Distributed Recursive Least Squares Strategies for Adaptive Reconstruction of Graph Signals

Paolo Di Lorenzo[1], Elvin Isufi[2], Paolo Banelli[1], Sergio Barbarossa[3], and Geert Leus[2]

[1] Department of Engineering, University of Perugia, Via G. Duranti 93, 06125, Perugia, Italy

[2] EEMCS, Delft University of Technology, 2826 CD Delft, The Netherlands

[3] Department of Information Engineering, Electronics, and Telecommunications, Sapienza University of Rome, Via Eudossiana 18, 00184, Rome, Italy

Email: paolo.dilorenzo@unipg.it, paolo.banelli@unipg.it
sergio.barbarossa@uniroma1.it,e.isufi-1@tudelft.nl, g.j.t.leus@tudelft.nl

***Abstract*—This work proposes distributed recursive least squares (RLS) strategies for adaptive reconstruction and learning of signals defined over graphs. First, we introduce a centralized RLS estimation strategy with probabilistic sampling, and we propose a sparse sensing method that selects the sampling probability at each node in the graph in order to guarantee adaptive signal reconstruction and a target steady-state performance. Then, a distributed RLS strategy is derived and is shown to be convergent to its centralized counterpart. The performed numerical tests show the performance of the proposed adaptive method for distributed learning of graph signals.**

***Index Terms*—Recursive least squares estimation, graph signal processing, sampling, adaptive networks.**

I. Introduction

Nowadays, there is an increasing demand to process signals living in irregular domains. Several examples abound in the context of sensor networks, social media, big data or biological networks. The underlying structure of data is often represented by a graph and the signal living on top of this graph is defined as a graph signal. Graph signal processing (GSP) [1]–[3] is a promising area that aims to develop novel analysis and processing tools for signals defined over graphs. For instance, there exists a specific definition for the graph Fourier transform (GFT), where the graph signal is projected onto the eigenvectors of either the Laplacian [1], [4], [5], or of the adjacency matrix [2], [6]. Several processing methods for graph signals were proposed in [2], [7]–[10], and a fundamental aspect is their dependence from the graph topology.

A central topic in GSP is the development of a *sampling theory*, whose aim is to reconstruct a graph signal from a subset of its samples. An important contribution is given by [4], later extended in [11] and, very recently, in [6], [12], [13], [14]. Based on the developed sampling theory, several reconstruction methods were proposed, either iterative as in [13], [15], or batch, as in [6], [12]. On the other hand, [16]–[18] propose adaptive strategies for online graph signal reconstruction and learning. Specifically, [16] proposed an LMS estimation strategy enabling adaptive learning and tracking from a limited number of smartly sampled observations, which was then extended to the distributed setting in [17]. In [18], the authors proposed a kernel-based reconstruction framework to accommodate time-evolving signals over possibly time-evolving topologies, leveraging spatio-temporal dynamics of the observed data.

In this work we propose the following contributions: (a) We propose a centralized RLS reconstruction strategy that collects data over the graph through a probabilistic sampling mechanism, where each node in the graph has an assigned probability to be sampled at each time instant; (b) We formulate an optimization problem to select the sampling probability at each node in the graph, enforcing sparsity from one side, while also guaranteeing signal reconstruction and a prescribed steady-state performance; (c) we derive a distributed RLS strategy exploiting the alternating direction method of multipliers (ADMM) [19], which is shown to be convergent to the centralized method. Numerical results validate our findings, and illustrate the performance of the proposed strategies.

II. Background on GSP

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ composed of a set of N nodes $\mathcal{V} = \{1, 2, ..., N\}$, and a set of (weighted) edges $\mathcal{E} = \{a_{ij}\}_{i,j\in\mathcal{V}}$, such that $a_{ij} > 0$, if nodes j and i share a link, or $a_{ij} = 0$, otherwise. The graph adjacency matrix and the graph Laplacian are respectively $\mathbf{A} = \{a_{ij}\}, i, j = 1, \ldots, N$ and $\mathbf{L} = \text{diag}(\mathbf{1}^T\mathbf{A}) - \mathbf{A}$. Since $\mathcal{G}$ is undirected, the graph Laplacian can be eigendecomposed as $\mathbf{L}=\mathbf{U}\mathbf{\Lambda}\mathbf{U}^H$, where $\mathbf{U}$ contains the eigenvectors in its columns, whereas $\mathbf{\Lambda}$ is a diagonal matrix containing the non-negative eigenvalues.

