Apprentissage distribué de la topologie d'un graphe à partir de signaux temporels sur graphe

Mircea MOSCU[1], Roula NASSIF[2], Fei HUA[3], Cédric RICHARD[1]

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

[2]École Polytechnique Fédérale de Lausanne, Suisse

[3]School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, Chine

mircea.moscu@oca.eu, roula.nassif@epfl.ch, fei.hua@oca.eu, cedric.richard@unice.fr

Résumé – Cet article concerne l'estimation de la structure d'un graphe à partir de signaux temporels sur graphe, en capturant les relations sous-jacentes aux observations acquises en chaque nœud, sous la forme d'une matrice d'adjacence parcimonieuse éventuellement symétrique. Il existe de nombreuses solutions hors ligne pour résoudre ce problème, mais peu prennent en compte la nature distribuée des réseaux. Nous partons d'un cadre centralisé et, par des relaxations appropriées, aboutissons à un algorithme d'apprentissage en ligne distribué. Notre algorithme est testé expérimentalement et comparé avec succès à une solution centralisée hors ligne de la littérature.

Abstract – This paper focuses on estimating a network structure capturing the dependencies among streaming signals in the form of a possibly symmetric or sparse matrix. Several offline solutions to address this problem exist, but do not pay much attention to the distributed nature of networks. We start from a centralized setting and then introduce a simple yet powerful data model under sparsity assumptions, aimed to infer network connections from streaming data in a distributed and online setting. Our algorithm is tested experimentally and successfully compared to a centralized offline solution of the literature.

1 Introduction

Au cours de la dernière décennie, les données sont devenues une ressource brute qui nécessite d'être affinée afin de devenir une information utile. Elles prennent différentes formes et proviennent de différentes sources : commerce électronique, événements sportifs, médias et interactions sociales, pour n'en nommer que quelques-unes. Les données structurées, où chaque composante est liée à d'autres composantes, sont omniprésentes et évoluent au cours du temps, ce qui les rend difficiles à traiter et à analyser. Depuis des travaux précurseurs tels que [14], le traitement du signal sur graphe a attiré une attention grandissante en raison des multiples applications potentielles qu'il offre. L'imagerie cérébrale, l'analyse des réseaux sociaux et des réseaux de transport en sont des exemples caractéristiques.

La plupart des algorithmes de traitement du signal sur graphe introduits au cours des cinq dernières années supposent une connaissance préalable de la structure du graphe. Cependant, il y a des situations où le graphe n'est pas aisément accessible et doit être inféré à partir des données. Cet article s'intéresse à l'estimation de la structure d'un graphe à partir de flux de données en capturant les relations sous-jacentes aux observations acquises en chaque nœud, sous la forme d'une matrice d'adjacence éventuellement dirigée et pondérée.

Définitions : Un graphe $\mathcal{G}$ est constitué d'un ensemble $\mathcal{N}$ de N nœuds, et d'un ensemble $\mathcal{E}$ d'arêtes tel que si les nœuds m et n sont liés, alors $(m, n) \in \mathcal{E}$. Pour les graphes non-dirigés, ces paires de nœuds ne sont pas ordonnées. Soit $\mathcal{N}_n$ le voisinage du nœud n, c.à.d., $\mathcal{N}_n = \{m : (m, n) \in \mathcal{E}\}$.

Un signal réel $\boldsymbol{x} \triangleq [x_1, \ldots, x_N]^\top$ est collecté au niveau des nœuds, où x_n est l'échantillon du signal $\boldsymbol{x}$ au nœud n. L'opérateur de translation [14] sur $\mathcal{G}$ est défini comme une matrice de taille $N \times N$ et noté $\boldsymbol{S}$. Il décrit les interactions entre nœuds et, par extension, est utilisé pour représenter des relations entre données. La composante s_{nm} de $\boldsymbol{S}$ est non nulle si $(m, n) \in \mathcal{E}$. Des choix possibles pour cet opérateur sont la matrice d'adjacence (pondérée ou pas) ou la matrice Laplacienne (et ses variations) [1]. L'opération $\boldsymbol{S}^k\boldsymbol{x}$ opère une translation (pondérée) de k sauts des données sur le graphe.

