# Détection de changements dans des signaux sur graphe à valeur dans des variétés riemanniennes

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

# Détection de changements dans des signaux sur graphe à valeur dans des variétés riemanniennes

Xiuheng WANG[1] Ricardo Augusto BORSOI[2] Cédric RICHARD[1] André FERRARI[1]

[1]Université Côte d'Azur, CNRS, OCA, Nice, France

[2]Université de Lorraine, CNRS, CRAN, Vandoeuvre-lès-Nancy, France

**Résumé –** Plusieurs méthodes de détection de changements dans des signaux sur graphe ont été proposées dans la littérature. Cependant, si elles exploitent à bon escient la topologie des graphes, elles se limitent au traitement de séries temporelles sur graphe dans des espaces euclidiens. Dans cet article, nous proposons une méthode de détection de changements dans des signaux sur graphe à valeur dans des variétés riemanniennes. Elle combine une statistique de test en chaque nœud qui tient compte de la géométrie des données sur la variété considérée, et un filtre sur graphe distribué qui exploite la topologie de celui-ci.

**Abstract –** Graph-based methods have been proposed to detect change points occurring in localized communities of networks. However, existing methods are limited to time series data in Euclidean domains. In this paper, we propose a graph-based change point detection method for streaming manifold-valued signals. The proposed framework combines a node-level test statistic, which accounts for the geometry of data on Riemannian manifolds, with a fully distributed graph filter that incorporates the network topology information.

## 1 Introduction

La détection des changements vise à identifier les moments où la distribution de probabilité d'un processus stochastique ou d'une série temporelle change. Elle constitue une tâche cruciale dans nombre de problèmes d'analyse de données en ligne. Récemment, de nouveaux travaux se sont intéressés à la détection de changements dans des séries temporelles sur graphe [1, 2, 3]. Ce problème a de nombreuses applications, dans les domaines de la physique, de la biologie et de la finance pour ne citer que ces quelques exemples.

Les changements observés dans ces problèmes interviennent souvent simultanément dans des groupes de nœuds fortement connectés du réseau ou graphe considéré, appelés aussi communautés. Les outils de traitement des signaux sur graphe, tels que les méthodes d'analyse spectrale [4, 5] ou de filtrage [6, 7, 8, 9], sont efficaces pour extraire des informations associant mesures et données relationnelles. Dans [1], les auteurs introduisent la méthode GFSS (Graph Fourier Scan Statistic). Elle exploite un filtre passe-bas sur graphe pour détecter les anomalies dans des communautés de signaux sur graphe. Le travail présenté dans [2] décrit une méthode de détection de changement en ligne exploitant les principes du GFSS, à la fois distribuée et adaptative, pour détecter les changements dans de grands réseaux. Cet algorithme est appliqué à la détection de changements dans des séquences d'images multicanaux dans [10]. En estimant un rapport de vraisemblance par des méthodes à noyau, la stratégie en ligne et distribuée développée dans [3] s'inscrit dans un cadre semi-paramétrique nécessitant peu d'hypothèses sur la distribution des données.

Toutes les méthodes de détection de changement sur graphe énumérées ci-dessus sont limitées à des séries temporelles dans des espaces euclidiens. À notre connaissance, il n'existe pas de généralisation d'algorithmes de ce type à des données sur graphe appartenant à des variétés riemanniennes et traitées en ligne. Une application caractéristique est la détection de changements dans des vidéos par une séquence de descripteurs de covariance locaux estimés sur les images [11]. Dans cette application, les changements peuvent facilement affecter plusieurs régions connexes de l'image, mais cette information n'est pas prise en compte par les algorithmes qui traitent chaque région séparément. Une possibilité pour exploiter cette information est de construire un graphe pour décrire la relation entre les régions, ce qui permet de détecter les points de changement de manière coopérative. Il convient de noter que pour développer des algorithmes efficaces de traitement de données sur variétés riemanniennes, il est nécessaire de tenir compte de la géométrie de l'espace grâce à des métriques appropriées.