A signal $\boldsymbol{x}$ defined over a graph $\mathcal{G}$ is a mapping from the vertex set to the set of complex numbers, i.e. $\boldsymbol{x} : \mathcal{V} \to \mathbb{C}$. In several applications like brain signal monitoring, big data, or biological networks [1], the signal $\boldsymbol{x}$ admits an exact (or approximate) compact representation, i.e., it can be cast as:

$$\boldsymbol{x} = \mathbf{U}\boldsymbol{s} \tag{1}$$

where $\boldsymbol{s}$ is exactly (or approximately) sparse. The GFT $\boldsymbol{s}$ of a signal $\boldsymbol{x}$ is defined as the projection onto the set of vectors $\{\boldsymbol{u}_i\}_{i=1,\ldots,N}$ [1], i.e., $\boldsymbol{s} = \mathbf{U}^H\boldsymbol{x}$. Alternative definitions for the GFT are proposed in [1], [2], [6]. In this paper, we follow the approach from [1], but the theory can be easily extended to general graph shift operators.

The work of Paolo Di Lorenzo was supported by the "Fondazione Cassa di Risparmio di Perugia".

ISBN 978-0-9928626-7-1 © EURASIP 2017

III. RLS Estimation of Graph Signals

We consider a bandlimited graph signal $\boldsymbol{x}^o = \{x_i^o\}_{i=1}^N \in \mathbb{C}^N$, with graph spectral content perfectly localized only over a limited set of frequency indices $\mathcal{F}$. Assuming the frequency support $\mathcal{F}$ to be known a priori, from (1), the graph signal $\boldsymbol{x}^o$ can be cast as the linear model:

$$\boldsymbol{x}^o = \mathbf{U}_{\mathcal{F}} \boldsymbol{s}^o, \tag{2}$$

where $\mathbf{U}_{\mathcal{F}} \in \mathbb{C}^{N \times |\mathcal{F}|}$ denotes the subset of columns of matrix $\mathbf{U}$ in (1) corresponding to the frequency set $\mathcal{F}$, and $\boldsymbol{s}^o \in \mathbb{C}^{|\mathcal{F}| \times 1}$ collects the coefficients of the GFT of $\boldsymbol{x}^o$ over the set $\mathcal{F}$. The graph signal is then sampled over a time-varying subset of vertices, thus resulting in streaming and noisy observations available for processing at each time instant t, which can be expressed as:

$$\boldsymbol{y}(t) = \mathbf{D}(t)\left(\boldsymbol{x}^o + \boldsymbol{v}(t)\right) = \mathbf{D}(t)\mathbf{U}_{\mathcal{F}}\boldsymbol{s}^o + \mathbf{D}(t)\boldsymbol{v}(t) \tag{3}$$

where $\mathbf{D}(t) = \mathrm{diag}\{d_i(t)\}_{i=1}^N$, with $d_i(t) \in \{0, 1\}$ denoting a Bernoulli random variable, which is equal to 1 if node i is sampled at time t, and 0 otherwise; and $\boldsymbol{v}(t)$ is zero-mean observation noise, assumed to be spatially and temporally independent, with covariance matrix $\mathbf{C}_v = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_N^2)$.

The learning goal consists in recovering the GFT vector $\boldsymbol{s}^o$ from the partial and streaming observations $\boldsymbol{y}(t)$ in (3). Using an RLS approach [20], the optimal (centralized) estimate for $\boldsymbol{s}^o$ at time t, say, $\widehat{\boldsymbol{s}}_c(t)$, is given by the solution of the following optimization problem:

$$\min_{\boldsymbol{s}} \sum_{\tau=1}^{t} \beta^{t-\tau} \left\| \mathbf{D}(\tau)(\boldsymbol{y}(\tau) - \mathbf{U}_{\mathcal{F}}\boldsymbol{s}) \right\|_{\mathbf{C}_v^{-1}}^2 + \beta^t \|\boldsymbol{s}\|_{\mathbf{\Pi}}^2 \tag{4}$$

where $0 \ll \beta \leq 1$ is the exponential forgetting factor, $\|\boldsymbol{s}\|_{\mathbf{\Pi}}^2 = \boldsymbol{s}^H \mathbf{\Pi} \boldsymbol{s}$ with $\mathbf{\Pi} \succeq \mathbf{0}$ denoting a regularization matrix, and we have exploited the fact that $\mathbf{D}(t)$ is an idempotent matrix for all t. Typically, $\mathbf{\Pi} = \delta \mathbf{I}$, where $\delta > 0$ is small [20]. Solving (4) and using (2), the optimal estimate for the graph signal $\boldsymbol{x}^o$ at time t is given by:

$$\widehat{\boldsymbol{x}}_c(t) \;=\; \mathbf{U}_{\mathcal{F}} \widehat{\boldsymbol{s}}_c(t) \;=\; \mathbf{U}_{\mathcal{F}} \mathbf{\Psi}^{-1}(t) \boldsymbol{\psi}(t) \tag{5}$$

where

$$\mathbf{\Psi}(t) \;=\; \sum_{\tau=1}^{t} \beta^{t-\tau} \mathbf{U}_{\mathcal{F}}^H \mathbf{D}(\tau) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} + \beta^t \mathbf{\Pi}, \tag{6}$$

$$\boldsymbol{\psi}(t) \;=\; \sum_{\tau=1}^{t} \beta^{t-\tau} \mathbf{U}_{\mathcal{F}}^H \mathbf{D}(\tau) \mathbf{C}_v^{-1} \boldsymbol{y}(\tau). \tag{7}$$

