PREDICTING THE EVOLUTION OF STATIONARY GRAPH SIGNALS

Andreas Loukas[†]*, Elvin Isufi*[‡] *and Nathanael Perraudin*[*]

[†] Swiss Federal Insitute of Technology in Lausanne (EPFL), Switzerland
[‡] Circuits and Systems Group, Delft University of Technology, The Netherlands
[*]Swiss Data Science Center, ETH Zurich and EPFL, Switzerland
e-mails: andreas.loukas@epfl.ch; e.isufi-1@tudelft.nl; nathanael.perraudin@sdsc.ethz.ch

ABSTRACT

One way of tackling the dimensionality issues arising in the modeling of a multivariate process is to assume that the inherent data structure can be captured by a graph. We here focus on the problem of predicting the evolution of a process that is time and graph stationary, i.e., a time-varying signal whose first two statistical moments are invariant over time and correlated to a known graph topology. This stationarity assumption allows us to regularize the estimation problem, reducing the variance and computational complexity, two common issues plaguing high-dimensional vector autoregressive models. In addition, our method compares favorably to state-of-the-art graph and time-based methods: it outperforms previous graph causal models as well as a purely time-based method.

***Index Terms*—** Signal processing on graphs, multivariate processes, prediction, joint stationarity, time-varying graph signals.

1. INTRODUCTION

In the problem of modeling and predicting statistical processes wide-sense stationarity is a helpful assumption that allows us to learn the statistics of a process using very few samples [1]. Especially for time-series prediction, learning from few samples is crucial as one needs to estimate future values after only partially observing a single realization of the statistical process. This is the main reason why classical models for estimation and prediction of univariate processes, such as Wiener filters and autoregressive moving average models (ARMA), rely on stationarity to produce predictions.

For high-dimensional multivariate processes, following the same methodology is often problematic as the number of parameters to be estimated increases quadratically with the number of variables, often rendering the problem intractable [2]. A common way to deal with this dimensionality issue is to utilize the inherent relationships between variables often represented by a graph [3]. Initially the graph assumption has been considered for time-invariant problems, to address tasks such as clustering [4, 5], low-rank extraction [3], spectral estimation [6–8] and semi-supervised learning [9, 10]. Very recently, following the generalization of harmonic analysis [11–13] and filtering [14–16] to time-varying graph signals, the graph assumption was also leveraged for modeling time-vertex processes [17–20].

The objective of this paper is to propose a multivariate model that exploits the graph structure so as to facilitate the task of prediction. From a statistical perspective, our model amounts to assuming stationarity not only with respect to the time-dimension, but also with respect of the graph topology [6, 7]. The concept of *joint* (time-vertex) stationarity, which was introduced in [21], was shown to facilitate regression even when the graph is only approximately known or the process is only close to stationary. Unlike previous work however, we here focus on prediction, where *causality* is important as one needs to forecast the future in a timely manner.

Concretely, we bring forth a decoupling theorem that makes it possible to decouple a joint (time-vertex) multivariate process into independent univariate processes. This allows us to (*i*) estimate the model parameters using traditional univariate techniques, and (*ii*) to further reduce the computational complexity by combining the training stage with (an optimal) low-rank approximation of our data. The learned models are causal and can be used to provide optimal predictions (in the mean-squared error (MSE) sense) at a cost that is equivalent to a constant number of matrix-vector multiplications. Our numerical results for two real datasets show that the proposed method outperforms the state-of-the-art graph causal models as well the univariate ARMA and multivariate vector autoregressive (VAR) approaches.

2. BACKGROUND

This section recalls some background information that will be used throughout the paper.