Cet article présente une première approche pour la détection de changements dans des séries temporelles de données riemanniennes sur graphe. Cette méthode distribuée et en ligne repose sur une statistique de test en chaque nœud qui tient compte de la topologie du graphe. Le filtre sur graphe utilisé vise à améliorer les performances de détection des anomalies localisées dans des communautés de nœuds. Les résultats de simulation confirment la pertinence des principes développés.

## 2 Préalables

Dans cette section, nous introduisons quelques notions de géométrie riemannienne [12], que l'on illustre par la variété des matrices symétriques définies positives (SDP) de dimension $d \times d$, notée $\mathcal{S}_d^{++}$. Une *variété riemannienne* $(\mathcal{M}, g)$ est définie par une variété différentielle $\mathcal{M}$ et une métrique $g$. En chaque point $\boldsymbol{x}$ de $\mathcal{M}$, ils définissent un espace tangent $T_x\mathcal{M}$ et une métrique $g_x : T_x\mathcal{M} \times T_x\mathcal{M} \mapsto \mathbb{R}$ tels que $(T_x\mathcal{M}, g_x)$ est un espace euclidien. On note $d_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{y})$ la distance entre les points $\boldsymbol{x}$ et $\boldsymbol{y}$ de $\mathcal{M}$. Soit $\boldsymbol{v}$ appartenant à $T_x\mathcal{M}$ un vecteur tangent en $\boldsymbol{x}$ à la variété. L'application *exponentielle* $\boldsymbol{y} = \exp_x(\boldsymbol{v})$ définie le point $\boldsymbol{y}$ de $\mathcal{M}$ situé sur l'unique géodésique $\gamma_v(t)$ telle que $\gamma_v(0) = \boldsymbol{x}$, $\gamma_v'(0) = v$ et $\gamma_v(1) = \boldsymbol{y}$. Soit $f$ une fonction lisse de $\mathcal{M}$ dans $\mathbb{R}$. Le *gradient riemannien* de $f$ au point $\boldsymbol{x}$ est défini comme étant l'unique vecteur tangent $\nabla f(\boldsymbol{x})$ dans $T_x\mathcal{M}$ tel que $\frac{d}{dt}\big|_{t=0} f(\exp_x(t\boldsymbol{v})) = \langle \nabla f(\boldsymbol{x}), \boldsymbol{v} \rangle_x$ pour tout $\boldsymbol{v}$ de $T_x\mathcal{M}$. Soulignons enfin que l'application exponentielle peut être coûteuse à évaluer, ou numériquement peu stable pour certaines variétés. Il est possible de recourir à une rétraction $\boldsymbol{y} = R_x(\boldsymbol{v})$ si elle offre une alternative stable et efficace.

## 3 Formulation du problème

On considère un graphe non orienté $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ composé de $N$ sommets $\mathcal{N} = \{1, \ldots, N\}$ et $M$ arêtes $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ telles que $(i, j) \in \mathcal{E}$ si les nœuds $i$ et $j$ sont connectés. Le graphe $\mathcal{G}$ est associé à une matrice d'adjacence $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ dont chaque entrée $w_{ij} \geq 0$ représente le poids du lien entre les nœuds $i$ et $j$, avec $w_{ij} \neq 0$

si $(i,j) \in \mathcal{E}$. Une communauté $\mathcal{C}$ est un sous-ensemble de nœuds densément connectés de $\mathcal{G}$.

À chaque instant $t \in \mathbb{N}$, on observe un signal sur le graphe $\mathcal{X}_t = \{\boldsymbol{x}_t(n)\}_{n=1}^N$, avec $\boldsymbol{x}_t(n) \in \mathcal{M}$ désignant la mesure au nœud $n$ appartenant à une variété riemannienne $(\mathcal{M}, g)$. Dans le présent document, l'objectif est de détecter un changement dans le signal sur graphe $\mathcal{X}_t$ qui pourrait se produire à un instant inconnu $t_r \in \mathbb{N}$, appelé *instant de changement*. En particulier, on suppose que les changements se produisent dans une communauté inconnue $\mathcal{C}^*$ de $\mathcal{G}$ :

$$t < t_r : \boldsymbol{x}_t(n) \sim P_{0,n}, \quad t \geq t_r : \boldsymbol{x}_t(n) \sim P_{1,n}, \qquad (1)$$
$$\text{où } \forall n \in \mathcal{C}^*, P_{0,n} \neq P_{1,n}, \text{ et } \forall n \notin \mathcal{C}^*, P_{0,n} = P_{1,n}$$

avec $P_{0,n}$ et $P_{1,n}$ les densités de probabilité gouvernant $\boldsymbol{x}_t(n)$ avant et après l'instant de changement $t_r$. Afin de simplifier la présentation, le problème (1) ne fait référence qu'à un instant de changement. L'algorithme décrit ci-après peut cependant prendre en considération plusieurs changements.