Travaux antérieurs : De nombreux travaux ont été proposés pour l'estimation de la topologie d'un graphe à partir de signaux sur graphe. Une proposition très précoce est dans [3], où une méthode basée sur l'estimation de covariance pour déduire les liens est introduite. De la même façon, dans [5], le graphical Lasso est utilisé pour estimer l'inverse de la matrice de covariance à partir des données. Dans [12], les auteurs estiment la connectivité d'un graphe à partir de modèles spectraux, en supposant que le signal sur graphe $\boldsymbol{x}$ est stationnaire et généré par un processus de diffusion. Ils estiment un opérateur de translation $\boldsymbol{S}$ satisfaisant un ensemble de contraintes telles que la parcimonie et la symétrie. Les auteurs de [15] proposent un algorithme adaptatif pour apprendre la topologie d'un graphe à partir de signaux dynamiques sur graphe engendrés par un processus de diffusion. Dans [13], des noyaux sont utilisés avec un

modèle autorégressif pour le suivi de signaux de connectivité cérébraux au cours du temps. L'approche par noyaux multiples de [16] utilise des corrélations partielles pour accéder à la topologie du graphe, et une régression en norme ℓ_p pour améliorer les performances. Un état de l'art sur l'inférence de topologie est disponible dans [4]. Contrairement aux méthodes existantes, cet article se concentre sur l'identification de la topologie d'un graphe à partir de signaux temporels sur graphe d'une manière distribuée et en ligne (adaptative).

Notations : Les lettres minuscules désignent les scalaires, et les lettres minuscules et majuscules en caractères gras représentent respectivement des vecteurs et des matrices. Les caractères calligraphiques majuscules désignent des ensembles. On note $|\mathcal{N}|$ le cardinal d'un ensemble $\mathcal{N}$. L'opérateur signe est noté $\mathrm{sgn}\{\cdot\}$ et est appliqué élément par élément d'une matrice.

2 Formulation du problème centralisé

Un filtre sur graphe prend un signal sur graphe $\boldsymbol{x}(i)$ en entrée, et fournit un signal $\boldsymbol{y}(i)$ en sortie donné par $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}$ et indexé par le même graphe. Une forme souvent retenue pour $\boldsymbol{H}$ est le filtre d'ordre K invariant par translation, défini par [11] :

$$\boldsymbol{y}(i) = \sum_{k=0}^{K-1} h_k \boldsymbol{S}^k \boldsymbol{x}(i), \quad i \geq 0, \tag{1}$$

avec $\boldsymbol{S}$ la matrice de translation et $\{h_k\}_{k=0}^{K-1}$ les coefficients du filtre. Notons que ce modèle suppose la diffusion instantanée de l'information, ce qui en est une limitation. Le modèle dynamique suivant s'affranchit de cette restriction [8, 9] :

$$\boldsymbol{y}(i) = \sum_{k=0}^{K-1} h_k \boldsymbol{S}^k \boldsymbol{x}(i-k), \qquad i \geq K-1. \tag{2}$$

En supposant que la matrice de translation $\boldsymbol{S}$ est connue, les auteurs dans [6, 9] montrent comment les stratégies de diffusion peuvent être appliquées pour estimer les coefficients $\{h_k\}_{k=0}^{K-1}$ à partir de signaux temporels sur graphe $\{(\boldsymbol{x}(i), \boldsymbol{y}(i))\}$.

Un outil possible pour l'inférence d'une topologie causale d'un réseau est le modèle autorégressif multivarié suivant [8] :

$$\boldsymbol{y}(i) = \sum_{k=0}^{K-1} \boldsymbol{S}_{0:k} \boldsymbol{x}(i-k) + \boldsymbol{v}(i), \quad i \geq K-1 \tag{3}$$

où $\boldsymbol{S}_{a:b} \triangleq \prod_{k=a}^{b} \boldsymbol{S}_k$ a pour composantes $\{s_{nm,b-a}\}$ décrivant l'influence du nœud m sur le nœud n à une distance de $b-a$ sauts, et $\boldsymbol{v}(i)$ est un bruit d'innovation. On note que $\boldsymbol{S}_0 = \boldsymbol{I}$ et $\boldsymbol{S}_1 = \boldsymbol{S}$. Ce modèle est utile pour évaluer la causalité au sens de Granger, où l'on dit que x_m influe sur x_ℓ si la connaissance du premier améliore la prédiction du second [2].

