Online Recovery of Time-varying Signals Defined over Dynamic Graphs

Paolo Di Lorenzo and Elena Ceci
Department of Information Engineering, Electronics, and Telecommunications, Sapienza University of Rome,
Via Eudossiana 18, 00184 Rome, Italy. E-mail: {paolo.dilorenzo,elena.ceci}@uniroma1.it

***Abstract*—The goal of this work is to devise least mean square (LMS) strategies for online recovery of time-varying signals defined over dynamic graphs, which are observed over a (randomly) time-varying subset of vertices. We also derive a mean-square analysis illustrating the effect of graph variations and sampling on the reconstruction performance. Finally, an optimization strategy is developed in order to design the sampling probability at each node in the graph, with the aim of finding the best tradeoff between steady-state performance, graph sampling rate, and learning rate of the proposed method. Numerical simulations carried out over both synthetic and real data illustrate the good performance of the proposed learning strategies.**

***Index Terms*—Graph signal processing, online learning, sampling on graphs, time-varying graphs.**

I. Introduction

In recent years there was a large interest in developing novel analysis and processing methods for signals defined over irregular discrete domains, typically represented by a graph, see, e.g., [1]–[3]. A fundamental task in graph signal processing (GSP) is to infer the values of a signal by interpolating the samples collected from a known set of vertices. To solve this task, machine learning methods typically exploit smoothness of the graph signal over the graph, see, e.g., [4]–[6], whereas GSP usually hinges on the bandlimited signal model, i.e., signals that belong to the span of some eigenvectors of the graph shift operator, see, e.g., [7], [8]. A first seminal contribution to sampling/interpolation theory in GSP is given by [9]; the approach was then extended in [10], [11]. The work in [7] derives conditions guaranteeing stable and unique reconstruction of bandlimited graph signals. Reference [8] connects uncertainty principle and sampling of graph signals. The work in [12] proposes the so called aggregation sampling, which involves successively shifting the graph signal and aggregating its values at a given node. The work in [13] devises strategies to design the sampling set based on powers of the variation operator. Greedy sampling methods with performance close to optimality are proposed in [14], and also randomized sampling strategies are considered in [15], [16]. Finally, adaptive sampling and interpolation methods that are capable to handle time-varying graph signals have been recently proposed in [17]–[20]. Specifically, [17] proposes an LMS algorithm enabling online recovery and tracking from a small number of smartly sampled observations. The LMS method in [17] is then extended to the distributed setting in [18], and to incorporate a probabilistic sampling scheme in [19]. Finally, the work in [20] proposes a reconstruction framework to infer time-evolving signals over possibly time-evolving topologies, leveraging kernels and spatio-temporal dynamics of the observed data.

Contribution. The goal of this work is to propose an online learning method for time-varying signals defined over dynamic graphs, which appear in many applications such as, e.g., brain networks, communication networks, transportation networks, etc. Incorporating knowledge related to how the graph varies over time in GSP algorithm is expected to lead to enhanced performance of filtering/recovery tasks, see, e.g., [20], [21]. To this aim, we extend the methods proposed in [17]–[19] to incorporate a random graph variation model, where a nominal graph is perturbed such that each edge has an assigned probability to be deleted (or added) at each time instant. In particular, to derive an efficient algorithm and simplify the analysis, we assume a small perturbation model in order to quantify the effect of the graph variations in terms of perturbation of the Laplacian eigenvectors. We also derive a mean-square analysis that illustrates the role played by the sampling and edge deletion probabilities on the performance. Then, we design a sampling strategy aimed at minimizing the graph sampling rate while imposing learning performance constraints. Finally, we assess the performance of proposed strategy via numerical simulations.

Notation. We indicate scalars by normal letters (e.g., a); vector variables with bold lowercase letters (e.g., $\boldsymbol{a}$) and matrix variables with bold uppercase letters (e.g., $\mathbf{A}$). $\lambda_i(\mathbf{A})$ represents the i-th eigenvalue of $\mathbf{A}$. The trace of matrix $\mathbf{A}$ is indicated with $\mathrm{Tr}(\mathbf{A})$; $\mathrm{diag}(\boldsymbol{a})$ is a diagonal matrix having $\boldsymbol{a}$ as main diagonal. The superscript ^{H} denotes the hermitian operator. $\mathbb{E}\{\cdot\}$ represents the expectation operator. A set of elements is denoted by a calligraphic letter (e.g., $\mathcal{S}$), and $|\mathcal{S}|$ represents the cardinality of set $\mathcal{S}$.

II. Background

In this section, we recall some basic definitions and theoretical results that will be largely used along the paper.