# 4 Méthodologie

Les stratégies de détection de changements sur graphe [1, 2, 3] conçues pour des séries temporelles dans un espace vectoriel ne sont pas appropriées pour des données sur variété. Dans ce travail, nous proposons un nouveau cadre pour la détection de changements dans des signaux sur graphe issus de variétés riemanniennes. Tout d'abord, nous envisageons une stratégie en ligne opérant en chaque nœud sur des variétés riemanniennes afin de prendre en compte la géométrie des données. Ensuite, nous tirons parti de la topologie du graphe en filtrant les statistiques de test au niveau du graphe, sans compromettre l'appartenance des signaux à une variété. Enfin, le filtre sur graphe, centralisé, est mis en œuvre de manière distribuée, ce qui fournit une méthode efficace de détection de changement pour les grands réseaux.

## 4.1 Détection de changements sur variété

Certains algorithmes récents ont été conçus pour détecter en ligne des points de changement dans des données sur variété. Par exemple, dans [13], un algorithme en ligne paramétrique a été spécifiquement élaboré pour les distributions gaussiennes composites. Pour un autre problème, une technique hors ligne et non paramétrique est décrite dans [14]. Dans cette section, nous examinons un algorithme en ligne et non paramétrique déjà proposé dans [11], qui permet la détection de changements à partir de l'analyse de la *moyenne de Karcher*. Soit un signal aléatoire $\boldsymbol{x} \in \mathcal{M}$ distribué selon une loi $P$. Sa moyenne de Karcher $\boldsymbol{m} \in \mathcal{M}$ est définie par :

$$\boldsymbol{m}_{\text{Karcher}} = \arg\min_{\boldsymbol{m}} \left\{ f(\boldsymbol{m}) \triangleq \mathsf{E}[d^2_{\mathcal{M}}(\boldsymbol{m}, \boldsymbol{x})] \right\}. \qquad (2)$$

La méthode décrite dans [11], qui s'inspire de la stratégie originelle NEWMA [15], détecte les changements en comparant deux moyennes de Karcher estimées à l'aide d'un algorithme d'optimisation stochastique où, à l'instant $t$, le gradient de $f(\boldsymbol{m})$ est remplacé par son estimation instantanée en $\boldsymbol{x}_t(n)$. Plus précisément, en chaque nœud $n$, deux moyennes de Karcher sont estimées à l'aide de l'algorithme stochastique riemannien [16], avec deux pas de gradient distincts $\lambda < \Lambda$ de la manière suivante :

$$\boldsymbol{m}_{\lambda,t}(n) = R_{\boldsymbol{m}_{\lambda,t-1}(n)}\big(-\lambda H(\boldsymbol{m}_{\lambda,t-1}(n), \boldsymbol{x}_t(n))\big), \qquad (3)$$
$$\boldsymbol{m}_{\Lambda,t}(n) = R_{\boldsymbol{m}_{\Lambda,t-1}(n)}\big(-\Lambda H(\boldsymbol{m}_{\Lambda,t-1}(n), \boldsymbol{x}_t(n))\big), \qquad (4)$$

où $R_{\boldsymbol{m}}$ est une rétraction en $\boldsymbol{m}$, et $H(\boldsymbol{m}, \boldsymbol{x})$ représente le gradient stochastique riemannien de (2) estimé au point $\boldsymbol{m}$ à partir de la nouvelle observation $\boldsymbol{x}$. La contrainte $\lambda < \Lambda$ signifie donc que $\boldsymbol{m}_{\lambda,t}(n)$ est plus susceptible de s'adapter aux nouvelles données tandis que $\boldsymbol{m}_{\Lambda,t}(n)$ estime la tendance à long terme de la série.

