Decentralized clustering for node-variant graph filtering with graph diffusion LMS

Fei Hua*†, Roula Nassif‡, Cédric Richard†, Haiyan Wang*, Ali H. Sayed‡
*School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China
† Laboratoire Lagrange, Université Côte d'Azur, OCA, CNRS, Nice 06108, France
‡Ecole Polytechnique Fédérale de Lausanne, Switzerland
Email: fei.hua@oca.eu; roula.nassif@epfl.ch; cedric.richard@unice.fr; hywang@nwpu.edu.cn; ali.sayed@epfl.ch

Abstract

In this work, we consider the problem of estimating the coefficients of linear shift-invariant FIR graph filters. We assume hybrid node-varying graph filters where the network is decomposed into clusters of nodes and within each cluster all nodes have the same filter coefficients to estimate. We assume that there is no prior information on the clusters composition and that the nodes do not know which other nodes share the same estimation task. We are interested in distributed, adaptive, and collaborative solutions. In order to limit the cooperation between clustered agents sharing the same estimation task, we propose an extended diffusion preconditioned LMS strategy allowing the nodes to perform automatic network clustering. Simulation results illustrate the effectiveness of the proposed unsupervised method in clustering and collaborative estimation.

Extended summary

A. Graph filter and data model

We consider an undirected weighted graph $\mathcal{G}$ that consists of a set $\mathcal{N}$ of N nodes, and a set $\mathcal{E}$ of edges such that if node k is connected to node ℓ, then $(k,\ell) \in \mathcal{E}$. We denote by $\mathcal{N}_k$ the neighborhood of node k with respect to $\mathcal{E}$, including node k. Graph signals are defined as $\boldsymbol{x} = [x_1, \dots, x_N]^\top \in \mathbb{R}^N$ where x_k is the signal sample at node k. Let $\boldsymbol{x}(i)$ denote the graph signal at time i. We assume that the graph is endowed with a graph-shift operator defined as an $N \times N$ shift matrix $\boldsymbol{S}$ whose entry $s_{k\ell}$ can be non-zero only if $k = \ell$ or $(k,\ell) \in \mathcal{E}$. Possible choices include the adjacency matrix, the graph Laplacian matrix, and their normalized counterparts [1]–[3]. We consider linear shift-invariant FIR graph filters defined by the linear operator:

$$\boldsymbol{H} \triangleq \sum_{m=0}^{M-1} h_m \boldsymbol{S}^m, \tag{1}$$

with $\boldsymbol{h}^o = \{h_m\}_{m=0}^{M-1}$ denoting the filter coefficients and M its order. One common filtering model assumes that the filtered graph signal $\boldsymbol{y}(i)$ is generated from the input graph signal $\boldsymbol{x}(i)$ as follows [4]:

$$\boldsymbol{y}(i) = \sum_{m=0}^{M-1} h_m \boldsymbol{S}^m \boldsymbol{x}(i-m) + \boldsymbol{v}(i), \qquad i \geq M-1. \tag{2}$$

where $\boldsymbol{v}(i)$ is an i.i.d. zero-mean noise with covariance matrix $\boldsymbol{R}_v$. With this model, each shift $\boldsymbol{S}^m$ is carried out in m time slots. By retaining the following shifted signals at each node ℓ:

$$x_\ell(i-1), [\boldsymbol{S}\boldsymbol{x}(i-2)]_\ell, \dots, [\boldsymbol{S}^{M-2}\boldsymbol{x}(i-M+1)]_\ell,$$

note that only one graph shift is required at each time i to carry out the filtered signal. From model (2), sample $y_k(i)$ at node k can be written as:

$$y_k(i) = \boldsymbol{z}_k^\top(i)\boldsymbol{h}^o + v_k(i) \tag{3}$$

with $i \geq M-1$, where $\boldsymbol{z}_k(i)$ is given by:

$$\boldsymbol{z}_k(i) \triangleq \text{col}\{[\boldsymbol{x}(i)]_k,\, [\boldsymbol{S}\boldsymbol{x}(i-1)]_k, \dots, [\boldsymbol{S}^{M-1}\boldsymbol{x}(i-M+1)]_k\}. \tag{4}$$

In order to estimate $\boldsymbol{h}^o$ from $\{y_k(i), \boldsymbol{z}_k(i)\}$ in a collaborative, distributed, and adaptive manner, diffusion LMS strategies can be employed [5]–[7], as explained in [4]. However, since the shift matrix $\boldsymbol{S}$ is not energy preserving in general [8], this may result in a large eigenvalue spread and a slow convergence rate of LMS type strategies. Preconditioned diffusion LMS strategy is then proposed in [9] to solve this problem.