A. GSP Basic Tools

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is composed of a set of N vertices $\mathcal{V} = \{1, 2, ..., N\}$ and a set of weighted edges $\mathcal{E}$. A signal $\boldsymbol{x}$ defined over $\mathcal{G}$ associates a complex number (vector) to each vertex belonging to the set $\mathcal{V}$, i.e., $x : \mathcal{V} \to \mathbb{C}$. The structure of $\mathcal{G}$ is typically described by a graph-shift operator

ISBN 978-90-827970-1-5 © EURASIP 2018

$\mathbf{S}$, which is an $N \times N$ matrix, whose elements can be non-zero only if $i = j$ or the link $(j, i) \in \mathcal{E}$. The sparsity pattern of matrix $\mathbf{S}$ captures the local structure of $\mathcal{G}$, and common choices for $\mathbf{S}$ are the adjacency matrix [2], the Laplacian [1], and its generalizations [13]. In this paper, we choose the undirected Laplacian $\mathbf{L}$ as the graph-shift operator, which admits the decomposition $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H$ for some eigenvector matrix $\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N]$ and diagonal matrix $\mathbf{\Lambda}$. Then, the Graph Fourier Transform (GFT) of a signal $\boldsymbol{x}$ is defined as its projection onto the set of eigenvectors $\{\boldsymbol{u}_i\}_{i=1,\ldots,N}$ [1], i.e., $\mathrm{GFT}(\boldsymbol{x}) = \mathbf{U}^H\boldsymbol{x}$. Perfect recovery of a graph signal from its samples is possible if $\boldsymbol{x}$ is *bandlimited* in the graph frequency domain, i.e., we have:

$$\boldsymbol{x} = \mathbf{U}_{\mathcal{F}}\boldsymbol{s}_{\mathcal{F}}, \tag{1}$$

where $\mathbf{U}_{\mathcal{F}} \in \mathbb{C}^{N\times|\mathcal{F}|}$ collects the graph Fourier vectors associated with a subset of frequency indices $\mathcal{F}$, and $\boldsymbol{s}_{\mathcal{F}} \in \mathbb{C}^{|\mathcal{F}|}$ are the corresponding graph frequency coefficients. Thus, $\mathcal{F}$ denotes the support of the signal in the graph Fourier domain.

B. Small Perturbation Analysis on Graphs

In this paragraph we recall approximated formulas for the eigendecomposition of a perturbed Laplacian $\mathbf{L} + \Delta\mathbf{L}$, where $\mathbf{L}$ denotes the Laplacian associated with a nominal graph $\mathcal{G}$, and $\Delta\mathbf{L}$ is a matrix representing a small perturbation of $\mathcal{G}$, see, e.g. [22], [23]. We denote by $\widetilde{\lambda}_i = \lambda_i + \Delta\lambda_i$ the perturbed i-th eigenvalue, and by $\widetilde{\boldsymbol{u}}_i = \boldsymbol{u}_i + \Delta\boldsymbol{u}_i$ its associated eigenvector. If one link $m \in \mathcal{E}$ fails, the perturbation matrix can be written as $\Delta\mathbf{L}(m) = -\boldsymbol{a}_m\boldsymbol{a}_m^T$, where $\boldsymbol{a}_m = [a_m(1), \cdots, a_m(N)]^T \in \mathbb{R}^N$ has all entries equal to zero, except for $a_m(i_m) = 1$ and $a_m(f_m) = -1$, with i_m and f_m denoting the indexes of the initial and final vertices of the failing edge. In case of addition of a new edge, the perturbation matrix is simply the opposite of the previous expression, i.e., $\Delta\mathbf{L}(m) = \boldsymbol{a}_m\boldsymbol{a}_m^T$. Also, the perturbation matrix associated with the simultaneous deletion of a small set of edges is $\Delta\mathbf{L} = -\sum_{m\in\mathcal{E}_p} \boldsymbol{a}_m\boldsymbol{a}_m^T$, where $\mathcal{E}_p$ denotes the set of perturbed edges. Exploiting a first order approximation, in the case all eigenvalues of the nominal Laplacian $\mathbf{L}$ are distinct, the following formulas for the perturbed eigenvalues and eigenvectors hold [22], [23]:

$$\widetilde{\lambda}_i \simeq \lambda_i + \boldsymbol{u}_i^T \Delta\mathbf{L}\, \boldsymbol{u}_i, \tag{2}$$

$$\widetilde{\boldsymbol{u}}_i \simeq \boldsymbol{u}_i + \sum_{j\neq i} \frac{\boldsymbol{u}_j^T \Delta\mathbf{L}\, \boldsymbol{u}_i}{\lambda_i - \lambda_j}\boldsymbol{u}_j. \tag{3}$$