Nous supposons que le signal $\boldsymbol{x}(i)$ est stationnaire à sens large et de moyenne nulle, c.à.d. que la séquence de corrélation $\boldsymbol{R}_x(k) = \mathbb{E}\{\boldsymbol{x}(i)\boldsymbol{x}^\top(i-k)\}$ est une fonction qui dépend du décalage temporel k. Le bruit $\boldsymbol{v}(i)$ est supposé de moyenne nulle, i.i.d., de covariance $\boldsymbol{R}_v = \mathrm{diag}\{\sigma_{v,n}\}_{n=1}^N$. Sous ces hypothèses, $\boldsymbol{S}$ dans (3) peut être estimé en résolvant :

$$\begin{aligned} \boldsymbol{S}^* = \underset{\boldsymbol{S}}{\mathrm{argmin}}\ & \mathbb{E}\Big\|\boldsymbol{y}(i) - \sum_{k=0}^{K-1} \boldsymbol{S}_{0:k}\boldsymbol{x}(i-k)\Big\|^2 + \eta\,\Phi(\boldsymbol{S}) \\ \text{s.c. } & s_{nm} = 0 \ \text{ si } \ m \notin \mathcal{N}_n, \quad n = 1, \ldots, N \end{aligned} \tag{4}$$

avec $\eta > 0$. La fonction coût (4) inclut un terme de régularisation $\Phi(\boldsymbol{S})$ pour tenir compte d'informations a priori sur $\boldsymbol{S}$ telles que la symétrie ou la parcimonie. Les contraintes forçant les entrées s_{nm} de $\boldsymbol{S}$ à zéro correspondent aux paires $(n, m) \notin \mathcal{E}$. Le problème (4) est non convexe compte tenu de son caractère polynomial en $\boldsymbol{S}$, ce qui induit pour les algorithmes de résolution de converger éventuellement vers un minimum local plutôt que vers un minimum global.

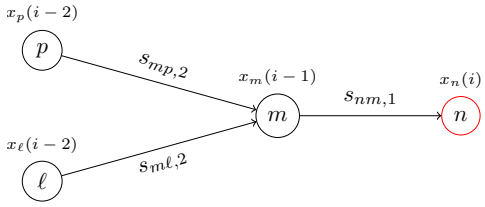


FIGURE 1 – Chemins des données pondérées vers le nœud n représentés sous forme d'arêtes dirigés. Pour estimer le poids $s_{nm,1}$, le nœud n reçoit de son voisin m le vecteur à $(K-1)$ éléments $[x_m(i-1), s_{mp,2}x_p(i-2) + s_{m\ell,2}x_\ell(i-2)]^\top$.

3 Solution distribuée

Selon (3), la sortie $y_n(i)$ en chaque nœud n est donnée par :

$$y_n(i) = \sum_{k=0}^{K-1} \Big[\boldsymbol{S}_{0:k}\boldsymbol{x}(i-k)\Big]_n + v_n(i) \tag{5}$$

Ceci peut être réécrit comme :

$$\begin{aligned} \bar{y}_n(i) \triangleq y_n(i) - x_n(i) = \boldsymbol{s}_n^\top [\boldsymbol{x}(i-1)]_{m\in\mathcal{N}_n} + \ldots + \\ \boldsymbol{s}_n^\top [\boldsymbol{S}_{2:K-1}\boldsymbol{x}(i-K+1)]_{m\in\mathcal{N}_n} + v_n(i), \end{aligned} \tag{6}$$

avec

$$\boldsymbol{s}_n = \mathrm{col}\{s_{nm} \colon m \in \mathcal{N}_n\} \tag{7}$$

le vecteur $|\mathcal{N}_n| \times 1$ agrégeant toutes les entrées non nulles de la n^{e} ligne de $\boldsymbol{S}$. Il s'en suit que (6) peut être exprimé comme :

$$\bar{y}_n(i) = \boldsymbol{z}_n^\top(i)\boldsymbol{s}_n + v_n(i) \tag{8}$$

où $\boldsymbol{z}_n(i)$ est un vecteur colonne $|\mathcal{N}_n| \times 1$ défini par :

$$\boldsymbol{z}_n(i) = [\boldsymbol{x}(i-1)]_{m\in\mathcal{N}_n} + \sum_{k=2}^{K-1} \Big[\boldsymbol{S}_{2:k}\boldsymbol{x}(i-k)\Big]_{m\in\mathcal{N}_n} \tag{9}$$