The model (1) is referred to as the *node-invariant* graph filter where the coefficients $\boldsymbol{h}^o$ are the same for all the nodes. A more flexible model was recently introduced in [10]. It is called *node-variant* graph filter and allows the coefficients to vary across nodes:

$$\boldsymbol{H} \triangleq \sum_{m=0}^{M-1} \text{diag}\{\boldsymbol{h}^{(m)}\}\boldsymbol{S}^m, \tag{5}$$

where $\boldsymbol{h}^{(m)} \in \mathbb{R}^N$ is an $N \times 1$ vector. By setting $\boldsymbol{h}^{(m)} = h_m \mathbb{1}_N$ for all m, model (5) reduces to the node-invariant model (1). If the entries of $\boldsymbol{h}^{(m)}$ are different, each node applies different weight to the shifted graph signal $\boldsymbol{S}^m\boldsymbol{x}$, and $y_k(i)$ in (3) can be re-written as:

$$y_k(i) = \boldsymbol{z}_k^\top(i)\boldsymbol{h}_k^o + v_k(i), \;\text{ with }\; i \geq M-1, \tag{6}$$

where $\boldsymbol{h}_k^o = \text{col}\{[\boldsymbol{h}^{(m)}]_k\}_{m=0}^{M-1}$ is the $M \times 1$ filter coefficient vector associated with node k. In this work, we consider the problem of estimating $\boldsymbol{h}_k^o$ at each node k from the signals $\{y_k(i), \boldsymbol{z}_k(i)\}$. We further assume that the graph is decomposed into Q clusters of nodes $\mathcal{C}_q$ and, within each cluster $\mathcal{C}_q$, there is a common filter coefficient vector $\boldsymbol{h}_q^o$ to estimate, namely,

$$\boldsymbol{h}_k^o = \boldsymbol{h}_q^o, \qquad \text{if } k \in \mathcal{C}_q.$$

This model is the so-called *hybrid node-varying* graph filter [11]. In this paper, we consider unsupervised scenarios where there is no prior information on the clusters composition, and where the nodes do not know which other nodes share the same filter coefficient vector. The only available information is that clusters may exist in the network (but their structures are unknown).

B. Diffusion preconditioned LMS for graph signal filtering

Let us first recall the diffusion preconditioned LMS algorithm initially proposed in [9] in order to estimate $\boldsymbol{h}^o$ in (3) in a fully distributed and adaptive manner.

$$\begin{cases} \boldsymbol{\psi}_k(i+1) = \boldsymbol{h}_k(i) + \mu_k[\epsilon \boldsymbol{I} + \boldsymbol{P}_k]^{-1}\boldsymbol{z}_k(i)\big[y_k(i) - \boldsymbol{z}_k^\top(i)\boldsymbol{h}_k(i)\big] & \text{(7a)} \\ \boldsymbol{h}_k(i+1) = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_\ell(i+1) & \text{(7b)} \end{cases}$$

where $\boldsymbol{h}_k(i)$ is the estimate of $\boldsymbol{h}^o$ at node k and iteration i, $\boldsymbol{\psi}_k(i)$ is an intermediate estimate, $\mu_k > 0$ is a local step-size parameter, $\epsilon \geq 0$ is a small regularization parameter, $\boldsymbol{P}_k$ is an $M \times M$ preconditioning matrix constructed locally according to:

$$\boldsymbol{P}_k \triangleq \text{diag}\{\|[\boldsymbol{S}^{(m-1)}]_{k,\bullet}\|^2\}_{m=1}^{M}. \tag{8}$$

and $\{a_{\ell k}\}$ are non-negative combination coefficients chosen to satisfy:

$$a_{\ell k} > 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \;\text{ and }\; a_{\ell k} = 0 \;\text{ if }\; \ell \notin \mathcal{N}_k. \tag{9}$$