Graph signal processing. We consider a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W}_\mathcal{G})$, with $\mathcal{V}$ the set of N nodes (vertices), $\mathcal{E}$ the edge set, and $\boldsymbol{W}_\mathcal{G}$ the weighted adjacency matrix. The graph Fourier transform (GFT) of a vector $\boldsymbol{x} \in \mathbb{R}^N$ suported on $\mathcal{V}$ is defined as $\text{GFT}\{\boldsymbol{x}\} = \boldsymbol{U}_\mathcal{G}^\mathsf{H}\boldsymbol{x}$, where $\boldsymbol{U}_\mathcal{G}$ is the eigenvector matrix of the discrete[1] Laplacian matrix $\boldsymbol{L}_\mathcal{G} = \text{diag}(\boldsymbol{W}_\mathcal{G}\mathbf{1}_N) - \boldsymbol{W}_\mathcal{G} = \boldsymbol{U}_\mathcal{G}\boldsymbol{\Lambda}_\mathcal{G}\boldsymbol{U}_\mathcal{G}^\mathsf{H}$. Matrix $\boldsymbol{\Lambda}_\mathcal{G}$ is diagonal and contains the graph Laplacian eigenvalues (often referred as the graph frequencies [22]) in its main diagonal. The GFT allows us to extend filtering to graphs, where filtering $\boldsymbol{x}$ with the graph filter $h(\boldsymbol{L}_\mathcal{G})$ corresponds to element-wise multiplication in the spectral domain

$$h(\boldsymbol{L}_\mathcal{G})\boldsymbol{x} \triangleq \boldsymbol{U}_\mathcal{G} h(\boldsymbol{\Lambda}_\mathcal{G})\boldsymbol{U}_\mathcal{G}^\mathsf{H}\boldsymbol{x}.$$

The scalar function $h : \mathbb{R}_+ \mapsto \mathbb{R}$ indicates the graph filter frequency response and is applied to the diagonal elements of $\boldsymbol{\Lambda}_\mathcal{G}$.

Time-vertex signal processing. Let now $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T] \in \mathbb{R}^{N\times T}$ be the matrix that collects T successive temporal realizations of the graph process $\boldsymbol{x}_t$ evolving over $\mathcal{G}$. Let also from now on $\boldsymbol{x} = \text{vec}(\boldsymbol{X})$ be the vectorized form of $\boldsymbol{X}$. Then, from [13] the joint (time-vertex) Fourier transform (JFT) of $\boldsymbol{X}$ is

$$\text{JFT}\{\boldsymbol{X}\} = \boldsymbol{U}_\mathcal{G}^\top \boldsymbol{X}\boldsymbol{U}_T^*, \tag{1}$$

where $\boldsymbol{U}_\mathcal{G}$ is the graph Laplacian eigenvector matrix and $\boldsymbol{U}_T^*$ is the complex conjugate of the DFT matrix. Matrix $\boldsymbol{U}_T$ can also be interpreted as the eigenvector matrix of the symmetric time Laplacian matrix $\boldsymbol{L}_T = \boldsymbol{U}_T\,\boldsymbol{\Lambda}_T\,\boldsymbol{U}_T^\mathsf{H}$, where $\boldsymbol{\Lambda}_T(k,k) = 2(1-\cos(\omega_k))$ and the

[1]Though we use the combinatorial Laplacian in our presentation, our results are also applicable for alternative matrix representations of a graph, such as the graph shift, the random-walk, and the normalized Laplacian matrices.

angular frequency $\omega_k = 2\pi(k-1)/T$. In vectorized form, the JFT in (1) is JFT$\{\boldsymbol{x}\} = \boldsymbol{U}_J^{\mathsf{H}}\boldsymbol{x}$, with $\boldsymbol{U}_J = \boldsymbol{U}_T \otimes \boldsymbol{U}_G$ being a unitary matrix and $\otimes$ denoting the Kronecker operator.

Similar to the graph filters, now we talk about *joint* (time-vertex) filters $h(\boldsymbol{L}_G, \boldsymbol{L}_T)$ [15, 16]. The joint frequency response $h : \mathbb{R}_+ \times \mathbb{R} \mapsto \mathbb{R}$ is a function that operates on graph frequencies $\lambda_{\mathcal{G}}$ and angular frequencies ω. The output of a joint filter is

$$h(\boldsymbol{L}_G, \boldsymbol{L}_T)\boldsymbol{x} \triangleq \boldsymbol{U}_J\, h(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Omega})\, \boldsymbol{U}_J^{\mathsf{H}}\boldsymbol{x}, \tag{2}$$

with $h(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Omega})$ is a $NT \times NT$ diagonal matrix with $[h(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Omega})]_{k,k} = h(\lambda_n, \omega_\tau)$ and $k = N(\tau-1)+n$.