Thus, from (2)-(3), the perturbations associated with the deletion of one edge $m \in \mathcal{E}$ are:

$$\begin{aligned}\Delta\lambda_i(m) &= \boldsymbol{u}_i^T\Delta\mathbf{L}(m)\boldsymbol{u}_i = -\boldsymbol{u}_i^T\boldsymbol{a}_m\boldsymbol{a}_m^T\boldsymbol{u}_i \\ &= -[u_i(f_m) - u_i(i_m)]^2 \end{aligned} \tag{4}$$

$$\begin{aligned}\Delta\boldsymbol{u}_i(m) &= \sum_{j=2,j\neq i}^{N} \frac{\boldsymbol{u}_j^T\Delta\mathbf{L}(m)\boldsymbol{u}_i}{\lambda_i - \lambda_j}\boldsymbol{u}_j = -\sum_{j=2,j\neq i}^{N} \frac{\boldsymbol{u}_j^T\boldsymbol{a}_m\boldsymbol{a}_m^T\boldsymbol{u}_i}{\lambda_i - \lambda_j}\boldsymbol{u}_j \\ &= \sum_{j=2,j\neq i}^{N} \frac{[u_j(i_m) - u_j(f_m)][u_i(f_m) - u_i(i_m)]}{\lambda_i - \lambda_j}\boldsymbol{u}_j. \end{aligned} \tag{5}$$

Within the limits of validity of this first order analysis, the perturbation resulting from the deletion of multiple edges in $\mathcal{E}_p$ is the sum of all the perturbations, i.e.,

$$\Delta\lambda_i = \sum_{m\in\mathcal{E}_p} \Delta\lambda_i(m) \quad \text{and} \quad \Delta\boldsymbol{u}_i = \sum_{m\in\mathcal{E}_p} \Delta\boldsymbol{u}_i(m). \tag{6}$$

In the case we perturb the graph by adding some edges, the perturbation writes again as in (6), but with opposite sign.

III. Online Learning of Signals over Dynamic Graphs

Let us consider a time-varying signal $\boldsymbol{x}[n] \in \mathbb{C}^N$ defined over the dynamic graph $\mathcal{G}[n] = (\mathcal{V}, \mathcal{E}[n])$, where n is the time index, $\mathcal{V}$ is the fixed set of vertices, and $\mathcal{E}[n]$ is the time-varying set of edges. At each time n, noisy samples of the signal are taken over a (randomly) time-varying subset of vertices, according to the following model:

$$\boldsymbol{y}[n] = \mathbf{D}_{\mathcal{S}[n]}\left(\boldsymbol{x}[n] + \boldsymbol{v}[n]\right), \tag{7}$$

where $\mathbf{D}_{\mathcal{S}[n]} = \mathrm{diag}\{d_i[n]\}_{i=1}^N \in \mathbb{R}^{N\times N}$ is a random sampling operator over the dynamic vertex set $\mathcal{S}[n]$, such that its diagonal elements are equal to 1 if $i \in \mathcal{S}[n]$, and 0 otherwise; and $\boldsymbol{v}[n] \in \mathbb{C}^N$ is zero-mean, spatially and temporally independent observation noise, with covariance matrix $\mathbf{C}_v = \mathrm{diag}\{\sigma_1^2, \ldots, \sigma_N^2\}$. The broad goal of this paper is to recover the time-varying graph signals $\boldsymbol{x}[n]$ from the noisy, partial, and streaming observation in (7), in an online fashion and with limited complexity per time slot n. To this aim, we build an online estimator that exploits the sequence of dynamic graphs $\mathcal{G}[n]$ to interpolate the time-varying signal from the collected samples. At each instant n, we assume to observe the graph variations, which evolve over time as:

$$\mathbf{L}[n] = \mathbf{L} + \Delta\mathbf{L}[n] = \mathbf{L} - \sum_{m\in\mathcal{E}_p[n]} \boldsymbol{a}_m\boldsymbol{a}_m^T, \tag{8}$$

such that the observed Laplacian at time n, i.e., $\mathbf{L}[n]$, can be expressed as the sum of the Laplacian $\mathbf{L}$, associated with a nominal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, plus a perturbation that involves the deletion of a (small) set $\mathcal{E}_p[n] \subseteq \mathcal{E}$ of edges.[1] Then, according to (8) and to the analysis in (3), (5), in the case of small perturbations, the eigenvector matrix $\mathbf{U}[n]$ of the instantaneous Laplacian $\mathbf{L}[n]$ can be approximated as:

$$\widetilde{\mathbf{U}}[n] = \mathbf{U} + \Delta\mathbf{U}[n] = \mathbf{U} + \sum_{m\in\mathcal{E}} z_m[n]\mathbf{U}\mathbf{B}_m, \tag{9}$$

where $z_m[n]$ is a Bernoulli random variable that is equal to 1 if $m \in \mathcal{E}_p[n]$, and 0 otherwise; $\mathbf{U} \in \mathbb{C}^{N\times N}$ collects the eigenvectors of $\mathbf{L}$, and $\mathbf{B}_m = \left\{b_{ij}^{(m)}\right\}_{i,j=1}^N \in \mathbb{C}^{N\times N}$ where

$$b_{ij}^{(m)} = \frac{[u_j(i_m) - u_j(f_m)][u_i(f_m) - u_i(i_m)]}{\lambda_i - \lambda_j}(1 - \delta_{ij}), \tag{10}$$

[1] Model (8) might be easily extended to incorporate a further perturbation term due to the addition of edges to the nominal graph. However, for simplicity, in this paper we consider only a random edge deletion model.

