, 2020.

[15] P. Dubey and H.-G. Müller, "Fréchet change-point detection," *The Annals of Statistics*, vol. 48, no. 6, pp. 3312–3335, 2020.

[16] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Communications on pure and applied mathematics*, vol. 30, no. 5, pp. 509–541, 1977.

[17] S. Bonnabel, "Stochastic gradient descent on riemannian manifolds," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2217–2229, 2013.

[18] 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, pp. P10008, 2008.