Jointly stationary processes. The first step in predicting the evolution of a process is to choose a good model for it. Motivated by the importance of stationarity for modeling statistical processes, our recent work generalized stationarity to time-vertex processes [18]. Specifically, a jointly wide-sense stationary (JWSS) process is defined as:

Definition 1 *(JWSS process). A process $\boldsymbol{x} = vec(\boldsymbol{X})$ is called Jointly Wide-Sense Stationary (JWSS), if and only if (i) the first moment of the process is constant $\mathbb{E}[\boldsymbol{x}] = c\mathbf{1}_{NT}$ and (ii) the covariance matrix of the process is a joint filter $\boldsymbol{\Sigma}_x = h(\boldsymbol{L}_G, \boldsymbol{L}_T)$, where $h(\cdot,\cdot)$ is a non-negative real function referred to as joint power spectral density (JPSD).*

This definition is a generalization of the classical notion of wide-sense stationarity, where now one assumes simultaneously wide-sense stationarity w.r.t. both the *time* and *vertex* domains. Indeed, assuming that a process is JWSS is equivalent to asserting that the process is (multivariate) time wide-sense stationary w.r.t. the time domain and (multivariate) vertex wide-sense stationary w.r.t. the graph domain (see Section 3.A in [18]).

In the sequel, we make use of this JWSS hypothesis and answer the question: "*How to construct models that concisely capture the characteristics of a JWSS process in order to facilitate short-term prediction?*". One way to exploit the JWSS of graph processes is by modeling it with non-causal models. Specifically, since the covariance of a JWSS signal $\boldsymbol{x}$ is diagonalizable by $\boldsymbol{U}_J$, without loss of generality, $\boldsymbol{x}$ can be modeled as

$$a(\boldsymbol{L}_G, \boldsymbol{L}_T)\, \boldsymbol{x} = b(\boldsymbol{L}_G, \boldsymbol{L}_T)\, \boldsymbol{\varepsilon}, \tag{3}$$

where the innovation vector $\boldsymbol{\varepsilon} = \text{vec}(\boldsymbol{E})$ is a random vector of some arbitrary distribution, with constant mean and identity covariance matrix $\boldsymbol{\Sigma}_\epsilon = \boldsymbol{I}$, implying that the above model is equivalent to the one considered in [18]. Matrices $a(\boldsymbol{L}_G, \boldsymbol{L}_T)$ and $b(\boldsymbol{L}_G, \boldsymbol{L}_T)$ are arbitrary joint filters (and not necessarily polynomials).

Despite its generality, model (3) does not suit our task of efficient prediction. First, the computational complexity of (3) scales with the number of nodes N and the graph process observations T, rendering the forecasting a computationally heavy task even for moderate graph dimensions. Second, (3) is not always causal. This is problematic for the task of prediction, where one needs to forecast the future in a timely manner. To this end, next we introduce a *graph causal model* which alleviates the computational costs and models the JWSS graph process in a causal fashion.

3. PREDICTING EVOLVING GRAPH SIGNALS

This section contains the theoretical contribution of this paper, where our objective is to forecast to the evolution of an observed JWSS process $\boldsymbol{x}$ with zero mean and JPSD $h(\omega, \lambda)$. We first introduce a causal model for $\boldsymbol{x}$ and then derive an optimal predictor for the future values of the process, in a mean-squared error (MSE) sense. The section is concluded by showing how the model parameters that approximate an arbitrary process (not necessarily causal) are estimated.

Graph causal models. As with classical VARMA models, for forecasting purposes it is more practical to assume that the output at time t can be expressed as a function of the input-output variables at the previous time steps, yielding the graph causal model[2]

$$\sum_{p=0}^{P} a_p(\boldsymbol{L}_{\mathcal{G}})\, \boldsymbol{x}_{t-p} = \sum_{q=0}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\, \boldsymbol{\varepsilon}_{t-q}, \tag{4}$$

where $a_p(\boldsymbol{L}_{\mathcal{G}})$ and $b_q(\boldsymbol{L}_{\mathcal{G}})$ represent the kernel matrices for some model orders P and Q and $\boldsymbol{\varepsilon}_t$ is the tth column of $\boldsymbol{E}$. The canonical form of (4) is directly obtained by setting $a_0(\boldsymbol{L}_{\mathcal{G}}) = b_0(\boldsymbol{L}_{\mathcal{G}}) = \boldsymbol{I}$. Obviously, every causal model can be written in the form (4) for $P \to \infty$ and $Q \to \infty$. Note also that (4) reduces the computational complexity to that of sparse matrix vector multiplication. We will refer to (4) as a graph causal VARMA (GC-VARMA) model.