ISBN 978-90-827970-1-5 © EURASIP 2018

for all $i, j = 1, \ldots, N$, where δ_{ij} denotes the Kronecker delta. In particular, to recover the time-varying signal $\boldsymbol{x}[n]$ from the sampled observations in (7), from (1) and (9), we exploit a parsimonious representation of the signals on a basis composed by the sequence of approximated eigenvector matrices $\widetilde{\mathbf{U}}[n]$ in (9), but considering only a subset $\mathcal{F}$ of columns (frequency indexes), i.e.,

$$\widetilde{\mathbf{U}}_{\mathcal{F}}[n] = \mathbf{U}_{\mathcal{F}} + \sum_{m \in \mathcal{E}} z_m[n] \mathbf{U} \mathbf{B}_{\mathcal{F},m}, \tag{11}$$

where $\mathbf{B}_{\mathcal{F},m} \in \mathbb{C}^{N \times |\mathcal{F}|}$ represents the collection of columns of $\mathbf{B}_m$ associated with the graph frequency indexes $\mathcal{F}$. Thus, from (7) and (11), following an LMS approach [24], we seek for the optimal GFT vector say, e.g., $\boldsymbol{s}_{\mathcal{F}} \in \mathbb{C}^{|\mathcal{F}|}$, that solves the following optimization problem:

$$\min_{\boldsymbol{s}_{\mathcal{F}}} \; \mathbb{E} \left\| \mathbf{D}_{\mathcal{S}[n]} \left(\boldsymbol{y}[n] - \widetilde{\mathbf{U}}_{\mathcal{F}}[n] \boldsymbol{s}_{\mathcal{F}} \right) \right\|^2 \tag{12}$$

where $\mathbb{E}(\cdot)$ denotes the expectation operator. An LMS-type solution iteratively solves (12) by means of a stochastic steepest-descent procedure exploiting only instantaneous information $\{\boldsymbol{y}[n], \widetilde{\mathbf{U}}_{\mathcal{F}}[n]\}_n$. Thus, letting $\widehat{\boldsymbol{x}}[n]$ be the current estimate of signal $\boldsymbol{x}[n]$, the LMS method evolves as illustrated in Algorithm 1, where $\mu > 0$ is a fixed step-size, and we have exploited the fact that $\mathbf{D}_{\mathcal{S}[n]}$ is an idempotent operator.

Algorithm 1: LMS on Dynamic Graphs

```
Data: $(\boldsymbol{y}[n], \widetilde{\mathbf{U}}_{\mathcal{F}}[n])_{n \geq 0}$. Start with random $\widehat{\boldsymbol{s}}_{\mathcal{F}}[0]$. Given a (small) step-size $\mu > 0$, for each time $n \geq 0$, repeat:
    $\widehat{\boldsymbol{x}}[n] = \widetilde{\mathbf{U}}_{\mathcal{F}}[n] \widehat{\boldsymbol{s}}_{\mathcal{F}}[n]$
    $\widehat{\boldsymbol{s}}_{\mathcal{F}}[n+1] = \widehat{\boldsymbol{s}}_{\mathcal{F}}[n] + \mu \, \widetilde{\mathbf{U}}_{\mathcal{F}}[n]^H \mathbf{D}_{\mathcal{S}[n]} \left( \boldsymbol{y}[n] - \widetilde{\mathbf{U}}_{\mathcal{F}}[n] \widehat{\boldsymbol{s}}_{\mathcal{F}}[n] \right)$
```

At every iteration n, given the current values of $\widehat{\boldsymbol{s}}_{\mathcal{F}}[n]$ and $\widetilde{\mathbf{U}}_{\mathcal{F}}[n]$, the first step of Algorithm 1 provides an estimate $\widehat{\boldsymbol{x}}[n]$ of the graph signal. Then, the second step updates the estimate $\widehat{\boldsymbol{s}}_{\mathcal{F}}[n]$ of the GFT vector. An important feature of the proposed approach is that, thanks to the small perturbation assumption that has led to (11), the algorithm does not need to recompute the eigenvectors of the dynamic graph at each time, but can update $\widetilde{\mathbf{U}}_{\mathcal{F}}[n]$ via (11), thus reducing the complexity from $O\left(|\mathcal{F}|^3 |\mathcal{S}[n]|\right)$ to $O(|\mathcal{F}||\mathcal{S}[n]|)$. Furthermore, as we will see in the sequel, the mean-square analysis of the algorithm will be greatly simplified. Algorithm 1 extends the methods previously proposed in [17], [19] for adaptive GSP, by incorporating the temporal variations of the graph defining the signal support.