The regularization term in (6) avoids invertibility issues in (5), especially at early values of t. Given the structure of the recursion of $\mathbf{\Psi}(t)$ and $\boldsymbol{\psi}(t)$ in (6)-(7), we obtain

$$\mathbf{\Psi}(t) \;=\; \beta\, \mathbf{\Psi}(t-1) + \mathbf{U}_{\mathcal{F}}^H \mathbf{D}(t) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} \tag{8}$$

$$\boldsymbol{\psi}(t) \;=\; \beta\, \boldsymbol{\psi}(t-1) + \mathbf{U}_{\mathcal{F}}^H \mathbf{D}(t) \mathbf{C}_v^{-1} \boldsymbol{y}(t) \tag{9}$$

with $\mathbf{\Psi}(0) = \mathbf{\Pi}$, which recursively update both $\mathbf{\Psi}(t)$ and $\boldsymbol{\psi}(t)$ given their previous values. Thus, the main steps of the algorithm, named RLS on graphs, are illustrated in Algorithm 1, which has computational complexity of the order of $O(|\mathcal{F}|^3)$, due to the presence of the inverse operation $\mathbf{\Psi}^{-1}(t)$ in (10). Since typically we have $|\mathcal{F}| \ll N$, the cost $O(|\mathcal{F}|^3)$ is often affordable. The performance of the RLS method in (10) heavily depend on the properties of the random sampling operator $\mathbf{D}(t)$. Thus, in the sequel, we will show how to optimally design the graph sampling strategy in order to guarantee graph signal reconstruction and a target steady-state performance of the proposed RLS algorithm.

Algorithm 1: RLS on Graphs

Start with random $\boldsymbol{\psi}(0)$, and $\mathbf{\Psi}(0) = \mathbf{\Pi}$. For $t > 0$, repeat:

$$\begin{aligned} \mathbf{\Psi}(t) \;&=\; \beta\, \mathbf{\Psi}(t-1) + \mathbf{U}_{\mathcal{F}}^H \mathbf{D}(t) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} \\ \boldsymbol{\psi}(t) \;&=\; \beta\, \boldsymbol{\psi}(t-1) + \mathbf{U}_{\mathcal{F}}^H \mathbf{D}(t) \mathbf{C}_v^{-1} \boldsymbol{y}(t) \\ \widehat{\boldsymbol{x}}_c(t) \;&=\; \mathbf{U}_{\mathcal{F}} \mathbf{\Psi}^{-1}(t) \boldsymbol{\psi}(t) \end{aligned} \tag{10}$$

A. Mean-Square Performance

The study of the mean-square performance of RLS adaptive filters is rather challenging [20]. To allow a tractable analysis, we will hinge on the following ergodicity assumption.

Assumption 1 (Ergodicity): $\exists t_0$ such that for all $t > t_0$, $\mathbf{\Psi}(t)$ in (6) can be replaced by $\overline{\mathbf{\Psi}} = \mathrm{E}\mathbf{\Psi}(t)$. ■

Assumption 1 states that the time average of a random process can be replaced by its expected value, for t sufficiently large. This assumption is very common for RLS-type algorithms, see, e.g., [20], and leads to good approximations in practice. Thus, under Assumption 1, the steady state expression for matrix $\mathbf{\Psi}(t)$ can be approximated as:

$$\lim_{t\to\infty} \mathbf{\Psi}(t) \;\simeq\; \lim_{t\to\infty} \mathrm{E}\mathbf{\Psi}(t) = \frac{1}{1-\beta} \mathbf{U}_{\mathcal{F}}^H \mathrm{diag}(\boldsymbol{p}) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} \tag{11}$$

where $\boldsymbol{p} = (p_1, \ldots, p_N)^T \in \mathbb{R}^N$ represents the sampling probability vector, with $p_i = \mathrm{E}\{d_i(t)\}$, $i = 1, \ldots, N$, denoting the probability that node i samples at time t. From (5) and (11), we deduce that asymptotic reconstruction of $\boldsymbol{x}^o$ is possible only if the positive (semi)definite matrix $\mathbf{U}_{\mathcal{F}}^H\, \mathrm{diag}(\boldsymbol{p}) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}}$ is invertible (or full rank), i.e., iff

$$\lambda_{\min}\left(\mathbf{U}_{\mathcal{F}}^H\, \mathrm{diag}(\boldsymbol{p}) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} \right) > 0, \tag{12}$$

