

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 for clustering nodes into clusters and collaborative estimation.

Index Terms—Graph signal processing, graph filter, node-varying, diffusion LMS, clustering.

I. INTRODUCTION

Shift-invariant FIR graph filters are defined as polynomials in a given graph shift linear operator such as the Laplacian matrix [1], or the adjacency matrix [2], [3]. Since graph shift operators capture local network topology, FIR graph filters naturally admit distributed implementations. A standard class of FIR graph filters includes *node-invariant* graph filters where all nodes in the graph share a common set of filter coefficients. An extension of this model involving *node-variant* graph filters is considered in [4]. A *hybrid node-varying* filter model, which assumes clusters in the graph and common filter coefficients within each cluster is also proposed in [5]. Finally, besides FIR graph filters, IIR extensions are also considered in [6], [7].

Most of these works consider filtering static input signals. Several other works consider time-varying graph signals. For example, in [8], the authors analyze the behavior of FIR and ARMA graph filters for random time-varying graph signals. In [9], a time-vertex framework is proposed for joint harmonic analysis of time-varying graph signals. In [10], a causal

The work of F. Hua was supported in part by China Scholarship Council and NSFC grant 61471298. The work of C. Richard was supported in part by Idex UCA-Jedi (ACADY project) and CNRS Mastodons (AGADIR project). The work of H. Wang was partly supported by NSFC under grants 61401364, 61571365, 61671386 and National Key R&D Program of China (2016YFC1400200). The work of A. H. Sayed was supported in part by NSF grant CCF-1524250.

model based on graph filters is used to characterize graph signal processes. Several methods have also been proposed to estimate filter coefficients. In particular, in [11], [12], the authors show how to estimate the coefficients of node-invariant graph filters from streaming graph signals.

In this work, we consider the problem of estimating the filter coefficients of hybrid node-varying graph filters in a cooperative manner from streaming graph signals. We propose an unsupervised clustering strategy where nodes are able to select from among their neighbors those other nodes that may share the same filter coefficients. This allows them to collaborate efficiently on a shared estimation task. The remainder of paper is organized as follows. In Section II, we introduce the filter and data models. In Section III, we briefly review how to estimate the coefficients of node-invariant graph filters in a distributed and adaptive manner. An online learning algorithm for clustering nodes and estimating filter coefficients is provided in Section IV. Simulation results are given in Section V.

Notation: We use normal font letters to denote scalars, boldface lowercase letters to denote column vectors, and boldface uppercase letters to denote matrices. We use $\text{diag}\{x_1, \dots, x_N\}$ to denote a diagonal matrix consisting of diagonal entries x_1, \dots, x_N and $\text{diag}\{P\}$ to denote a vector collecting the diagonal entries of matrix P . The symbol $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of its matrix argument. The m -th entry of a vector x is denoted by $[x]_m$, the (m, n) -th entry of a matrix X is denoted by $[X]_{mn}$, and the k -th row of a matrix X is denoted by $[X]_{k, \bullet}$.

II. PRELIMINARIES

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 . Assume that the graph is endowed with a graph-shift operator defined as an $N \times N$ shift matrix S whose entry $s_{k\ell}$ can be non-zero only if $k = \ell$ or $(k, \ell) \in \mathcal{E}$. Popular choices include the adjacency matrix, the graph Laplacian matrix, and

their normalized counterparts [13]. We consider linear shift-invariant FIR graph filters defined by the linear operator [2]:

$$\mathbf{H}_{\text{ni}} \triangleq \sum_{m=0}^{M-1} h_m \mathbf{S}^m, \quad (1)$$

with $\mathbf{h}^o = \{h_m\}_{m=0}^{M-1}$ denoting the filter coefficients and M its order. Model (1) is referred to as the *node-invariant* graph filter since the coefficients \mathbf{h}^o are the same for all nodes. Graph signals are defined as $\mathbf{x} = [x_1, \dots, x_N]^\top \in \mathbb{R}^N$ where x_k is the signal sample at node k . Let $\mathbf{x}(i)$ denote the graph signal at time i . One common filtering model assumes that the filtered graph signal $\mathbf{y}(i)$ is generated from the input graph signal $\mathbf{x}(i)$ as follows [10], [12]:

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

where $\mathbf{v}(i)$ is an i.i.d. zero-mean noise with covariance matrix \mathbf{R}_v . With this model, each shift \mathbf{S}^m is carried out in m time slots. By retaining the following shifted signals at each node ℓ at time $i-1$:

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

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

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

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

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

The input vector $\mathbf{z}_k(i)$ at each node k can be computed from its one-hop neighbors using the retained shifted signals. In order to estimate \mathbf{h}^o from $\{y_k(i), \mathbf{z}_k(i)\}$ in a collaborative, distributed, and adaptive manner, diffusion LMS strategies can be employed [14]–[16] as already explained in [12]. However, since the shift matrix \mathbf{S} is not energy preserving in general [17], this may result in a large eigenvalue spread and reduce the convergence rate of LMS type strategies. A preconditioned diffusion LMS strategy was proposed in [18] to address this issue.

A more flexible model than (1) is introduced in [4]. It is referred to as a *node-variant* graph filter and allows the coefficients to vary across nodes:

$$\mathbf{H}_{\text{nv}} \triangleq \sum_{m=0}^{M-1} \text{diag}\{\mathbf{h}^{(m)}\} \mathbf{S}^m, \quad (5)$$

with $\mathbf{h}^{(m)} \in \mathbb{R}^N$, an $N \times 1$ vector. By setting $\mathbf{h}^{(m)} = h_m \mathbf{1}_N$ for all m , model (5) reduces to the node-invariant model (1). If the entries of $\mathbf{h}^{(m)}$ are different, each node applies different weight to the shifted graph signal $\mathbf{S}^m \mathbf{x}$. This model introduces additional degrees of freedom but more complexity into the design of the filter. To get a trade-off between flexibility

and complexity, the authors of [5] propose the *hybrid node-varying* graph filter:

$$\mathbf{H}_{\text{hv}} \triangleq \sum_{m=0}^{M-1} \text{diag}\{\mathbf{C}_Q \mathbf{h}_Q^{(m)}\} \mathbf{S}^m, \quad (6)$$

with $\mathbf{C}_Q \in \{0, 1\}^{N \times Q}$ a tall binary matrix with only one non-zero entry at each row and zeros elsewhere, $\mathbf{h}_Q^{(m)} \in \mathbb{R}^Q$ a $Q \times 1$ vector, where $1 < Q < N$. This model indicates that there are Q clusters \mathcal{C}_q of nodes in the graph and, within each cluster \mathcal{C}_q , there is a common filter coefficient vector $\mathbf{h}_q^o \triangleq \{\mathbf{h}_Q^{(m)}\}_{m=0}^{M-1}$. Therefore, the corresponding filtered signal $y_k(i)$ in (3) can be re-written as:

$$y_k(i) = \mathbf{z}_k^\top(i) \mathbf{h}_k^o + v_k(i), \quad i \geq M-1, \quad (7)$$

and, for each node k , the coefficients satisfy:

$$\mathbf{h}_k^o = \mathbf{h}_q^o, \quad \text{if } k \in \mathcal{C}_q. \quad (8)$$

In this paper, we consider unsupervised scenarios where there is no prior information on the clusters composition. We also assume that 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.