A. Mean-square Analysis

To carry out a mean-square analysis of Algorithm 1, we assume the following model for the graph signal variation:

$$\boldsymbol{x}[n] = \widetilde{\mathbf{U}}_{\mathcal{F}}[n] \boldsymbol{s}_{\mathcal{F}}^o = \mathbf{U}_{\mathcal{F}} \boldsymbol{s}_{\mathcal{F}}^o + \sum_{m \in \mathcal{E}} z_m[n] \mathbf{U} \mathbf{B}_{\mathcal{F},m} \boldsymbol{s}_{\mathcal{F}}^o, \tag{13}$$

for all $n \geq 0$, where $\boldsymbol{s}^o \in \mathbb{C}^{|\mathcal{F}| \times 1}$. Also, we introduce an independence assumption on the random sampling process and the random edge deletion model.

Assumption 1 (Sampling process): The stationary sampling process $\{d_i[l]\}$ is temporally and spatially independent, for all $i = 1, \ldots, N$ and $l \leq n$. ■

Assumption 2 (Edge deletion model): The stationary edge deletion process $\{z_m[l]\}$ is temporally and spatially independent, for all $m \in \mathcal{E}$ and $l \leq n$. ■

Let $\widetilde{\boldsymbol{s}}_{\mathcal{F}}[n] = \widehat{\boldsymbol{s}}_{\mathcal{F}}[n] - \boldsymbol{s}_{\mathcal{F}}^o$ be the error vectors on the GFT at time n. Thus, using (7) and (13) in Algorithm 1, we obtain:

$$\begin{aligned} \widetilde{\boldsymbol{s}}_{\mathcal{F}}[n+1] \; &= \; \left(\mathbf{I} - \mu \, \widetilde{\mathbf{U}}_{\mathcal{F}}[n]^H \mathbf{D}_{\mathcal{S}[n]} \widetilde{\mathbf{U}}_{\mathcal{F}}[n] \right) \widetilde{\boldsymbol{s}}_{\mathcal{F}}[n] \\ & \quad + \mu \, \widetilde{\mathbf{U}}_{\mathcal{F}}[n]^H \mathbf{D}_{\mathcal{S}[n]} \boldsymbol{v}[n]. \end{aligned} \tag{14}$$

Starting from (14), we derive a mean-square analysis of Algorithm 1, which relies also on the following assumption.

Assumption 3 (Small step-size): The step-size μ is chosen sufficiently small so that terms that depend on higher-order powers of μ can be ignored. ■

Let us now define some quantities that will be useful in the sequel. Let $\boldsymbol{p} = [p_1, \ldots, p_m]^T$ be the sampling probability vector, where $p_i = \mathbb{E}\{d_i[n]\}$ denotes the probability to sample node i at time n. Also, let $\zeta_m = \mathbb{E}\{z_m[n]\}$ be the probability that edge m is deleted at time n, and $r_{ml} = \mathbb{E}\{z_m[n] z_l[n]\} = \zeta_m \delta_{ml} + \zeta_m \zeta_l (1 - \delta_{ml})$. Finally, let us define the following matrix quantities:

$$\begin{aligned} \mathbf{H}(\boldsymbol{p}) \; &= \; \mathbb{E}\left\{ \widetilde{\mathbf{U}}_{\mathcal{F}}[n]^H \mathbf{D}_{\mathcal{S}[n]} \widetilde{\mathbf{U}}_{\mathcal{F}}[n] \right\} \; = \; \mathbf{U}_{\mathcal{F}}^H \mathrm{diag}(\boldsymbol{p}) \mathbf{U}_{\mathcal{F}} \\ & + \sum_{m \in \mathcal{E}} \zeta_m \mathbf{U}_{\mathcal{F}}^H \mathrm{diag}(\boldsymbol{p}) \mathbf{U} \mathbf{B}_{\mathcal{F},m} + \sum_{m \in \mathcal{E}} \zeta_m \mathbf{B}_{\mathcal{F},m}^H \mathbf{U}^H \mathrm{diag}(\boldsymbol{p}) \mathbf{U}_{\mathcal{F}} \\ & + \sum_{m \in \mathcal{E}} \sum_{l \in \mathcal{E}} r_{ml} \mathbf{B}_{\mathcal{F},m}^H \mathbf{U}^H \mathrm{diag}(\boldsymbol{p}) \mathbf{U} \mathbf{B}_{\mathcal{F},m} \end{aligned} \tag{15}$$