where $\lambda_{\min}(\mathbf{Y})$ is the minimum eigenvalue of matrix $\mathbf{Y}$. Let

$$\overline{\mathcal{S}} = \{ i = 1, \ldots, N \,|\, p_i > 0 \}$$

be the *expected sampling set* (i.e., the set of nodes of the graph that are sampled with a probability greater than zero). Then, condition (12) holds true for a large enough expected sampling set, i.e., if data is collected from a sufficiently large set of nodes with a probability greater than zero. In particular, a necessary condition to guarantee condition (12) is that $|\overline{\mathcal{S}}| \geq |\mathcal{F}|$, i.e., the number of nodes that are sampled with a probability greater than zero must be greater than or equal to the graph signal bandwidth.

Finally, we illustrate how the sampling probability vector $\boldsymbol{p}$ affects the mean-square behavior of Algorithm 1. The results are summarized in the following Theorem.

ISBN 978-0-9928626-7-1 © EURASIP 2017

Theorem 1: Assume model (3), Assumption 1, and condition (12) hold. Then, the RLS strategy (10) is mean-square stable, with mean-square deviation (MSD) given by

$$\begin{aligned}\text{MSD} &= \lim_{t\to\infty} \sup_{t} \mathbb{E}\|\widehat{\boldsymbol{x}}_c(t) - \boldsymbol{x}^o\|^2 \\ &= \frac{1-\beta}{1+\beta} \operatorname{Tr}\left[\left(\mathbf{U}_{\mathcal{F}}^H \operatorname{diag}(\boldsymbol{p}) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}}\right)^{-1}\right]. \end{aligned} \tag{13}$$

Proof. The proof is similar to derivations in [20], [21, eq. (29)], and is omitted due to lack of space. ■

B. Optimal Sampling Strategies

The mean-square analysis in Sec. III.A illustrates how the mean-square performance of algorithm (10) strongly depends on the sampling probability vector $\boldsymbol{p}$ [cf (13) and (12)]. Then, following a sparse sensing approach [22], [23], the goal of this section is to develop optimal sampling strategies aimed at selecting a sparse probability vector $\boldsymbol{p}$ that enables graph signal reconstruction from its samples, while guaranteing a target value of mean-square performance. To this aim, we propose the following design for the sampling strategy:

$$\begin{aligned} \min_{\boldsymbol{p}} \;& \mathbf{1}^T \boldsymbol{p} \\ \text{s.t.} \;& \operatorname{Tr}\left[\left(\mathbf{U}_{\mathcal{F}}^H \operatorname{diag}(\boldsymbol{p}) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}}\right)^{-1}\right] \leq \gamma \frac{1+\beta}{1-\beta} \\ & \mathbf{0} \leq \boldsymbol{p} \leq \boldsymbol{p}^{\max} \end{aligned} \tag{14}$$

The objective function in (14) enforces sparsity by minimizing the overall graph sampling rate. From (13), the first constraint in (14) imposes a target value of MSD less than or equal to a constant $\gamma > 0$. Finally, the last constraint limits $\boldsymbol{p}$ to lie in the box $\mathbf{0} \leq \boldsymbol{p} \leq \boldsymbol{p}^{\max}$, with $\boldsymbol{p}^{\max} = \{p_i^{\max}\}_{i=1}^N \leq \mathbf{1}$ denoting an upper bound on the sampling probabilities that might depend on external factors such as, e.g., limited energy, processing, and/or communication resources, failures, etc. It is important to remark that, differently from other sparse sensing methods proposed for batch estimation strategies [22], [23], the problem in (14) is convex, and its global solution can be found using efficient numerical tools [24].

IV. Distributed RLS Estimation of Graph Signals

In many real systems, data are not available at a single processing unit, since they are collected by separate agents of a distributed network. Furthermore, sending local information to a fusion center might be either impossible or not efficient, because of the large volume of data and dimension of the network, random impairments, latency and energy constraints, and/or privacy issues [21], [25], [26]. Motivated by these observations, in this section we extend the RLS strategy in Algorithm 1 to a distributed setting, where the nodes of the graph are connected through a sparse communication network. The topology of the communication graph does not necessarily coincide with that of the graph used to process the data. Let $\mathbf{B} = \{b_{ij}\}, i, j = 1, \ldots, N$ be the adjacency matrix of the communication graph. To ensure the diffusion of information over the network, we assume the following.