III. GRAPH DIFFUSION LMS

Let us first describe the graph diffusion LMS strategy with preconditioning from [18], which allows us to estimate \mathbf{h}^o in (3) in a fully distributed and adaptive manner. Consider the mean-square-error cost $J_k(\mathbf{h})$ at node k :

$$J_k(\mathbf{h}) = \mathbb{E}|y_k(i) - \mathbf{z}_k^\top(i) \mathbf{h}|^2, \quad (9)$$

with the aggregate cost given by

$$J(\mathbf{h}) = \sum_{k=1}^N J_k(\mathbf{h}). \quad (10)$$

Several strategies exist that can be used to minimize costs of the form (10) in a distributed manner, e.g., incremental, consensus and diffusion techniques — see [16]. Taking into account the non-energy preserving property of the graph shift operator \mathbf{S} , the diffusion preconditioned LMS strategy takes the following form at each node k and time instant i [18]:

$$\boldsymbol{\psi}_k(i+1) = \mathbf{h}_k(i) + \mu_k (\epsilon \mathbf{I} + \mathbf{P}_k)^{-1} \mathbf{z}_k(i) e_k(i) \quad (11a)$$

$$\mathbf{h}_k(i+1) = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_\ell(i+1) \quad (11b)$$

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

$$\mathbf{P}_k \triangleq \text{diag}\left\{\|[\mathbf{S}^{(m-1)}]_{k,\bullet}\|^2\right\}_{m=1}^M \quad (12)$$

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

$$a_{\ell k} > 0, \quad \sum_{\ell=1}^N a_{\ell k} = 1, \quad \text{and} \quad a_{\ell k} = 0 \quad \text{if} \quad \ell \notin \mathcal{N}_k. \quad (13)$$

In the adaptation step (11a), each node k uses the data from its one-hop neighbors to compute $\mathbf{z}_k(i)$, then updates its local estimate $\mathbf{h}_k(i)$ to an intermediate estimate $\boldsymbol{\psi}_k(i+1)$. In the combination step (11b), node k aggregates all the intermediate estimates $\boldsymbol{\psi}_\ell(i+1)$ from its neighbors to obtain the updated estimate $\mathbf{h}_k(i+1)$. When algorithm (11a)–(11b) is applied to estimate filter coefficient vectors arising from different data models (7), automatic network clustering strategies should be used to inhibit cooperation between clustered agents [19]–[22] in order to avoid bias resulting from combining estimates in (11b) corresponding to different data models. In the following, we introduce an unsupervised clustering rule to address this issue and mitigate the estimation bias.

IV. UNSUPERVISED CLUSTERING METHOD

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

$$[\mathbf{E}_i]_{\ell k} = \begin{cases} 1, & \text{if } \ell \in \mathcal{N}_k \text{ and } k \text{ believes that } \mathbf{h}_k^o = \mathbf{h}_\ell^o, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

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 \mathbf{E}_i . We collect these indices into the set $\mathcal{N}_{k,i} \triangleq \{\ell \mid [\mathbf{E}_i]_{\ell k} = 1\}$. Then, node k will only combine the intermediate estimates from its neighbors in $\mathcal{N}_{k,i}$ and the combination rule (13) becomes:

$$a_{\ell k} > 0, \quad \sum_{\ell=1}^N a_{\ell k} = 1, \quad \text{and} \quad a_{\ell k} = 0 \quad \text{if} \quad \ell \notin \mathcal{N}_{k,i}. \quad (15)$$

Since the clustering information is not known beforehand, we propose to learn \mathbf{E}_i in an online 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 instant i , node k runs a stand-alone adaptation step (11a) and then computes a Boolean variable within its neighborhood \mathcal{N}_k :

$$b_{\ell k}(i) = \begin{cases} 1, & \text{if } \|\boldsymbol{\psi}_\ell(i+1) - \mathbf{h}_k(i)\|^2 \leq \alpha, \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where α is a predefined threshold. Depending on the spectrum of matrix \mathbf{S} , the variance of the shifted signal $\mathbf{S}^m \mathbf{x}$ in some eigen-subspaces of \mathbf{S} may dramatically increase or tend to zero as m increases. This numerical ill-conditioning may affect the estimation accuracy 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 propose to evaluate the distance in (16), in the subspace spanned by the dominant principal components of the local input data. To evaluate the relative contribution of

each principal component to the whole data space, one can compute the *proportion of total variance* [23] given by:

$$\pi_{k,m} \triangleq \frac{\lambda_m(\mathbf{R}_{\mathbf{z},k})}{\text{Tr}(\mathbf{R}_{\mathbf{z},k})}, \quad (17)$$