$$\begin{aligned} \mathbf{H}_2(\boldsymbol{p}) \; &= \; \mathbb{E}\left\{ \widetilde{\mathbf{U}}_{\mathcal{F}}[n]^H \mathbf{D}_{\mathcal{S}[n]} \boldsymbol{v}[n] \boldsymbol{v}[n]^H \mathbf{D}_{\mathcal{S}[n]} \widetilde{\mathbf{U}}_{\mathcal{F}}[n] \right\} \\ & = \; \mathbf{U}_{\mathcal{F}}^H \mathrm{diag}(\boldsymbol{p}) \mathbf{C}_v \mathbf{U}_{\mathcal{F}} + \sum_{m \in \mathcal{E}} \zeta_m \mathbf{B}_{\mathcal{F},m}^H \mathbf{U}^H \mathrm{diag}(\boldsymbol{p}) \mathbf{C}_v \mathbf{U}_{\mathcal{F}} \\ & + \sum_{m \in \mathcal{E}} \zeta_m \mathbf{U}_{\mathcal{F}}^H \mathrm{diag}(\boldsymbol{p}) \mathbf{C}_v \mathbf{U} \mathbf{B}_{\mathcal{F},m} \\ & + \sum_{m \in \mathcal{E}} \sum_{l \in \mathcal{E}} r_{ml} \mathbf{B}_{\mathcal{F},m}^H \mathbf{U}^H \mathrm{diag}(\boldsymbol{p}) \mathbf{C}_v \mathbf{U} \mathbf{B}_{\mathcal{F},m}. \end{aligned} \tag{16}$$

The main results are summarized in the following Theorem.

Theorem 1: Let Assumptions 1, 2, and 3 hold. Then, under the observation model (7), (13), for any initial condition, Algorithm 1 is stable in the mean-square error sense if the sampling probability vector $\boldsymbol{p}$ *and the step-size* μ *are chosen such that matrix* $\mathbf{H}(\boldsymbol{p})$ *in (15) is invertible, and*

$$0 < \mu < \frac{2 \lambda_{\min}\big(\mathbf{H}(\boldsymbol{p})\big)}{\lambda_{\max}^2\big(\mathbf{H}(\boldsymbol{p})\big)}. \tag{17}$$

Furthermore, the mean-square deviation (MSD) writes as

$$\begin{aligned} \mathrm{MSD}(\boldsymbol{p}) \; &= \; \lim_{n \to \infty} \; \mathbb{E}\|\widetilde{\boldsymbol{s}}_{\mathcal{F}}[n]\|^2 \\ &= \; \frac{\mu}{2} \, \mathrm{Tr}\left[\mathbf{H}(\boldsymbol{p})^{-1} \mathbf{H}_2(\boldsymbol{p}) \right] + O(\mu^2), \end{aligned} \tag{18}$$

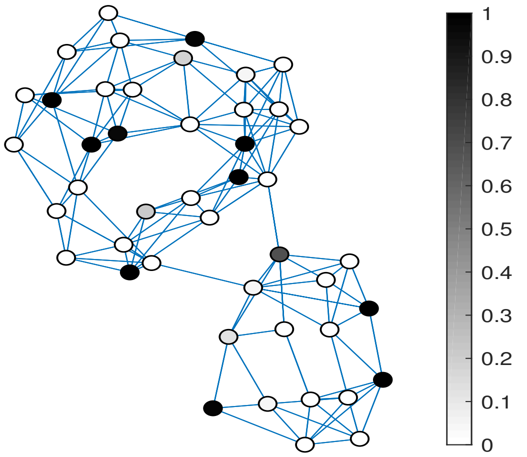

Fig. 1: Graph Topology and optimal sampling probabilities.

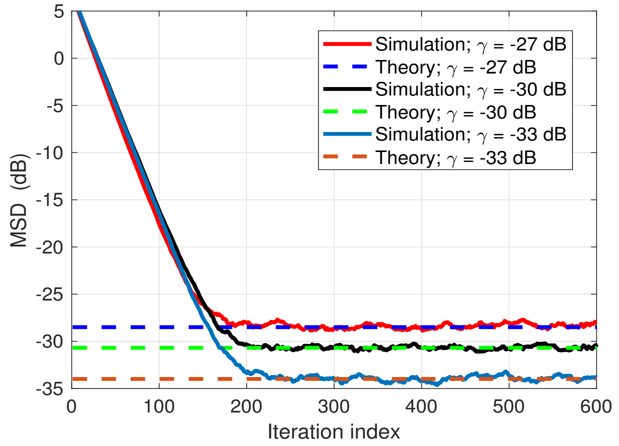

Fig. 2: Learning curve of Algorithm 1, where sampling probabilities are designed solving (20) for different values of γ.