Assumption 2 (Topology): The communication graph is symmetric and connected; i.e., there exists an undirected path connecting any two vertices of the network. ■

To derive distributed solution methods for problem (4), let us introduce local copies $\{\boldsymbol{s}_i\}_{i=1}^N$ of the global variable $\boldsymbol{s}$, and recast problem (4) in the following equivalent form:

$$\begin{aligned} \min_{\{\boldsymbol{s}_i\}_{i=1}^N} \;& \sum_{i=1}^N \sum_{\tau=1}^t \beta^{t-\tau} \widetilde{d}_i(t) (y_i(t) - \boldsymbol{u}_{\mathcal{F},i}^H \boldsymbol{s}_i)^2 + \frac{\beta^t}{N} \sum_{i=1}^N \|\boldsymbol{s}_i\|_{\boldsymbol{\Pi}}^2 \\ \text{s.t.} \;& \boldsymbol{s}_i = \boldsymbol{s}_j \quad \text{for all } i \in \mathcal{V},\, j \in \mathcal{N}_i, \end{aligned} \tag{15}$$

where $\mathcal{N}_i = \{j \in \mathcal{V} \,|\, b_{ij} > 0\}$ is the neighborhood of agent i, $\boldsymbol{u}_{\mathcal{F},i}^H$ is the i-th row of matrix $\mathbf{U}_{\mathcal{F}}$, and $\widetilde{d}_i(t) = d_i(t)/\sigma_i^2$.

Proposition 1: Let $\{\widehat{\boldsymbol{s}}_i(t)\}_{i=1}^N$ be the optimal solutions of problem (15). Under Assumption 2, problems (4) and (15) are equivalent, i.e., $\widehat{\boldsymbol{s}}_i(t) = \widehat{\boldsymbol{s}}_c(t)$ for all $i \in \mathcal{V}$ and $t \geq 0$. ■

Letting $\boldsymbol{s} = \{\boldsymbol{s}_i\}_{i=1}^N$ and $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_{ij}\}_{i\in\mathcal{V}}^{j\in\mathcal{N}_i}$, the augmented Lagrangian for problem (15) writes as:

$$\begin{aligned} \mathcal{L}_a(\boldsymbol{s}, \boldsymbol{\lambda}) = & \sum_{i=1}^N \sum_{\tau=1}^t \beta^{t-\tau} \widetilde{d}_i(\tau) (y_i(\tau) - \boldsymbol{u}_{\mathcal{F},i}^H \boldsymbol{s}_i)^2 + \frac{\beta^t}{N} \sum_{i=1}^N \|\boldsymbol{s}_i\|_{\boldsymbol{\Pi}}^2 \\ & + \sum_{i=1}^N \sum_{j\in\mathcal{N}_i} \boldsymbol{\lambda}_{ij}^T (\boldsymbol{s}_i - \boldsymbol{s}_j) + \frac{\varrho}{4} \sum_{i=1}^N \sum_{j\in\mathcal{N}_i} \|\boldsymbol{s}_i - \boldsymbol{s}_j\|^2, \end{aligned} \tag{16}$$

where $\varrho > 0$ is a positive regularization coefficient. Since the augmented Lagrangian function in (16) is strictly convex for all t, we can employ the alternating direction method of multipliers (ADMM) to solve problem (15), see, e.g., [19]. Letting k be the iteration index of ADMM, the first step updates the local estimates as:

$$\widehat{\boldsymbol{s}}(t, k+1) = \operatorname*{argmin}_{\boldsymbol{s}} \mathcal{L}_a(\boldsymbol{s}, \boldsymbol{\lambda}(t,k)). \tag{17}$$

From the separable structure of (16), problem (17) can be split into N subproblems:

$$\begin{aligned} \widehat{\boldsymbol{s}}_i(t, k+1) = & \operatorname*{argmin}_{\boldsymbol{s}_i} \sum_{\tau=1}^t \beta^{t-\tau} \widetilde{d}_i(\tau) (y_i(\tau) - \boldsymbol{u}_{\mathcal{F},i}^H \boldsymbol{s}_i)^2 \\ & + \frac{\beta^t}{N} \|\boldsymbol{s}_i\|_{\boldsymbol{\Pi}}^2 + \frac{1}{2} \sum_{j\in\mathcal{N}_i} \left[\boldsymbol{\lambda}_{ij}(t,k) - \boldsymbol{\lambda}_{ji}(t,k)\right]^T \boldsymbol{s}_i \\ & + \frac{\varrho}{2} \sum_{j\in\mathcal{N}_i} \|\boldsymbol{s}_i - \widehat{\boldsymbol{s}}_j(t,k)\|^2 \end{aligned} \tag{18}$$