where $\mathbf{R}_{\mathbf{z},k} \triangleq \mathbb{E}\{\mathbf{z}_k(i)\mathbf{z}_k^\top(i)\}$ is the covariance matrix of the input data $\mathbf{z}_k(i)$, $\text{Tr}(\cdot)$ is the trace of its matrix argument, and $\lambda_m(\cdot)$ is its m -th eigenvalue. Since $\mathbf{R}_{\mathbf{z},k}$ is usually unknown beforehand, we propose to approximate (17) as follows:

$$\hat{\pi}_{k,m} = \frac{[\mathbf{p}_k]_m}{\text{Tr}(\mathbf{P}_k)}, \quad (18)$$

where \mathbf{P}_k denotes the diagonal preconditioning matrix defined in (12), and \mathbf{p}_k is the vector collecting the diagonal entries of \mathbf{P}_k . The rationale behind (18) is that matrix \mathbf{P}_k can be used to approximate the covariance matrix (up to a scaling factor) of the observations $\mathbf{z}_k(i)$; see [18] for details. In practice, it is common to use some predefined percentage of total variance. We propose to use the following rule to identify the first M_k principal components:

$$\begin{aligned} & \text{minimize} \quad M_k \\ & \text{subject to} \quad \sum_{m=1}^{M_k} \hat{\pi}_{k,m} \geq \tau, \end{aligned} \quad (19)$$

where parameter τ denotes a threshold in $[0, 1]$. Since \mathbf{P}_k is diagonal, expression (19) then represents the ratio of total inertia explained by the first M_k principal components 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 (16):

$$b_{\ell k}(i) = \begin{cases} 1, & \text{if } \frac{\|[\boldsymbol{\psi}_\ell(i+1) - \mathbf{h}_k(i)]_{1:M_k}\|^2}{\|[\mathbf{h}_k(i)]_{1:M_k}\|^2} \leq \alpha', \\ 0, & \text{otherwise,} \end{cases} \quad (20)$$

where α' is a small positive value. Compared with (16), note that we suggest to use a normalized distance in order to simplify the choice of α' . To reduce the influence of noise, we further introduce a smoothing step:

$$t_{\ell k}(i) = \nu t_{\ell k}(i-1) + (1-\nu)b_{\ell k}(i), \quad (21)$$

where $0 < \nu < 1$ denotes a forgetting factor and $t_{\ell k}(i)$ is a trust level. Once $t_{\ell k}(i)$ exceeds a given threshold θ , node k sets $[\mathbf{E}_i]_{\ell k} = 1$, i.e., it believes that node ℓ belongs to its cluster. In this way, the clustering matrix \mathbf{E}_i and the neighborhood set $\mathcal{N}_{k,i} = \{\ell \mid [\mathbf{E}_i]_{\ell k} = 1\}$ are learned in an online and distributed manner. Note that the set \mathcal{N}_k in the combination step (11b) 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 (15) and have to be modified accordingly in an online manner. The proposed strategy is summarized in Algorithm 1.

Algorithm 1 Unsupervised clustering for graph diffusion LMS with preconditioning.

Initialize: $\mathbf{h}_k(1) = 0$, $\boldsymbol{\psi}_k(1) = 0$, $b_{kk}(1) = 1$, $t_{kk} = 1$ for $k \in \{1, \dots, N\}$, $\mathbf{E}_1 = \mathbf{I}$. Compute \mathbf{P}_k according to (12) and M_k according to (19) for all k .

```

1: for  $i \geq 1$  do
2:   for  $k = 1 : N$  do
3:     compute  $z_k(i)$  according to (4)
4:     adaptation:
5:      $\boldsymbol{\psi}_k(i+1) = \mathbf{h}_k(i) + \mu_k[\epsilon\mathbf{I} + \mathbf{P}_k]^{-1}z_k(i)e_k(i)$ 
6:     for  $\ell \in \mathcal{N}_k$  do
7:       compute the Boolean variable according to (20)
8:       smoothing step according to (21)
9:       update  $[\mathbf{E}_i]_{\ell k} = 1$  if  $t_{\ell k}(i) > \theta$ 
10:    end for
11:    choose  $a_{\ell k}$  according to (15)
12:    combination:  $\mathbf{h}_k(i+1) = \sum_{\ell \in \mathcal{N}_k, i} a_{\ell k}\boldsymbol{\psi}_\ell(i+1)$ 
13:  end for