and the convergence rate is well approximated by

$$\alpha(\boldsymbol{p}) = 1 - 2\mu\lambda_{\min}\big(\mathbf{H}(\boldsymbol{p})\big). \tag{19}$$

Proof. The proof follows similar arguments to those used in [19], and is omitted due to lack of space. ■

B. Sampling Strategies

The mean-square analysis in Sec. III-A illustrates how the convergence rate and the mean-square performance of Algorithm 1 are affected by the sampling probability vector $\boldsymbol{p}$ [cf. (18) and (19)]. Then, following a sparse sensing approach [25], [26], the goal of this Section is to propose sampling strategies aimed at designing the probability vector $\boldsymbol{p}$ that optimizes the tradeoff between graph sampling rate and learning performance of Algorithm 1. In the sequel, invoking Assumption 3, we neglect the term $O(\mu^2)$ in (18), and consider (19) as the convergence rate.

The proposed strategy aims at finding the sampling probability vector $\boldsymbol{p}$ that minimizes the sampling rate over the entire graph, while guaranteeing a target performance of Algorithm 1 in terms of MSD in (18) and convergence rate in (19). The optimization problem can be cast as:

$$\begin{aligned} &\min_{\mathbf{0}\le\boldsymbol{p}\le\mathbf{1}} \;\; \mathbf{1}^T\boldsymbol{p} \\ &\quad \text{s.t.} \quad \lambda_{\min}\big(\mathbf{H}(\boldsymbol{p})\big) \ge \frac{1-\bar{\alpha}}{\mu}, \\ &\qquad\quad \mathrm{Tr}\left(\mathbf{H}(\boldsymbol{p})^{-1}\mathbf{H}_2(\boldsymbol{p})\right) \le \frac{2\gamma}{\mu}. \end{aligned} \tag{20}$$

The first constraint imposes that the convergence rate of the algorithm is larger than a desired value, i.e., α in (19) is smaller than a target value, say, e.g., $\bar{\alpha} \in (0,1)$. The second constraint guarantees a target mean-square performance, i.e., the MSD in (18) must be less than or equal to a prescribed value, say, e.g., $\gamma > 0$. Problem (20) is non-convex due to the non-convex constraint on the MSD. To handle the non-convexity of (20), we proceed similarly to [19], i.e., we exploit an upper bound of the MSD in (18) given by:

$$\mathrm{MSD}(\boldsymbol{p}) \,\le\, \overline{\mathrm{MSD}}(\boldsymbol{p}) \triangleq \frac{\mu}{2}\,\frac{\mathrm{Tr}\big(\mathbf{H}_2(\boldsymbol{p})\big)}{\lambda_{\min}\big(\mathbf{H}(\boldsymbol{p})\big)}, \tag{21}$$

for all $\boldsymbol{p} \in \mathbb{R}^N$. Replacing function (18) with (21) in problem (20), not only the second constraint in (20) is always satisfied, but also the problem becomes convex and its global solution can be found using efficient numerical tools [27].

IV. Numerical Results

Let us consider a graph composed of 45 nodes, whose nominal topology is illustrated in Fig. 1, and such that each edge has a certain probability $\zeta_m = 0.2$, for all $m \in \mathcal{E}$, to be deleted at every time instant. The observation noise in (7) is zero-mean, Gaussian, with a diagonal covariance matrix such that each element is chosen uniformly random between 0 and 0.01. In Fig. 1, we also report the optimal sampling probability vector obtained solving problem (20) [with the upper bound in (21)], and considering $\bar{\alpha} = 0.95$, $|\mathcal{F}| = 8$, $\mu = 0.1$, and $\gamma = -33$ dB. As we can notice from Fig. 1, the method selects a sparse sampling probability vector in order to attain the requirements on the MSD and the learning rate. Then, to validate the theoretical results derived in Sec. III-A, in Fig. 2 we report the learning curve (in terms of MSD) of the LMS on Dynamic Graphs (cf. Algorithm 1), where the sampling strategy is selected solving problem (20) for different values of γ, while setting a target performance on the learning rate given by $\bar{\alpha} = 0.95$. The curves are averaged over 100 independent simulations. As we can see from Fig. 2, the numerical results match well with the theoretical findings.