Since each local subproblem corresponds to an unconstrained quadratic minimization, they all admit closed-form solutions

$$\begin{aligned} \widehat{\boldsymbol{s}}_i(t, k+1) = & \left(\boldsymbol{\Psi}_i(t) + \varrho |\mathcal{N}_i| \boldsymbol{I}\right)^{-1} \Bigg[\boldsymbol{\psi}_i(t) + \varrho \sum_{j\in\mathcal{N}_i} \widehat{\boldsymbol{s}}_j(t,k) \\ & - \frac{1}{2} \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}_{ij}(t,k) - \boldsymbol{\lambda}_{ji}(t,k))\Bigg], \end{aligned} \tag{19}$$

where, setting $\boldsymbol{\Psi}_i(0) = \boldsymbol{\Pi}$, we have

$$\boldsymbol{\Psi}_i(t) = \beta\, \boldsymbol{\Psi}_i(t-1) + \widetilde{d}_i(t) \boldsymbol{u}_{\mathcal{F},i} \boldsymbol{u}_{\mathcal{F},i}^H, \tag{20}$$

$$\boldsymbol{\psi}_i(t) = \beta\, \boldsymbol{\psi}_i(t-1) + \widetilde{d}_i(t) y_i(t) \boldsymbol{u}_{\mathcal{F},i}. \tag{21}$$

ISBN 978-0-9928626-7-1 © EURASIP 2017

Algorithm 2: Distributed RLS on Graphs

Start with $\{\boldsymbol{\psi}_i(0)\}_{i=1}^N$, $\{\boldsymbol{s}_i(0,0)\}_{i=1}^N$, $\{\boldsymbol{\lambda}_{ij}(0,0)\}_{i\in\mathcal{V}}^{j\in\mathcal{N}_i}$ cl
at random, and set $\{\boldsymbol{\Psi}_i(0)\}_{i=1}^N = \boldsymbol{\Pi}/N$, and $\varrho > 0$.

for $t > 0$ **do**

All $i \in \mathcal{V}$: update $\boldsymbol{\Psi}_i(t)$ and $\boldsymbol{\psi}_i(t)$ using (20) and (2

for $k = 0, \ldots, K-1$ **do**

All $i \in \mathcal{V}$: transmit $\boldsymbol{\lambda}_{ij}(t,k)$ to each $j \in \mathcal{N}_i$;

All $i \in \mathcal{V}$: update $\widehat{\boldsymbol{s}}_i(t,k+1)$ using (19);

All $i \in \mathcal{V}$: transmit $\widehat{\boldsymbol{s}}_i(t,k+1)$ to neighbors in $\mathcal{N}$

All $i \in \mathcal{V}$: update $\{\boldsymbol{\lambda}_{ij}(t,k+1)\}_{j\in\mathcal{N}_i}$ using (22)

end

end

Finally, the second step of the ADMM algorithm updates the Lagrange multipliers as:

$$\boldsymbol{\lambda}_{ij}(t,k+1) = \boldsymbol{\lambda}_{ij}(t,k) + \frac{\varrho}{2}\left(\widehat{\boldsymbol{s}}_j(t,k+1) - \widehat{\boldsymbol{s}}_i(t,k+1)\right), \tag{22}$$

for $i \in \mathcal{V}$, $j \in \mathcal{N}_i$. Recursions (19) and (22) constitute the ADMM-based Distributed RLS algorithm (DRLS), whereby all sensors $i \in \mathcal{V}$ update their local estimate $\widehat{\boldsymbol{s}}_i$ and their multipliers $\{\boldsymbol{\lambda}_{ij}\}_{j\in\mathcal{N}_i}$, which can be arbitrarily initialized. Then, all the steps of the distributed RLS strategy for adaptive reconstruction of graph signals are summarized in Algorithm 2, which also describes communications of multipliers and local estimates taking place within local neighborhoods. By direct application of [19, Sec. 2.2.1], the following convergence result applies to Algorithm 2.

Proposition 2: For arbitrarily initialized $\{\boldsymbol{\lambda}_{ij}(t,0)\}_{i\in\mathcal{V}}^{j\in\mathcal{N}_i}$, $\boldsymbol{s}_i(t,0)$ and $\varrho > 0$; the local estimates $\widehat{\boldsymbol{s}}_i(t,k)$ generated by Algorithm 2 reach consensus as $k \to \infty$; i.e.,

$$\lim_{k\to\infty} \widehat{\boldsymbol{s}}_i(t,k) = \widehat{\boldsymbol{s}}_c(t), \quad \text{for all } i \in \mathcal{V}. \tag{23}$$

Proposition 2 asserts that Algorithm 2 yields a sequence of local estimates that converges to the global estimate produced by Algorithm 1, as $k \to \infty$, or, pragmatically for large enough k. However, when the network is deployed to track a time-varying graph signal, one cannot afford large delays in-between consecutive sensing instants. In this case, we can run a single consensus iteration per acquired observation, i.e., $K = 1$ in Algorithm 2, thus making the method suitable for operation in nonstationary environments.