In the adaptation step (7a), each node k uses the data from its one-hop neighbors to compute $\boldsymbol{z}_k(i)$, then updates its local estimate $\boldsymbol{h}_k(i)$ to an intermediate estimate $\boldsymbol{\psi}_k(i+1)$. In the combination step (7b), node k aggregates all the intermediate estimates $\boldsymbol{\psi}_\ell(i+1)$ from its neighbors to obtain the updated estimate $\boldsymbol{h}_k(i+1)$. When algorithm (7a)–(7b) is applied to estimate filter coefficient vectors arising from different data models (6), and in order to avoid the bias resulting from combining estimates in (7b) corresponding to different data models, automatic network clustering strategies should be used to inhibit cooperation between clustered agents [12]–[15]. In the following, we introduce an unsupervised clustering rule.

C. Unsupervised clustering method

We first introduce an $N \times N$ clustering matrix $\boldsymbol{E}_i$ at time i, whose (ℓ, k)-th element is given by:

$$[\boldsymbol{E}_i]_{\ell k} = \begin{cases} 1, & \text{if } \;\ell \in \mathcal{N}_k\; \text{ and } \;k\; \text{ believes that } \;\boldsymbol{h}_k^o = \boldsymbol{h}_\ell^o, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

At each time i, node k can infer which neighbors belong to the same cluster based on the non-zero elements of the k-th column of $\boldsymbol{E}_i$. We collect these indices into the set $\mathcal{N}_{k,i} \triangleq \{\ell \,|\, [\boldsymbol{E}_i]_{\ell k} = 1\}$. Then, node k will only combine the intermediate estimates from its neighbors in $\mathcal{N}_{k,i}$ and the combination rule (9) becomes:

$$a_{\ell k} > 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \;\text{ and }\; a_{\ell k} = 0 \;\text{ if }\; \ell \notin \mathcal{N}_{k,i}. \tag{11}$$

Since the clustering information is not known beforehand, we propose to learn $\boldsymbol{E}_i$ in an on-line manner by evaluating the ℓ_2-norm distance between the estimates at two different nodes. If the distance is smaller than a predefined threshold, the two nodes are assigned to the same cluster. At each time i, node k runs a stand-alone adaptation step (7a) and then computes a Boolean variable within its neighborhood $\mathcal{N}_k$:

$$b_{\ell k}(i) = \begin{cases} 1, & \text{if } \|\boldsymbol{\psi}_\ell(i+1) - \boldsymbol{h}_k(i)\|^2 \leq \alpha, \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

where α is a predefined threshold. Depending on the spectrum of $\boldsymbol{S}$, the variance of the shifted signal $\boldsymbol{S}^m\boldsymbol{x}$ in some eigen subspaces of $\boldsymbol{S}$ may dramatically increase or tend to zero as m increases. This numerical ill-conditioning may affect the accuracy of the estimates of some entries of $\boldsymbol{\psi}_\ell(i+1)$, and result in poor clustering performance as illustrated in the sequel. To address this issue, we suggest to evaluate the distance in (11) based on the dominant principal components of the local estimates. In order to identify the first M_k dominant entries at each node k (principle components), we propose to use the following rule:

$$\min_{M_k} \quad \frac{\sum_{m=1}^{M_k}[\text{diag}(\boldsymbol{P}_k)]_m}{\sum_{m=1}^{M}[\text{diag}(\boldsymbol{P}_k)]_m} \geq \tau, \tag{13}$$

where $\boldsymbol{P}_k$ is the diagonal preconditioning matrix defined in (8), with diagonal entries arranged in descending order. Parameter τ denotes a threshold in $[0, 1]$. The rationale of this rule is that, in (7a), matrix $\boldsymbol{P}_k$ approximates the covariance matrix (up to a scaling factor) of the observations $\boldsymbol{z}_k(i)$; see [9] for details. As $\boldsymbol{P}_k$ is diagonal, (13) then represents the ratio of total inertia explained by the first M_k principal components (or entries, as $\boldsymbol{P}_k$ is diagonal) of the observations. Once M_k is computed, each node k uses the following rule to compute the Boolean variable $b_{\ell k}(i)$ instead of the rule (12):