Prediction. Suppose that $\boldsymbol{x}$ is the output of a graph causal model, where the input $\boldsymbol{\varepsilon}$ has zero mean and identity covariance. This implies that $\boldsymbol{x}$ abides to

$$\boldsymbol{x}_t = \sum_{q=0}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\, \boldsymbol{\varepsilon}_{t-q} - \sum_{p=1}^{P} a_p(\boldsymbol{L}_{\mathcal{G}})\, \boldsymbol{x}_{t-p}. \tag{5}$$

We consider that we have observed the vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}$ and that we want to estimate $\boldsymbol{x}_t$ from these values. Note that from (5) the knowledge of $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}$ implies also the knowledge of the realizations $\{\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_{t-1}\}$. Let us then denote with $\boldsymbol{x}_{t|t-1}$ the random vector realization of $\boldsymbol{x}_t$ conditioned to the vectors observed until $t-1$. We predict $\boldsymbol{x}_t$, thus obtaining the *one-step predictior* $\tilde{\boldsymbol{x}}_t$, as the conditional expectation of $\boldsymbol{x}_t$ given the realizations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}$, i.e.,

$$\tilde{\boldsymbol{x}}_t = \mathbb{E}[\boldsymbol{x}_{t|t-1}] = \sum_{q=0}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\mathbb{E}[\boldsymbol{\varepsilon}_{t-q|t-1}] - \sum_{p=1}^{P} a_p(\boldsymbol{L}_{\mathcal{G}})\, \boldsymbol{x}_{t-p}. \tag{6}$$

Since in (6) $\boldsymbol{\varepsilon}_{t-q|t-1}$ is the known realization of $\boldsymbol{\varepsilon}_{t-q}$, we can substitute $\boldsymbol{\varepsilon}_{t-q|t-1} = \boldsymbol{x}_{t-q|t-1} - \tilde{\boldsymbol{x}}_{t-q}$ for $q = 1, \ldots, Q$ and write $\tilde{\boldsymbol{x}}_t$ as

$$\begin{aligned}\tilde{\boldsymbol{x}}_t &= \mathbb{E}[\boldsymbol{\varepsilon}_t] + \sum_{q=1}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\, \left(\boldsymbol{x}_{t-q|t-1} - \tilde{\boldsymbol{x}}_{t-q}\right) - \sum_{p=1}^{P} a_p(\boldsymbol{L}_{\mathcal{G}})\boldsymbol{x}_{t-p} \\ &= \sum_{q=1}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\, (\boldsymbol{x}_{t-q} - \tilde{\boldsymbol{x}}_{t-q}) - \sum_{p=1}^{P} a_p(\boldsymbol{L}_{\mathcal{G}})\, \boldsymbol{x}_{t-p}, \end{aligned} \tag{7}$$

where in the above expression we exploited the fact that $\mathbb{E}[\boldsymbol{\varepsilon}_t] = \mathbf{0}_N$. In the following, we will show that this corresponds to the optimal choice, as it yields the minimum MSE. The k-step predictor can be obtained by repeating the above computation k times.

MSE analysis. Similar to the purely temporal case [23], the one-step prediction error $\boldsymbol{e}_t = \boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t$ depends only on the unknown innovations $\boldsymbol{\varepsilon}_t$ and the achieved MSE is the smallest possible. To see this, we need to show that $\boldsymbol{e}_t = \boldsymbol{\varepsilon}_t$, or equivalently that $\boldsymbol{d}_t = \boldsymbol{e}_t - \boldsymbol{\varepsilon}_t = \mathbf{0}$. By

[2] Throughout this work, our definition of causality is identical to that used in multivariate processes [2], which is different from the restricted causality used in the graph vector autoregressive model [17]. To avoid confusion, when referring to [17] we will use the notation of graph restricted causality VAR (GRC-VAR) model.