Remark 1: In this work, we assume that processing and communication graphs have in general distinct topologies. We remark that both graphs play an important role in Algorithm 2. First, the processing graph determines the structure of the regression data $\boldsymbol{u}_{\mathcal{F},i}^H$, which are the rows of the matrix $\mathbf{U}_{\mathcal{F}}$, whose columns are the eigenvectors of the Laplacian matrix associated with the set of support frequencies $\mathcal{F}$. Then, the topology of the communication graph determines how the processed information is diffused over the network of agents. This illustrates how, when reconstructing graph signals in a distributed manner, we have to take into account both the processing and communication aspects of the problem. ■

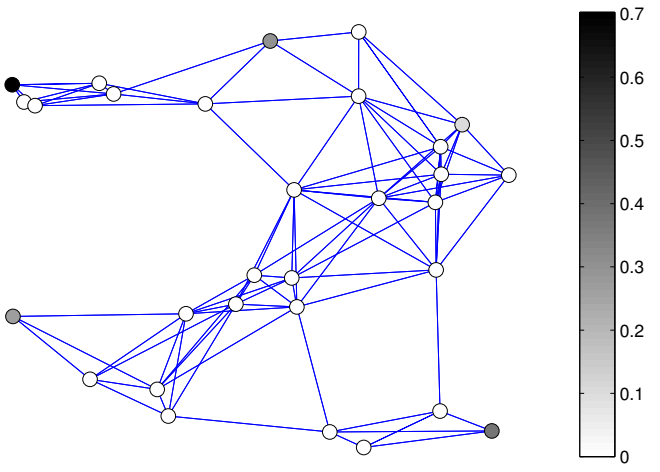

Fig. 1: Graph and optimal probability vector for $\gamma = 10^{-2}$.

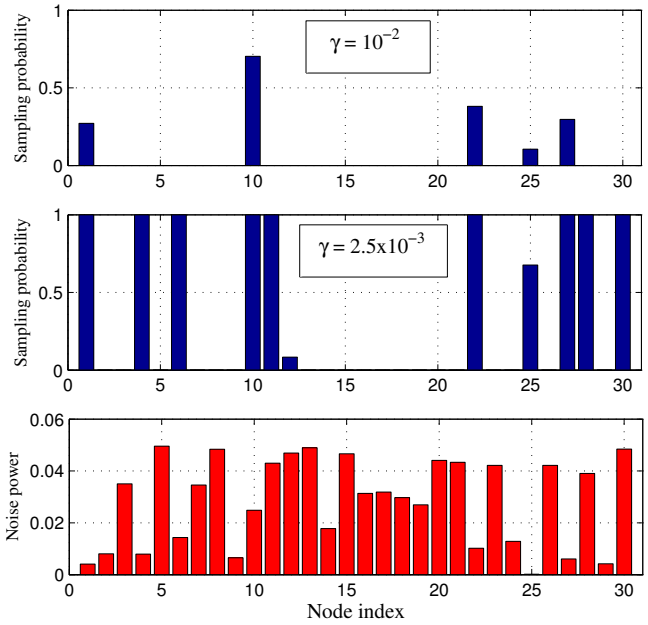

Fig. 2: Optimal probability vector, and noise variance, versus node index, for different values of γ.

V. Numerical Results

We consider a graph signal composed of $N = 30$ nodes, with topology depicted in Fig. 1. The graph signal can be represented using only the first five eigenvectors of the Laplacian matrix of the graph in Fig. 1, i.e. $|\mathcal{F}| = 5$. The Gaussian observation noise in (3) is zero-mean, with variance terms at each node illustrated in Fig. 2 (bottom).

An example of optimal probabilistic sampling, obtained solving Problem (14) with $\gamma = 10^{-2}$, $\beta = 0.95$, and $p_i^{\max} = 1$ for all i, is illustrated in Fig. 1, where the color (in gray scale) of the vertexes denotes the sampling probability. Problem (14) was solved using the CVX sedumi software [24]. As we can notice from Fig. 1, the method selects a very sparse probability vector in order to guarantee the target MSD value. As a further example, in Fig. 2 (top and middle), we report the optimal probability vector obtained solving Problem (14) for different values of γ. In all cases, the constraint on the MSD is attained

ISBN 978-0-9928626-7-1 © EURASIP 2017

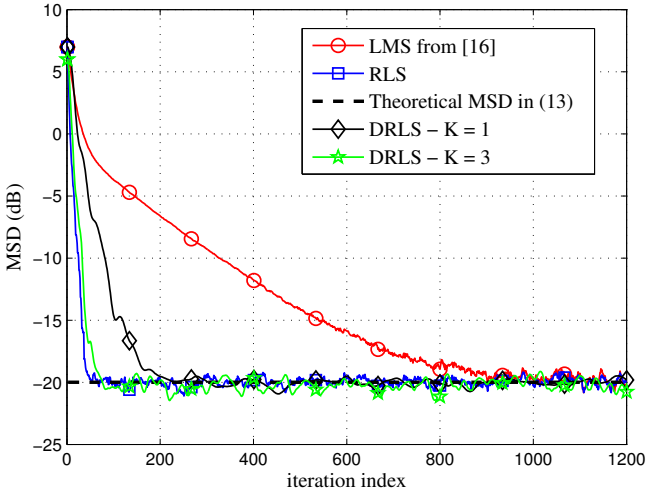

Fig. 3: MSD versus iteration index, for different algorithms.

strictly. From Fig. 2 (top and middle), as expected, we notice how the method enlarges the expected sampling set if we have a stricter requirement on the MSD, assigning large sampling probabilities to nodes having low noise variances.