À l'instant i, le nœud n pondère les données entrantes de ses voisins par les entrées correspondantes de la n^{e} ligne de $\boldsymbol{S}$. Le même raisonnement s'applique à tout nœud m voisin de n, voir Fig. 1, qui pondère ses propres données entrantes avec des entrées de la m^{e} ligne de $\boldsymbol{S}$. Cela signifie que les données à

deux sauts envoyées par le nœud ℓ à l'instant $i-2$, passant par le nœud m à l'instant $i-1$, et reçues par le nœud n à l'instant i, sont successivement pondérées par $s_{m\ell}$ et s_{nm}. Par conséquent, lors de l'estimation de $\boldsymbol{S}$, le nœud n peut simplement se concentrer sur ses propres poids stockés dans la n^{e} ligne de $\boldsymbol{S}$ à condition que tous les autres nœuds du réseau en fasse de même avec leurs propres poids. La formulation (7)–(9) présente plusieurs avantages par rapport à la solution centralisée de [8]. Il s'agit en particulier d'une relaxation convexe du problème (4) se prêtant à une résolution locale.

Pour résoudre ce problème, on définit le coût global :

$$J(\boldsymbol{S}) = \sum_{n=1}^{N} J_n(\boldsymbol{s}_n) \tag{10}$$

où $J_n(\boldsymbol{s}_n)$ désigne le coût au nœud n, à savoir :

$$J_n(\boldsymbol{s}_n) \triangleq \mathbb{E}\|\bar{y}_n(i) - \boldsymbol{z}_n^\top(i)\boldsymbol{s}_n\|^2 + \eta_n \Phi(\boldsymbol{s}_n). \tag{11}$$

Cette formulation permet à chaque nœud n d'estimer ses propres entrées $\boldsymbol{s}_n$ de $\boldsymbol{S}$, et éventuellement de tenir compte de certaines connaissances a priori sur $\boldsymbol{S}$ via $\Phi(\boldsymbol{s}_n)$.

Dans cet article, nous imposons une contrainte de parcimonie sur $\boldsymbol{S}$. Ceci permet, en l'absence d'information sur la topologie du réseau, de déterminer le voisinage $\mathcal{N}_n$ de tout nœud n à partir de $\mathcal{N}$. Pour se faire, nous introduisons les régularisations ZA (*zero-attracting*) :

$$\Phi_{ZA}(\boldsymbol{s}_n) = \sum_{m \in \mathcal{N}\setminus\{n\}} |s_{nm}| \tag{12}$$

et RZA(*reweighted zero-attracting*) :

$$\Phi_{RZA}(\boldsymbol{s}_n) = \sum_{m \in \mathcal{N}\setminus\{n\}} \log\left(1 + \frac{|s_{nm}|}{\epsilon}\right), \epsilon > 0. \tag{13}$$

Pour minimiser (11), nous proposons une solution incrémentale basée sur une descente de gradient, à savoir [7] :

$$\boldsymbol{s}_n(i+1) = \boldsymbol{s}_n(i) + \mu_n[\boldsymbol{r}_{z_n y} - \boldsymbol{R}_{z_n}\boldsymbol{s}_n(i) - \eta_n \text{sgn}\{\boldsymbol{s}_n(i)\}], \tag{14a}$$

$$\boldsymbol{s}_n(i+1) = \boldsymbol{s}_n(i) + \mu_n\left[\boldsymbol{r}_{z_n y} - \boldsymbol{R}_{z_n}\boldsymbol{s}_n(i) - \eta_n \frac{\text{sgn}\{\boldsymbol{s}_n(i)\}}{\boldsymbol{\epsilon} + |\boldsymbol{s}_n(i)|}\right] \tag{14b}$$

pour ZA et RZA respectivement. Cependant, comme les moments de second ordre de (14) sont rarement disponibles, une stratégie de descente de gradient stochastique est employée, avec les approximations instantanées suivantes :

$$\boldsymbol{R}_{z_n} \approx \boldsymbol{z}_n(i)\boldsymbol{z}_n^\top(i) \;\text{ et }\; \boldsymbol{r}_{z_n y} \approx \boldsymbol{z}_n(i)\bar{y}_n(i). \tag{15}$$

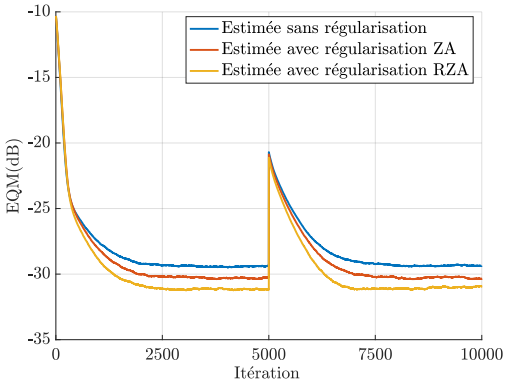
L'algorithme en chaque nœud n est résumé ci-dessous.