plugging (7) into the definition of $\boldsymbol{d}_t$, we get

$$\boldsymbol{d}_t = \boldsymbol{e}_t - \boldsymbol{\varepsilon}_t = \boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t - \boldsymbol{\varepsilon}_t = \sum_{q=1}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\left(\boldsymbol{\varepsilon}_{t-q} - \boldsymbol{e}_{t-q}\right) = -\sum_{q=1}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\boldsymbol{d}_{t-q}. \quad (8)$$

Therefore, for the considered canonical model $\sum_{q=0}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\boldsymbol{d}_{t-q} = \boldsymbol{0}_N$ for every t, which, under the assumption that the noise model is invertible[3], implies $\boldsymbol{d}_t = \boldsymbol{0}_N$. Directly, we find that $\boldsymbol{e}_t = \boldsymbol{\varepsilon}_t$ and the one-step MSE is equal to

$$\frac{\mathbb{E}[\|\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t\|_2^2]}{N} = \frac{\mathbb{E}[\boldsymbol{\varepsilon}_t^{\mathsf{H}}\boldsymbol{\varepsilon}_t]}{N} = \frac{\mathrm{tr}\left(\mathbb{E}[\boldsymbol{\varepsilon}_t\boldsymbol{\varepsilon}_t^{\mathsf{H}}]\right)}{N} = \frac{\mathrm{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}_t}\right)}{N}. \quad (9)$$

Since $\boldsymbol{\varepsilon}_t$ is unknown at time t, the above corresponds to the smallest achievable MSE.

Model estimation. We now want to identify the parameters of the graph causal model, i.e., the functions a_p and b_q which best match the observed process $\boldsymbol{X} \in \mathbb{R}^{N\times T}$. The canonical way to achieve this would be to minimize the prediction error residual by solving the following (non-linear) system of $N \times T$ equations involving $(P+Q)\,N$ unknowns[4]

$$\min_{a_p, b_q} \|\boldsymbol{x}_{t+1} - \tilde{\boldsymbol{x}}_{t+1}(a_p, b_q)\|_2^2, \quad (10)$$

where by $\tilde{\boldsymbol{x}}_{t+1}(a_p, b_q)$ we refer to the causal model based on $a_1(\boldsymbol{L}_{\mathcal{G}})$, $\ldots, a_P(\boldsymbol{L}_{\mathcal{G}})$ and $b_1(\boldsymbol{L}_{\mathcal{G}}), \ldots, b_Q(\boldsymbol{L}_{\mathcal{G}})$. In the sequel, we restrict the design to kernels that are linear, shift-invariant graph filters, i.e., kernels of the form $a_p(\boldsymbol{L}_{\mathcal{G}}) = \boldsymbol{U}_{\mathcal{G}} a_p(\boldsymbol{\Lambda}_{\mathcal{G}})\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$ and $b_q(\boldsymbol{L}_{\mathcal{G}}) = \boldsymbol{U}_{\mathcal{G}} b_q(\boldsymbol{\Lambda}_{\mathcal{G}})\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$ which decouple and simplify the problem into a N independent and well-studied problems with smaller complexity.

Proposition 1 *(Decoupling). Given the graph causal model* (4)

$$\sum_{p=0}^{P} a_p(\boldsymbol{L}_{\mathcal{G}})\boldsymbol{x}_{t-p} = \sum_{q=0}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\boldsymbol{\varepsilon}_{t-q}, \quad (11)$$

and denote as $\hat{\varepsilon}_t(n) = (\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}\boldsymbol{\varepsilon}_t)(n)$ and $\hat{x}_t(n) = (\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}\boldsymbol{x}_t)(n)$ the n-th entry of the GFT of $\boldsymbol{\varepsilon}_t$ and $\boldsymbol{x}_t$, respectively. Then, the input-output relation of $\hat{\varepsilon}_t(n)$ and $\hat{x}_t(n)$ for every n is given by an ARMA(P,Q) model $\sum_{p=0}^{P} a_p(n)\hat{x}_{t-p}(n) = \sum_{q=0}^{Q} b_q(n)\,\hat{\varepsilon}_{t-q}(n)$ with scalars $a_p(n) = [a_p(\boldsymbol{\Lambda}_{\mathcal{G}})]_{n,n}$ and $b_q(n) = [b_q(\boldsymbol{\Lambda}_{\mathcal{G}})]_{n,n}$.