**Algorithme 1 :** Procédure en chaque nœud

**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 (3) and (4)
    Compute $d_t(n)$ using (5)
  **end**
  **if** $\exists\, n \in \mathcal{N} : d_t(n) > \xi$ **then**
    Flag $t$ as a change point
  **end**
**end**

En comparant $\boldsymbol{m}_{\lambda,t}(n)$ et $\boldsymbol{m}_{\Lambda,t}(n)$ à l'aide d'une distance riemannienne sur $\mathcal{M}$, une statistique $d_t(n)$ pour la détection de changement en chaque nœud $n$ peut être élaborée ainsi :

$$d_t(n) = d_{\mathcal{M}}(\boldsymbol{m}_{\lambda,t}(n), \boldsymbol{m}_{\Lambda,t}(n)). \qquad (5)$$

La détection de changement au niveau de chaque nœud peut alors être réalisée en comparant $d_t(n)$ à un seuil $\xi$. La procédure correspondante est résumée par l'Algorithme 1.

À titre d'exemple, la distance géodésique sur $\mathcal{S}_d^{++}$ entre 2 matrices SDP $\boldsymbol{\Sigma}_1$ et $\boldsymbol{\Sigma}_2$ est donnée par [17] :

$$d_{\mathcal{S}_d^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = \left\| \log(\boldsymbol{\Sigma}_2^{-\frac{1}{2}} \boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_2^{-\frac{1}{2}}) \right\|_F, \qquad (6)$$

où $\|\cdot\|_F$ est la norme de Frobenius. Dans ce cas, le gradient riemannien de $d^2_{\mathcal{S}_d^{++}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2)$ par rapport à l'une ou l'autre de ses variables s'obtient en appliquant la transformation $\boldsymbol{\Sigma}_1 \boldsymbol{G} \boldsymbol{\Sigma}_1$ au gradient euclidien $\boldsymbol{G}$, voir par exemple [18, théorème 6.3.3], ce qui conduit à :

$$H(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = 2 \log\left(\boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_2^{-1}\right) \boldsymbol{\Sigma}_1 \qquad (7)$$

Enfin, la rétraction suivante est une approximation de second ordre de l'application exponentielle sur $\mathcal{S}_d^{++}$ :

$$R_{\boldsymbol{\Sigma}, \mathcal{S}_d^{++}}(\boldsymbol{\xi}) = \boldsymbol{\Sigma} + \boldsymbol{\xi} + \frac{1}{2} \boldsymbol{\xi} \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi}. \qquad (8)$$

Les équations (7)-(8) appliquées à (3)-(4), dont on compare les sorties $\boldsymbol{\Sigma}_{\lambda,t}(n)$ et $\boldsymbol{\Sigma}_{\Lambda,t}(n)$ grâce à la distance (6) que l'on compare à un seuil, mène à un algorithme de détection de changements dans une série temporelle de matrices SDP.

## 4.2 Détection de changements sur graphe

Soit $\boldsymbol{d}_t = [d_t(1), \ldots, d_t(N)]^\top$ les statistiques collectées en chaque nœud à l'instant $t$ calculées par (1), qui ne tiennent pas compte de la topologie du graphe. Pour améliorer la détection de changements affectant des communautés, inconnues a priori, on propose de recourir à un filtre sur graphe reposant sur le GFSS présenté dans [1]. Le GFSS vise à tester si un signal statique sur graphe à valeurs scalaires en chaque nœud est à moyenne nulle, contre l'hypothèse qu'il existe une communauté de nœuds connectés dont le signal sur graphe local a une moyenne différente de zéro. Plus précisément, on propose d'appliquer le GFSS aux statistiques de test $\boldsymbol{d}_t$ définies dans (5) plutôt qu'aux signaux originaux $\boldsymbol{x}_t$ afin de préserver les informations sur variété du problème (1).