4 Expérimentations

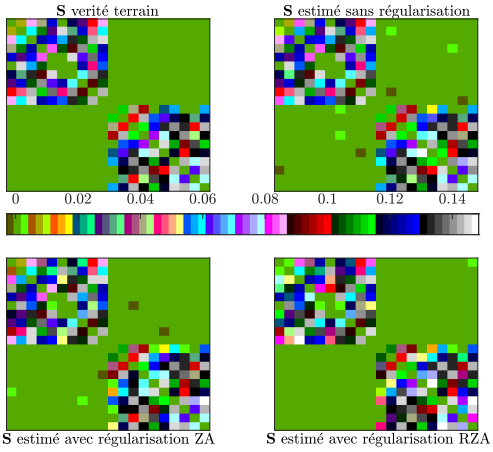
Un graphe de type communauté non-dirigé a été généré en utilisant GSPBOX [10], avec $N = 20$ nœuds formant deux communautés. L'opérateur de translation sur graphe $\boldsymbol{S}$ a été défini par $\boldsymbol{W}/[1.1 \cdot \lambda_{\max}(\boldsymbol{W})]$, la matrice d'adjacence pondérée normalisée. En notant $\boldsymbol{c}_n$ les coordonnées de l'intégration 2D du nœud n, les poids w_{mn} ont été fixés à $\exp(-\gamma \|\boldsymbol{c}_m - \boldsymbol{c}_n\|^2)$. Le signal $\boldsymbol{x}(i)$ a été généré selon une loi normale de moyenne nulle et de matrice de covariance $\boldsymbol{R}_x$ solution de l'équation de Lyapunov $\boldsymbol{S}\boldsymbol{R}_x\boldsymbol{S}^\top - \boldsymbol{R}_x + \boldsymbol{I} = \boldsymbol{0}$. Le bruit $\boldsymbol{v}(i)$ a été généré selon une loi normale de moyenne nulle et de matrice de covariance $\boldsymbol{R}_v = \text{diag}\{\sigma_{v,n}^2\}_{n=1}^N$. Les variances $\sigma_{v,n}^2$ ont été générées selon une distribution uniforme $\mathcal{U}(0.1, 0.15)$. L'ordre du filtre a été fixé à $K = 3$. La sortie $\boldsymbol{y}(i)$ a été générée avec le modèle (3). Nous avons utilisé un pas μ constant pour tous les nœuds, et les paramètres η_n et ϵ ont été fixés au cours de 50 passes de Monte-Carlo.

Algorithme 1 : Estimation locale de la topologie

Entrées : Paramètres μ_n et η_n
Initialisation : Initialiser toutes les entrées de $\boldsymbol{s}_n(0)$
Algorithme : À chaque instant $i \geq 1$:
Recevoir les données pondérées $[\boldsymbol{S}_{k-1}\boldsymbol{x}(i-k)]_{m\in\mathcal{N}_n}$
Calculer le régresseur $\boldsymbol{z}_n(i)$ avec (9)
Mettre à jour l'estimation locale $\boldsymbol{s}_n$ avec (14) ou (15)



(a) EQM



(b) Matrice de translation et estimées

FIGURE 2 – (a) Courbes EQM. (b) Matrice $\boldsymbol{S}$ et ses estimées.

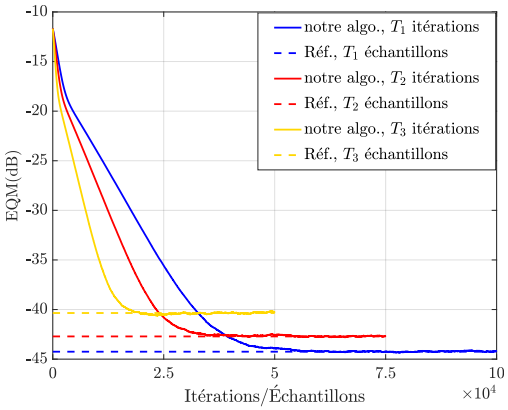


FIGURE 3 – Comparaison de notre algorithme avec Réf..