Proof (Sketch): The claim can be proven by substituting $a_p(\boldsymbol{L}_{\mathcal{G}}) = \boldsymbol{U}_{\mathcal{G}} a_p(\boldsymbol{\Lambda}_{\mathcal{G}})\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$ and $b_q(\boldsymbol{L}_{\mathcal{G}}) = \boldsymbol{U}_{\mathcal{G}} b_q(\boldsymbol{\Lambda}_{\mathcal{G}})\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$ in (11) and multiply both sides by $\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$. ∎

Proposition 1 suggests that the graph causal model can be decomposed into N univariate ARMA processes, one for each graph frequency. These univariate processes are always uncorelated and they become independent when the innovation $\boldsymbol{\varepsilon}_t$ follows a Gaussian distribution. The latter has two important consequences: (i) in the graph frequency domain the model estimation of the graph causal model is split in N independent problems involving T equations and $P+Q$ unknowns; (ii) despite being non-linear, the model estimation for each of the N problems corresponds in fitting an temporal ARMA to a time-series. We can therefore use a number of well studied methods to solve for it, such as the subspace Gauss-Newton approach of [24]. The exact coefficients of the graph causal model are then found by an inverse GFT.

We remark that in our analysis the eigendecomposition of the graph Laplacian $\boldsymbol{L}_{\mathcal{G}}$ is necessary to estimate the model parameters. Therefore, the proposed framework suits better cases for small to medium sized graphs, where the eigenvalue decomposition cost (inherent in joint models) is overshadowed by that of the model estimation.

Low-rank models. The computational overhead of model estimation can be reduced by considering that graph processes are often (approximately) sparse in the graph frequency domain and thus consider only a subset of time-series. To limit large increments in prediction error, we can perform the selection process after first rotating the data. Concretely, let $\mathcal{S}$ denote the index set of size $K = |\mathcal{S}|$ and let $\boldsymbol{U}$ be a unitary (rotation) matrix. The $\{\boldsymbol{U}, \mathcal{S}\}$ low-rank approximation of $\boldsymbol{X}$ is

$$\tilde{\boldsymbol{X}}_{\boldsymbol{U},\mathcal{S}} = \boldsymbol{U}\boldsymbol{I}_{\mathcal{S}}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{X}, \quad (12)$$

where $\boldsymbol{I}_{\mathcal{S}}$ is a diagonal indicator matrix such that $[\boldsymbol{I}_{\mathcal{S}}]_{i,i} = 1$ if $i \in \mathcal{S}$ and zero otherwise.

The following theorem asserts that, if we are interested in the expected behavior, the *best low-rank approximation* of a JWSS process uses the graph eigenvectors $\boldsymbol{U}_{\mathcal{G}}$ to rotate the data (corresponding to the GFT). The letter is beneficial to us since it jointly exploits the sparsity in the graph Fourier domain and decouples the time-series, which coincides also with the first step in the model estimation. Therefore, by modelling only the time-series specified by the set $\mathcal{S}$ we attain a complexity reduction by a factor of N/K in the model estimation.

Theorem 1 *Let $\boldsymbol{X}$ be a zero-mean JWSS process. The best rank-K approximation of $\boldsymbol{X}$ is given by*

$$\{\boldsymbol{U}_{\mathcal{G}}, \boldsymbol{S}^{\star}\} = \arg\min_{\boldsymbol{U},\mathcal{S}} \mathbb{E}\left[\left\|\boldsymbol{X} - \tilde{\boldsymbol{X}}_{\boldsymbol{U},\mathcal{S}}\right\|_F^2\right] \quad s.t. \quad |\mathcal{S}| = K,$$

where $\boldsymbol{S}^{\star}$ contains the indices of the top K-diagonal elements of $\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}\mathbb{E}\left[\boldsymbol{X}\boldsymbol{X}^{\mathsf{H}}\right]\boldsymbol{U}_{\mathcal{G}}$.