```

V. SIMULATION RESULTS

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 [24] where each node is connected to its 5 nearest neighbors. The graph shift operator was chosen as the normalized adjacency matrix $\mathbf{S} = \frac{\mathbf{A}}{1.1\lambda_{\max}(\mathbf{A})}$, with \mathbf{A} the adjacency matrix and $\lambda_{\max}(\mathbf{A})$ its largest eigenvalue. The graph signal $\mathbf{x}(i)$ was i.i.d. zero-mean Gaussian with covariance matrix $\mathbf{R}_x = \text{diag}\{\sigma_{x,k}^2\}_{k=1}^N$. The variances $\sigma_{x,k}^2$ were randomly generated from the uniform distribution $\mathcal{U}(1, 1.5)$. The noise $\mathbf{v}(i)$ was zero-mean Gaussian with covariance matrix $\mathbf{R}_v = \text{diag}\{\sigma_{v,k}^2\}_{k=1}^N$. 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 simulation results were averaged over 100 Monte-Carlo runs.

In the first experiment, 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 (11a)–(11b) without clustering, and algorithm (11a)–(11b) with the clustering rule that updates the Boolean variable according to (16) where 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. The nodes were decomposed into three clusters \mathcal{C}_q with $\mathcal{C}_1 = \{1, \dots, 20\}$, $\mathcal{C}_2 = \{21, \dots, 40\}$, and $\mathcal{C}_3 = \{41, \dots, 60\}$. The optimal graph filter coefficients \mathbf{h}_k^o were set 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 parameters $\{\tau, \alpha', \theta, \nu\}$ were set to $\{0.9, 0.01, 0.5, 0.98\}$. As shown in Fig.1, the proposed algorithm performed well compared to the ground truth and other methods. Figure 2 (Top) shows the topology of the graph given by the adjacency matrix \mathbf{A} (and the shift matrix \mathbf{S}). Figure 2 (Bottom) shows the clusters inferred by the

proposed method. These clusters perfectly match the ground truth clusters.

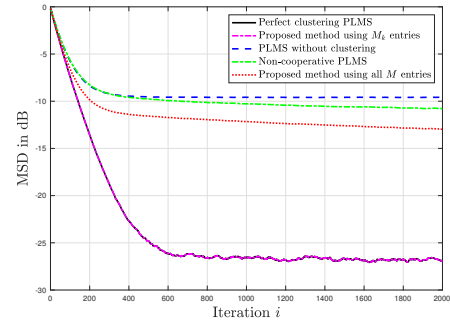


Fig. 1. Network MSD performance for different algorithms.

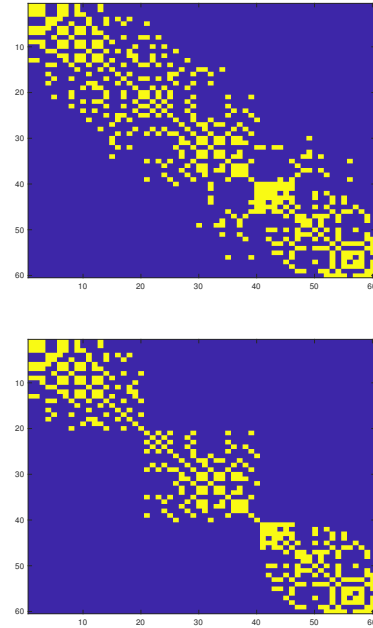


Fig. 2. (Top) Topology of the graph (adjacency matrix). (Bottom) Inferred cluster matrix at steady-state.