Finally, in Fig. 3, we report the temporal behavior of the MSD obtained using three different algorithms: the proposed RLS method in Algorithm 1; the LMS algorithm for graph signals from [16]; and the DRLS strategy in Algorithm 2, considering different numbers K of inner consensus iterations, i.e., $K = 1$ and $K = 3$. The parameters of the RLS are $\beta = 0.95$, $\gamma = 10^{-2}$, and the sampling strategy was selected solving problem (14). The parameters of the other algorithms are chosen to match the steady-state performance of the RLS method. The communication graph for the DRLS algorithm is chosen as a connected sub-graph of the processing graph in Fig. 1. The theoretical results in (13) are also reported as a benchmark. As we can see from Fig. 3, the theoretical expression in (13) well predicts the numerical results. Also, as expected, the RLS strategy is much faster than the LMS method proposed in [16], at the cost of a higher complexity. Finally, increasing the number K of inner consensus iterations, we notice how the behavior of the DRLS algorithm approaches the performance of the centralized RLS, at the cost of a larger number of exchanged parameters over the network.

VI. Conclusions

In this paper we have first introduced a novel RLS estimation strategy for graph signals based on a probabilistic sampling mechanism over the graph. Then, we have derived a mean-square analysis that shed light on how the probabilistic sampling strategy affects the performance of the RLS algorithm. On the basis of such analysis, we have formulated an optimization problem that selects a sparse sampling probability vector that guarantees a prescribed performance level in terms of MSD. Finally, a distributed RLS strategy is derived exploiting the ADMM-based decomposition method, and is shown to be convergent to the centralized method. Numerical results illustrate the good performance of proposed methods for distributed adaptive reconstruction of graph signals.

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 Proc. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[2] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. on Signal Proc.*, vol. 61, pp. 1644–1656, 2013.

[3] ——, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Proc. Mag.*, vol. 31, no. 5, pp. 80–90, 2014.

[4] I. Z. Pesenson, "Sampling in Paley-Wiener spaces on combinatorial graphs," *Trans. of the American Mathematical Society*, vol. 360, no. 10, pp. 5603–5627, 2008.

[5] X. Zhu and M. Rabbat, "Approximating signals supported on graphs," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 3921–3924.

[6] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. on Signal Proc.*, vol. 63, pp. 6510–6523, Dec. 2015.

[7] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[8] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2786–2799, 2012.

[9] ——, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4673–4685, 2013.

[10] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Transactions on Signal Processing*, vol. 65, pp. 274–288, 2017.

[11] S. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 5445–5449.

[12] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, 2016.

[13] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. on Signal Proc.*, vol. 63, no. 9, pp. 2432–2444, 2015.

[14] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1832–1843, 2016.

[15] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 764–778, 2017.

[16] P. Di Lorenzo, S. Barbarossa, P. Banelli, and S. Sardellitti, "Adaptive least mean squares estimation of graph signals," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 555–568, 2016.

[17] P. Di Lorenzo, P. Banelli, S. Barbarossa, and S. Sardellitti, "Distributed adaptive learning of graph signals," *To appear in IEEE Transactions on Signal Processing*, 2017.

[18] D. Romero, V. N. Ioannidis, and G. B. Giannakis, "Kernel-based reconstruction of space-time functions on dynamic graphs," *arXiv preprint arXiv:1612.03615*, 2016.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[20] A. H. Sayed, *Adaptive filters*. John Wiley & Sons, 2011.

[21] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, 2008.

[22] S. P. Chepuri and G. Leus, "Sparsity-promoting sensor selection for non-linear measurement models," *IEEE Transactions on Signal Processing*, vol. 63, no. 3, pp. 684–698, 2015.

[23] ——, "Sparse sensing for distributed detection," *IEEE Transactions on Signal Processing*, vol. 64, no. 6, pp. 1446–1460, 2016.

[24] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[25] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4583–4588, 2009.

[26] G. Mateos and G. B. Giannakis, "Distributed recursive least-squares: Stability and performance analysis," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3740–3754, 2012.

ISBN 978-0-9928626-7-1 © EURASIP 2017