Proof Let us define $\boldsymbol{A} = \boldsymbol{U}(\boldsymbol{I} - \boldsymbol{I}_{\mathcal{S}})\boldsymbol{U}^{\mathsf{H}}$. Then, following the Eckart–Young–Mirsky theorem [25, 26], the expected approximation error becomes

$$\mathbb{E}\left[\left\|\boldsymbol{X} - \tilde{\boldsymbol{X}}_{\boldsymbol{U},\mathcal{S}}\right\|_F^2\right] = \mathbb{E}\left[\|\boldsymbol{A}\boldsymbol{X}\|_F^2\right] = \mathrm{tr}\left(\boldsymbol{A}\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^{\mathsf{H}}]\boldsymbol{A}^{\mathsf{H}}\right). \quad (13)$$

Then, from Theorem 1 of [18], for each time instant $\boldsymbol{x}_t$, the graph signal is stationary with covariance $\boldsymbol{\Sigma}_t = \mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathsf{H}}] = \boldsymbol{U}_{\mathcal{G}} g_t^2(\boldsymbol{\Lambda}_{\mathcal{G}})\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$ implying that

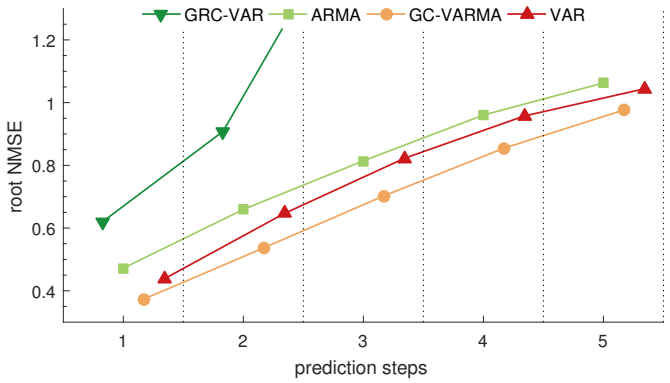
$$\boldsymbol{\Sigma}_{\mathcal{G}} \quad = \quad \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^{\mathsf{H}}] = \sum_{t=1}^{T}\mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{*}] = \sum_{t=1}^{T}\boldsymbol{\Sigma}_t = \boldsymbol{U}_{\mathcal{G}} g^2(\boldsymbol{\Lambda}_{\mathcal{G}})\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}},$$

where $g^2(\boldsymbol{\Lambda}_{\mathcal{G}}) = \sum_{t=1}^{T} g_t^2(\boldsymbol{\Lambda}_{\mathcal{G}})$ reordered such that $g(\lambda_1) \geq g(\lambda_2) \geq \ldots g(\lambda_N)$. Then, by substituting the expression of $\boldsymbol{\Sigma}_{\mathcal{G}}$ into (13) we obtain

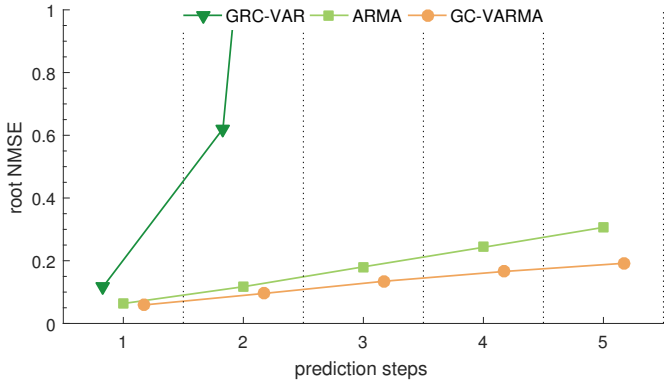
$$\mathbb{E}\left[\left\|\boldsymbol{X} - \tilde{\boldsymbol{X}}_{\boldsymbol{U},\mathcal{S}}\right\|_F^2\right] = \mathrm{tr}\left(\boldsymbol{A}\boldsymbol{\Sigma}_{\mathcal{G}}\boldsymbol{A}^{\mathsf{H}}\right) = \left\|\boldsymbol{\Sigma}_{\mathcal{G}}^{1/2} - \boldsymbol{U}\boldsymbol{I}_{\mathcal{S}}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{\Sigma}_{\mathcal{G}}^{1/2}\right\|_F^2 = \left\|g(\boldsymbol{\Lambda}_{\mathcal{G}}) - \boldsymbol{U}\boldsymbol{I}_{\mathcal{S}}\boldsymbol{U}^{\mathsf{H}}g(\boldsymbol{\Lambda}_{\mathcal{G}})\right\|_F^2. \quad (14)$$

[3] System $\sum_{q=0}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\boldsymbol{d}_{t-q} = \boldsymbol{0}_N$ has exactly one solution when matrix $b_0(\boldsymbol{L}_{\mathcal{G}}) \oplus b_1(\boldsymbol{L}_{\mathcal{G}}) \oplus \ldots \oplus b_Q(\boldsymbol{L}_{\mathcal{G}})$ is invertible, or equivalently when $b_q(\boldsymbol{L}_{\mathcal{G}})$ is invertible for each q.