Soit $\boldsymbol{L}$ le laplacien normalisé du graphe $\mathcal{G}$, et $\{\boldsymbol{u}_n\}_{n=1}^N$ une base de vecteurs propres orthonormée de $\boldsymbol{L}$ avec $\{\mu_n\}_{n=1}^N$ les valeurs propres associées. Étant donné les statistiques de test calculées au niveau des nœuds $\boldsymbol{d}_t$ de $\mathcal{G}$, le GFSS est défini par :

$$t_{\text{GFSS}}(\boldsymbol{d}_t) = \|\boldsymbol{g}_{\boldsymbol{d}_t}\|_2 \qquad (9)$$

$$\text{avec } \boldsymbol{g}_{\boldsymbol{d}_t} = \sum_{n=2}^{N} h^*(\mu_n)(\boldsymbol{u}_n^\top \boldsymbol{d}_t)\boldsymbol{u}_n \qquad (10)$$

**Algorithme 2 :** Détection distribuée sur graphe

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

où $\boldsymbol{g}_{\boldsymbol{d}_t}$ est la statistique $\boldsymbol{d}_t$ filtrée par le graphe, et $h^*(\mu)$ est la réponse en fréquence du filtre définie par [1] :

$$h^*(\mu) = \min\left\{1, \sqrt{\frac{\gamma}{\mu}}\right\},\ \mu > 0, \tag{11}$$

dont $\gamma > 0$ est un paramètre de réglage.

Pour mieux comprendre la procédure de filtrage dans (10), rappelons le rôle des vecteurs propres $\boldsymbol{u}_n$ de la matrice laplacienne du graphe $\boldsymbol{L}$ dans le *clustering* spectral [4, 5]. On considère le cas idéal d'un graphe avec $1 < k < N$ composantes totalement connectées $\mathcal{C}_k$, mais déconnectées les unes des autres. Chaque vecteur propre $\boldsymbol{u}_n$ est proportionnel à la fonction indicatrice de $\mathcal{C}_n$, et $\boldsymbol{u}_n^\top \boldsymbol{d}_t$ est proportionnel à la somme des composantes de $\boldsymbol{d}_t$ correspondant à $\mathcal{C}_n$. Par conséquent, $(\boldsymbol{u}_n^\top \boldsymbol{d}_t)\boldsymbol{u}_n$ dans (10) attribue la valeur moyenne des composantes $d_t(k)$ pour $k \in \mathcal{C}_n$ à chaque nœud $k$ de $\mathcal{C}_n$. Le nombre de communautés $k$ étant inconnu, la réponse du filtre dans (11) est conçue pour pénaliser un trop grand nombre de clusters ou communautés dans (10). Lorsque ces communautés $\mathcal{C}_n$ sont connectées par quelques arêtes, voir Fig. 1 par exemple, on admet que cette analyse demeure valable. Il convient de noter que cette hypothèse est également la pierre angulaire des méthodes de regroupement spectral [4, 5].

### 4.3 Implémentation distribuée

L'opération de filtrage définie dans (10) nécessite la décomposition en vecteurs et valeurs propres de la matrice laplacienne normalisée $\boldsymbol{L}$. Ce calcul peut s'avérer coûteux et être inapproprié pour de grands réseaux. Une stratégie permettant un passage à l'échelle de notre algorithme de détection de changement sur graphe vise à remplacer le filtre (10)-(11) par un filtre qui peut être distribué sur les sommets du graphe [6, 7]. Contrairement aux filtres à réponse impulsionnelle finie du type (10), un filtre sur graphe autorégressif à moyenne mobile a été proposé dans [8, 9]. Ce filtre agrège récursivement les signaux dans le voisinage de chaque nœud, ce qui entraîne de faibles coûts de calcul et de mémoire.