Expérience 1 : L'algorithme d'apprentissage 1 a été exécuté pour estimer $\boldsymbol{S}$. La courbe d'apprentissage d'écart quadratique moyen estimé (EQM), défini comme suit :

$$\text{EQM}(i) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}\|\boldsymbol{s}_n^* - \boldsymbol{s}_n(i)\|^2 \qquad (16)$$

est représentée dans la Fig. 2(a). Elle montre que l'algorithme a convergé de façon monotone vers un EQM raisonnablement faible et a réussi à s'adapter au changement de $\boldsymbol{S}$ à $i = 5000$, changement dû au passage de γ de 0.1 à 0.6, dans les trois cas concernés : sans régularisation, avec ZA et avec RZA. La Fig. 2(b) illustre en quoi le biais introduit par Φ affecte l'estimation des poids. On relève en revanche une très bonne estimation de la topologie du graphe, définie par le support de $\boldsymbol{S}$, d'où résulte une amélioration de l'EQM.

Expérience 2 : Des comparaisons ont été effectuées avec l'algorithme centralisé présenté dans [8], appelé algorithme de référence (Réf.). Nous avons considéré la même configuration expérimentale que dans l'Expérience 1. Aucun terme de régularisation n'a été utilisé. Puisqu'il s'agit d'un modèle polynomial plus complexe que notre algorithme, nous avons simplifié le modèle Réf. en fixant ses coefficients supplémentaires à 0. Comme Réf. est un algorithme de type batch qui utilise des données d'apprentissage, nous avons successivement fixé la taille du jeu d'apprentissage à $T_1 = 10^5$, $T_2 = 7.5 \cdot 10^4$, et $T_3 = 5 \cdot 10^4$ échantillons. Dans chaque cas, les paramètres de Réf. ont été établis de manière à obtenir le meilleur EQM possible. Nous avons fixé les pas μ_n de notre algorithme pour obtenir le même EQM en régime stationnaire que Réf. Les résultats sont présentés dans la Fig. 3. Nous observons que notre algorithme a pu atteindre le même EQM avec moitié moins d'observations dans chacun des cas.

5 Conclusion

Dans cet article, nous avons proposé une stratégie distribuée et en ligne pour l'identification de la topologie d'un graphe à partir de signaux sur graphe dynamiques. Cet algorithme permet d'estimer une matrice d'adjacence à partir de calculs locaux et peut s'adapter en ligne à des changements de topologie au cours du temps.

Références

[1] N. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1993.

[2] A. Bolstad, B. D. Van Veen, and R. Nowak. Causal network inference via group sparse regularization. *IEEE Transactions on Signal Processing*, 59(6) :2628–2641, 2011.

[3] A. P. Dempster. Covariance selection. *Biometrics*, 28(1) :157–175, 1972.

[4] Xiaowen Dong, Dorina Thanou, Michael Rabbat, and Pascal Frossard. Learning graphs from data : A signal representation perspective. *arXiv preprint arXiv :1806.00848*, 2018.

[5] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 9(3) :432–41, 2008.

[6] F. Hua, R. Nassif, C. Richard, H. Wang, and A. H. Sayed. A preconditioned graph diffusion LMS for adaptive graph signal processing. In *Proc. EUSIPCO*, pages 1–5, Rome, Italy, 2018.

[7] D. Jin, J. Chen, C. Richard, and J. Chen. Model-driven online parameter adjustment for zero-attracting LMS. *Signal Processing*, 152, 06 2018.

[8] J. Mei and J. M. F. Moura. Signal processing on graphs : Causal modeling of unstructured data. *IEEE Transactions on Signal Processing*, 65(8) :2077–2092, 2017.

[9] R. Nassif, C. Richard, J. Chen, and A. H. Sayed. Distributed diffusion adaptation over graph signals. In *Proc. IEEE ICASSP*, pages 4129–4133, Calgary, Canada, 2018.

[10] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond. GSPBOX : A toolbox for signal processing on graphs. *ArXiv e-prints*, 2014.

[11] A. Sandryhaila and J. M. F. Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7) :1644–1656, 2013.

[12] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro. Network topology inference from spectral templates. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3) :467–483, 2017.

[13] Y. Shen, B. Baingana, and G. B. Giannakis. Topology inference of directed graphs using nonlinear structural vector autoregressive models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6513–6517, 2017.

[14] 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*, 30(3) :83–98, 2013.

[15] S. Vlaski, H. P. Maretic, R. Nassif, P. Frossard, and A. H. Sayed. Online graph learning from sequential data. In *Proc. IEEE Data Science Workshop*, pages 190–194, Lausanne, Switzerland, 2018.

[16] L. Zhang, G. Wang, and G. B. Giannakis. Going beyond linear dependencies to unveil connectivity of meshed grids. In *Proc. IEEE CAMSAP*, pages 1–5, Curaçao, Dutch Antilles, 2017.