Finally, we test the adaptation capability of the proposed method on a road network example. We consider an intersection among two roads in the center of Rome, Italy. We have placed $N = 18$ landmarks (nodes of the graph) over the streets in a regular fashion, and connected adjacent landmarks on the same lane and at the junctions. A traffic light alternates the traffic over the two roads, thus producing a periodical

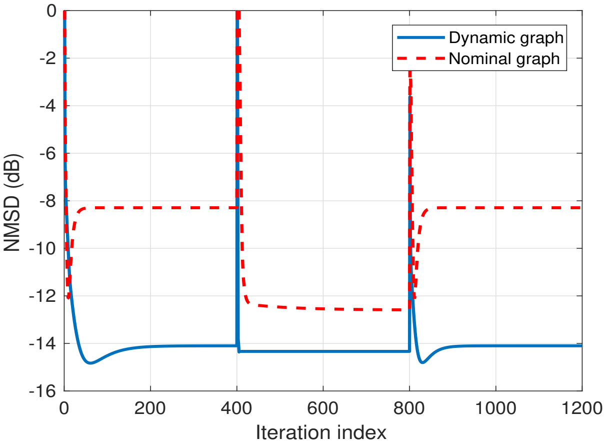

Fig. 3: Transient NMSD for different graph dynamics.

and known variation of the graph topology. The graph signal represents the number of of cars passing through the landmarks during a period of 20 seconds, and was obtained using a realistic simulator of urban mobility, namely, SUMO [28]. The goal is to infer the traffic situation from a small number of samples, exploiting the known variation of the graph. We consider $|F| = 4$, and we take a number of samples equal to $|S| = 8$. We consider a situation where the traffic light switches two times over the simulation period. Thus, in Fig. 3, we illustrate the behavior of the normalized MSD (NMSD), i.e., $\|\widehat{\boldsymbol{x}}[n] - \boldsymbol{x}^o[n]\|^2/\|\boldsymbol{x}^o[n]\|^2$, with $\boldsymbol{x}^o[n]$ denoting the true signal present at time n, considering the case where we exploit the known temporal variability of the graph, and the case where instead we assume a nominal graph fixed over time and agnostic of the traffic light. As expected, we can see from Fig. 3 how, exploiting the temporal variations of the graph, we obtain a large gain with respect to using a fixed graph.

V. Conclusions

In this paper, we have introduced a novel LMS strategy for learning time-varying signals over dynamic graphs. Assuming a small perturbation model, we have derived a mean-square analysis that assessed how the temporal variation of the graph and the sampling strategy affect the performance of the proposed method, thus paving the way to the formulation of an optimization criterion aimed at selecting the sampling probabilities as an optimal trade-off between graph sampling rate and learning performance of the algorithm. The resulting non-convex problem was then simplified in order to obtain a convex, but approximated, surrogate problem. Finally, some numerical tests were presented to verify the theoretical findings and assess the performance of the proposed approach.

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 Sig. Proc.*, vol. 61, pp. 1644–1656, 2013.

[3] 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.

[4] R. Lafferty and J. Kondor, "Diffusion kernels on graphs and other discrete structures," in *Machine Learning: Proceedings of the 19th International Conference*, 2002, pp. 315–322.

[5] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[6] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *The J. of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

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

[8] 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.

[9] 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.

[10] S. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *IEEE Conf. on Acous., Speech and Sig. Proc.*, May 2013, pp. 5445–5449.

[11] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 3864–3868.

[12] 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.

[13] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. on Sig. Proc.*, vol. 64, no. 14, pp. 3775–3789, 2016.

[14] L. F. Chamon and A. Ribeiro, "Greedy sampling of graph signals," *IEEE Transactions on Signal Processing*, vol. 66, no. 1, pp. 34–47, 2018.

[15] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Applied and Computational Harmonic Analysis*, 2016.

[16] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Signal recovery on graphs: Fundamental limits of sampling strategies," *IEEE Trans. on Sig. and Inf. Proc. over Net.*, vol. 2, no. 4, pp. 539–554, 2016.

[17] P. Di Lorenzo, S. Barbarossa, P. Banelli, and S. Sardellitti, "Adaptive least mean squares estimation of graph signals," *IEEE Trans. on Sig. and Inf. Proc. over Net.*, vol. 2, no. 4, pp. 555–568, 2016.

[18] P. Di Lorenzo, P. Banelli, S. Barbarossa, and S. Sardellitti, "Distributed adaptive learning of graph signals," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4193–4208, 2017.

[19] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, and G. Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *IEEE Trans. on Signal Processing*, vol. 20, no. 13, pp. 3584–3598, 2018.

[20] D. Romero, V. N. Ioannidis, and G. B. Giannakis, "Kernel-based reconstruction of space-time functions on dynamic graphs," *IEEE J. of Sel. Topics in Sig. Proc.*, vol. 11, no. 6, pp. 856–869, 2017.

[21] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Filtering random graph processes over random time-varying graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4406–4421, 2017.

[22] J. H. Wilkinson, *The algebraic eigenvalue problem.* Clarendon Press Oxford, 1965, vol. 87.

[23] E. Ceci and S. Barbarossa, "Small perturbation analysis of network topologies," *Proc. of Intl. Conf. on Acous. Speech and Signal Process.*, pp. 4194–4198, Calgary, Canada, Apr. 2018.

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

[25] 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.

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

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

[28] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility: an overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation.* ThinkMind, 2011.