Nous proposons de mettre en œuvre un filtre ARMA$_K$ sur graphe [9], qui peut être distribué, afin d'approximer le GFSS défini dans (10) :

$$\boldsymbol{y}_{\ell,t} = \psi_\ell \boldsymbol{L} \boldsymbol{y}_{\ell,t-1} + \varphi_\ell \boldsymbol{d}_t,\ \boldsymbol{y}_{\ell,-1} = \mathbf{0},\ \forall \ell = 1 \ldots K, \tag{12}$$

$$\hat{\boldsymbol{g}}_{\boldsymbol{d}_t} = \sum_{\ell=1}^{K} \boldsymbol{y}_{\ell,t} + c\boldsymbol{d}_t \tag{13}$$

L'opération $\boldsymbol{L}\boldsymbol{y}_{\ell,t-1}$ est une étape de combinaison sur graphe qui peut être réalisée localement en chaque nœud $n$ en combinant linéairement des statistiques de test voisines sous la forme $\sum_{k \in \mathcal{N}_p} \boldsymbol{L}_{pk}\, y_{\ell,t-1,k}$, où $\mathcal{N}_p$ désigne le voisinage du nœud $p$, comprenant $p$ lui-même, $y_{\ell,t-1,k}$ est le $k$-ième élément de $\boldsymbol{y}_{\ell,t-1}$ et $\boldsymbol{L}_{pk}$ le $(p,k)$-ième élément de $\boldsymbol{L}$. Cette opération joue un rôle central dans les procédures de filtrage sur graphe distribuées car elle n'implique que les mesures de nœuds voisins. Notons que les paramètres $c$ et $\{(\psi_\ell, \varphi_\ell)\}_{\ell=1\ldots K}$ doivent être synthétisés de sorte que $h(\mu)$ se rapproche de la réponse spectrale attendue $h^*(\mu)$ dans (11).

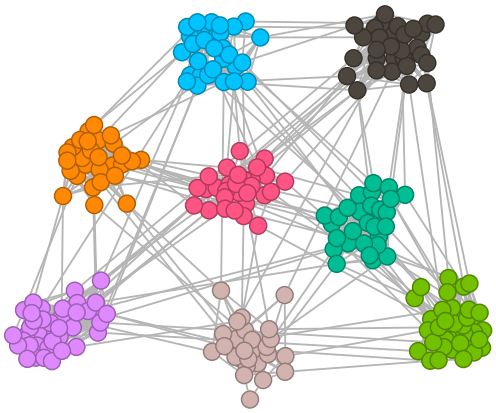

FIGURE 1 : Topologie du graphe et des communautés.

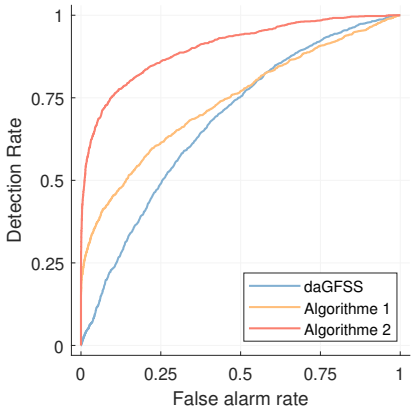


FIGURE 2 : Comparaison des courbes COR : daGFSS (euclidien, avec filtrage GFSS), Algorithme 1 (sur variété, sans filtrage GFSS), Algorithme 2 (proposé ici).

## 5 Simulations

Dans cette section, nous illustrons les performances de l'approche proposée en utilisant des signaux sur graphe $\boldsymbol{x}_t(n)$ appartenant à la variété $\mathcal{S}_d^{++}$. La topologie du graphe $\mathcal{G}$ utilisé pour les simulations est illustrée par la Fig. 1. Le graphe est composé de $p = 250$ nœuds et $m = 2508$ arêtes, divisés en 8 communautés. Ces communautés $\mathcal{C}_i$ ont été représentées à l'aide de [19] et colorées pour la visualisation. Nous avons généré $\mathcal{X}_t$ comme dans (1) avec un changement sur une communauté $\mathcal{C}^*$, dont les nœuds sont colorés en orange. Les matrices synthétiques $\boldsymbol{\Sigma}_t \in \mathcal{S}_d^{++}$ avec $d = 6$ ont été échantillonnées à partir d'une distribution de Wishart $\mathcal{W}_6(\boldsymbol{V}, 6)$. Nous avons généré 800 échantillons et fixé un point de changement à $t_r = 500$ dans (1) où l'on a réinitialisé $\boldsymbol{V}$.