In the second test, we considered the scenario where the model assignments changed while the structure of each cluster remained unchanged. The nodes were decomposed into two clusters with $\mathcal{C}_1 = \{1, \dots, 30\}$ and $\mathcal{C}_2 = \{31, \dots, 60\}$. The optimal coefficients changed for both clusters at time instant $i = 1000$. Simulation shows that the proposed method tracked well the changes as shown in Figure 3.

VI. CONCLUSION

We considered the problem of estimating the filter coefficients of hybrid node-varying graph filters, in an adaptive and cooperative manner, in the case where nodes do not know the composition of their own cluster. We proposed a decentralized clustering scheme, based on a graph diffusion LMS strategy

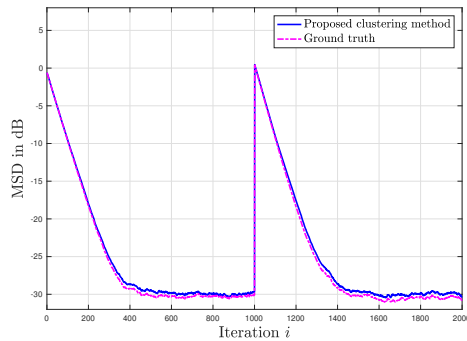


Fig. 3. Network MSD performance with models change.

and approximate principal components, to address this issue. This strategy provides each node the ability to select, among its neighbors, those that share the same set of filter coefficients to estimate. Simulation results were presented to demonstrate the effectiveness of the proposed clustering strategy compared to other methods.

REFERENCES

- [1] 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, Apr. 2013.
- [2] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.
- [3] L. Chen, S. Cheng, V. Stankovic, and L. Stankovic, "Shift-enabled graphs: Graphs where shift-invariant filters are representable as polynomials of shift operations," *IEEE Signal Processing Letters*, vol. 25, no. 9, pp. 1305–1309, Sep. 2018.
- [4] 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, Aug. 2017.
- [5] F. Gama, G. Leus, A. G. Marques, and A. Ribeiro, "Convolutional neural networks via node-varying graph filters," in *Proc. IEEE Data Science Workshop (DSW)*, Lausanne, Switzerland, Jun. 2018, pp. 1–5.
- [6] X. Shi, H. Feng, M. Zhai, T. Yang, and B. Hu, "Infinite impulse response graph filters in wireless sensor networks," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1113–1117, Aug. 2015.
- [7] 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, Jan. 2017.
- [8] —, "Filtering random graph processes over random time-varying graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4406–4421, Aug. 2017.
- [9] F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud, "A time-vertex signal processing framework: Scalable processing and meaningful representations for time-series on graphs," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 817–829, Feb. 2018.
- [10] J. Mei and J. M. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2077–2092, Apr. 2017.
- [11] R. Nassif, C. Richard, J. Chen, and A. H. Sayed, "A graph diffusion LMS strategy for adaptive graph signal processing," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Oct. 2017, pp. 1973–1976.
- [12] —, "Distributed diffusion adaptation over graph signals," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, AB, Canada, Apr. 2018, pp. 4129–4133.
- [13] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
- [14] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, May 2013.
- [15] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4–5, pp. 311–801, 2014.
- [16] —, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.
- [17] 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, Dec. 2017.
- [18] F. Hua, R. Nassif, C. Richard, H. Wang, and A. H. Sayed, "A preconditioned graph diffusion LMS for adaptive graph signal processing," in *Proc. European Signal Processing Conference*, Rome, Italy, Sep. 2018, pp. 1–5.
- [19] 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, May 2015.
- [20] X. Zhao and A. H. Sayed, "Distributed clustering and learning over networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3285–3300, Jul. 2015.
- [21] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, Shanghai, China, Mar. 2016, pp. 4159–4163.
- [22] 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, Jul. 2017.
- [23] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, vol. 374, no. 2065, pp. 1–16, Apr. 2016.
- [24] 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*, 2014.