$$b_{\ell k}(i) = \begin{cases} 1, & \text{if } \frac{\left\|[\boldsymbol{\psi}_\ell(i+1) - \boldsymbol{h}_k(i)]_{1:M_k}\right\|^2}{\left\|[\boldsymbol{h}_k(i)]_{1:M_k}\right\|^2} \leq \alpha', \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where α' is a small positive value. Compared with (12), note that we suggest to use a normalized distance in order to simplify parameter α' setting. To reduce the influence of noise, we further introduce a smoothing step:

$$t_{\ell k}(i+1) = \nu t_{\ell k}(i) + (1-\nu) b_{\ell k}(i), \tag{15}$$

with $0 < \nu < 1$ a forgetting factor and $t_{\ell k}(i)$ a trust level. Once $t_{\ell k}(i)$ exceeds a given threshold θ, node k sets $[\boldsymbol{E}_i]_{\ell k} = 1$, i.e., it believes that node ℓ belongs to its cluster. Thus, the clustering matrix $\boldsymbol{E}_i$ and the neighborhood set $\mathcal{N}_{k,i} \triangleq \{\ell \,|\, [\boldsymbol{E}_i]_{\ell k} = 1\}$ are learned in an on-line manner. Note that the set $\mathcal{N}_k$ in the combination step (7b) is replaced by $\mathcal{N}_{k,i}$ which contains only the neighbors $\ell \in \mathcal{N}_k$ that node k believes they belong to its cluster. Parameters $a_{\ell k}$ must satisfy (11) and have to be modified accordingly.

D. Simulation results

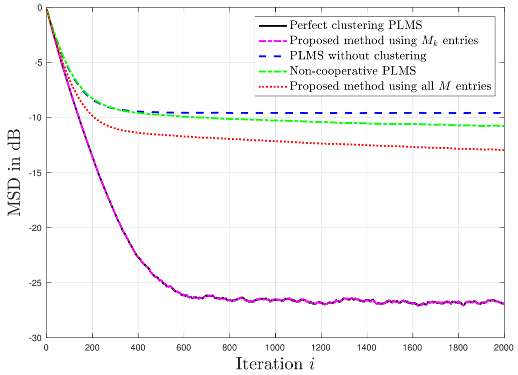

Fig. 1: Network MSD performance for different algorithms.

We tested the proposed clustering algorithm over an undirected weighted graph of $N = 60$ nodes. In particular, we considered a sensor network generated by GSPBOX [16] where each node is connected to its 5 nearest neighbors. The graph shift operator was chosen as the normalized adjacency matrix $\boldsymbol{S} = \frac{\boldsymbol{A}}{1.3\lambda_{\max}(\boldsymbol{A})}$ with $\boldsymbol{A}$ the adjacency matrix and $\lambda_{\max}(\boldsymbol{A})$ its largest eigenvalue. The graph signal process $\boldsymbol{x}(i)$ was i.i.d. zero-mean Gaussian with covariance matrix $\boldsymbol{R}_x = \text{diag}\{\sigma^2_{x,k}\}_{k=1}^N$. The variances $\sigma^2_{x,k}$ were randomly generated from the uniform distribution $\mathcal{U}(1, 1.5)$. The noise $\boldsymbol{v}(i)$ was zero-mean Gaussian with covariance

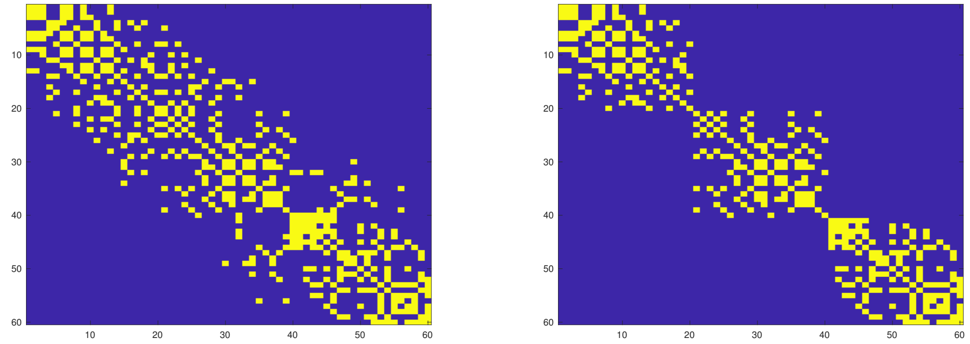

Fig. 2: *(Left)* Topology of the graph (adjacency matrix). *(Right)* Inferred cluster matrix at steady-state.