À partir de $\{\boldsymbol{\Sigma}_t\}_{t \in \mathbb{N}}$ et de la métrique (6), les moyennes de Karcher ont été estimées en minimisant la fonction objectif (2) en ligne avec les algorithmes de gradient stochastique riemanniens (3)-(4), de pas $\lambda = 0.01$ et $\lambda = 0.02$. Les moyennes estimées ont été utilisées pour calculer la statistique de test en ligne (5). Le paramètre $\gamma$ du filtre GFSS (11) a été fixé à 0,03 et $K$ dans le filtre ARMA$_K$ à 4. La synthèse du filtre par l'estimation des paramètres $c$ et $(\psi_\ell, \varphi_\ell)_{\ell=1,\ldots,K}$ est discutée en pratique dans [2].

Pour illustrer l'intérêt de prendre en compte à la fois la géométrie de la variété et la topologie du graphe, nous avons comparé l'algo-

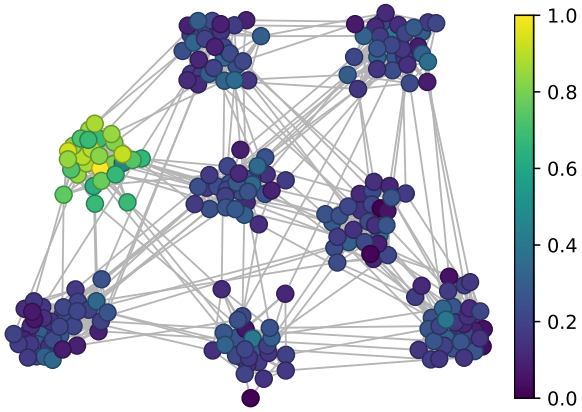


FIGURE 3 : Représentation normalisée de $\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}$ à $t_r + 8$.

rithme 2 à deux autres stratégies. La première est le daGFSS [2], qui équivaux à l'algorithme développé ici mais conçu dans le cas euclidien. Celui-ci a été appliqué ici après vectorisation des parties triangulaires inférieures et diagonale de $\boldsymbol{\Sigma}_t$. La seconde est la méthode de détection de changement basée sur les moyennes de Karcher, détaillée dans l'Algorithme 1, exécutée nœud par nœud sans filtrage GFSS.

Pour comparer les performances de détection de ces trois algorithmes, des simulations de Monte Carlo ont été réalisées afin d'estimer les courbes ROC. Les taux de détection et de fausse alarme sont définis, par exemple ici pour $\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}$, comme suit :

$$\mathbf{P}_{\text{d}} = \text{Prob}\big(\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}(n) > \xi \,|\, t > t_r, n \in \mathcal{C}^*\big), \tag{14}$$

$$\mathbf{P}_{\text{fa}} = \text{Prob}\big(\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}(n) > \xi \,|\, t > t_r, n \notin \mathcal{C}^*\big). \tag{15}$$

La figure 2 montre les courbes ROC des 3 détecteurs considérés. Comme prévu, l'algorithme 2 bénéficie clairement de la prise en compte de la géométrie par rapport à daGFSS, et de la prise en compte de la topologie par rapport à 1.

La figure 3 illustre les statistiques de test $\hat{\boldsymbol{g}}_{\boldsymbol{d}_t}$ normalisées. Elle montre la capacité de l'algorithme proposé à localiser la communauté où se produisent les points de changement.

# 6 Conclusion

Cet article a proposé un algorithme sur graphe pour la détection de changements dans des séries temporelles sur variété riemannienne, lorsqu'ils sont localisés en particulier dans des communautés. La procédure de détection proposée opère en ligne, selon une stratégie non paramétrique, et entièrement distribuée sur les nœuds du graphe. Les résultats de simulation en confirment les qualités.

# Références


[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*, vol. 31, 2013, 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," 2023, submitted.

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

[13] 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, p. 107716, 2020.

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

[15] N. Keriven, D. Garreau, and I. Poli, "Newma : a new method for scalable model-free online change-point detection," *IEEE Transactions on Signal Processing*, vol. 68, 2020.

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

[17] 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.

[18] R. Bhatia, *Positive definite matrices*, ser. Princeton series in applied mathematics. Princeton University Press, 2007.

[19] 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.