matrix $\boldsymbol{R}_v = \text{diag}\{\sigma_{v,k}^2\}_{k=1}^N$. The variances $\sigma_{v,k}^2$ were randomly generated from the uniform distribution $\mathcal{U}(0.1, 0.15)$. The filter degree was set to $M = 3$. The nodes were decomposed into three clusters $\mathcal{C}_q$ with $\mathcal{C}_1 = \{1, \ldots, 20\}$, $\mathcal{C}_2 = \{21, \ldots, 40\}$, and $\mathcal{C}_3 = \{41, \ldots, 60\}$. The optimal graph filter coefficients $\boldsymbol{h}_k^o$ were set according to $[0.5\ 0.4\ 0.9]^\top$ if $k \in \mathcal{C}_1$, $[0.3\ 0.1\ 0.4]^\top$ if $k \in \mathcal{C}_2$, and $[0.9\ 0.3\ 0.7]^\top$ if $k \in \mathcal{C}_3$. The simulation results were averaged over 200 Monte-Carlo runs. We compared the proposed algorithm with the ground truth algorithm where the clusters are assumed to be known a priori, the non-cooperative algorithm (where $a_{\ell k} = 1$ if $k = \ell$ and zero otherwise), the diffusion preconditioned LMS (PLMS) algorithm in (7a)–(7b) without clustering, and algorithm (7a)–(7b) with the clustering rule that updates the Boolean variable according to (12) where all the entries of the estimates are used ($M_k = M$ for all k). Note that all algorithms used the preconditioned LMS (PLMS) in the adaptation step. As shown in Fig.1, the proposed algorithm performs well compared to the ground truth. Figure 2 (left) shows the topology of the graph given by the adjacency matrix $\boldsymbol{A}$ (and the shift matrix $\boldsymbol{S}$). Figure 2 (right) shows the clusters inferred by the proposed method. These clusters perfectly match the ground truth clusters.

References

[1] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing," *arXiv:1712.00468*, Mar. 2017.

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

[3] A. Sandryhaila and J. M. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, 2014.

[4] R. Nassif, C. Richard, J. Chen, and A. H. Sayed, "Distributed diffusion adaptation over graph signals," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, Canada, 2018, pp. 1–5.

[5] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, R. Chellapa and S. Theodoridis, Eds. Elsevier, 2014. Also available as arXiv:1205.4220 [cs.MA], May 2012., pp. 322–454.

[6] ——, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[7] ——, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

[8] A. Gavili and X.-P. Zhang, "On the shift operator, graph frequency, and optimal filtering in graph signal processing," *IEEE Transactions on Signal Processing*, vol. 65, no. 23, pp. 6303–6318, 2017.

[9] F. Hua, R. Nassif, C. Richard, H. Wang, and A. H. Sayed, "A preconditioned graph diffusion LMS for adaptive graph signal processing," in *submitted for conference publication*, Mar. 2018.

[10] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4117–4131, 2017.

[11] F. Gama, G. Leus, A. G. Marques, and A. Ribeiro, "Convolutional neural networks via node-varying graph filters," *arXiv:1710.10355*, Oct. 2017.

[12] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, 2015.

[13] X. Zhao and A. H. Sayed, "Distributed clustering and learning over networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3285–3300, 2015.

[14] J. Plata-Chaves, M. H. Bahari, M. Moonen, and A. Bertrand, "Unsupervised diffusion-based LMS for node-specific parameter estimation over wireless sensor networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 4159–4163.

[15] S. Khawatmi, A. H. Sayed, and A. M. Zoubir, "Decentralized clustering and linking by networked agents," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3526–3537, 2017.

[16] 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:1408.5781*, Aug. 2014.