[4] A matrix function, i.e., a function that preserve the space spanned by the matrix, of an $N \times N$ matrix has N degrees of freedom.



(a) Molene dataset



(b) dog dynamic mesh

Fig. 1: Prediction error in two datasets. A small horizontal offset is inserted to improve visibility.

By setting $\boldsymbol{B} = \boldsymbol{U}_{\mathcal{G}}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{I}_{\mathcal{S}}\boldsymbol{U}^{\mathrm{H}}\boldsymbol{U}_{\mathcal{G}}$, (14) becomes

$$\mathbb{E}\left[\left\|\boldsymbol{X}-\tilde{\boldsymbol{X}}_{\boldsymbol{U},\mathcal{S}}\right\|_F^2\right] = \sum_{i=1}^{N}|g(\lambda_i)-\boldsymbol{B}_{i,i}g(\lambda_i)|^2+\sum_{i\neq j}|\boldsymbol{B}_{i,j}g(\lambda_i)|^2$$
$$\geq \sum_{i=1}^{N}|g(\lambda_i)|^2|1-\boldsymbol{B}_{i,i}|^2 \geq \sum_{i=K+1}^{N}|g(\lambda_i)|^2 = \mathbb{E}\left[\left\|\boldsymbol{X}-\tilde{\boldsymbol{X}}_{\boldsymbol{U},\mathcal{S}^\star}\right\|_F^2\right], \quad (15)$$

where in the third step we use the fact that $\boldsymbol{B}_{i,i} \leq 1$ and is exactly 1 at most K times. The last expression shows that the global minimum is achieved for $\tilde{\boldsymbol{X}}_{\boldsymbol{U},\mathcal{S}^\star}$ with $\mathcal{S}$ containing the K largest components of $g(\lambda)$. ■

4. NUMERICAL RESULTS

We evaluate our method with two real data sets, namely the Molene weather data set and with a dynamic mesh representing a dog walking. The Molene data set contains hourly temperature measurements of $N = 32$ weather stations in the region of Brest (France) for $T = 744$ hours. The used graph is a geometric graph constructed from the node coordinates, with an average degree of 12. The to-be-predicted signal on graph are the temperature measurements in degrees Celsius. The dog mesh consists of $N = 250$ points (nodes) over $T = 59$ timesteps. We build a 10-nearest neighbor graph based on the distances between the coordinates over the first timestep. The

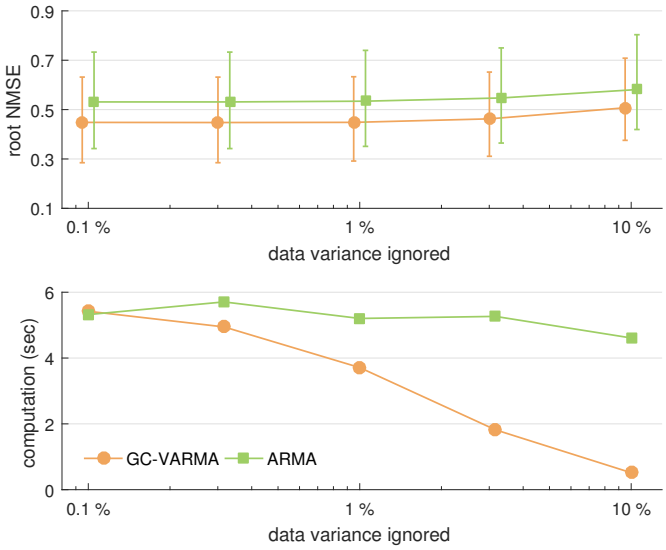


Fig. 2: Prediction error (top) and computational time (bottom) as a function of the ignored time-series in the low-rank prediction of the Molene dataset. At the expense of a small decrease in accuracy –here measured by the percentage of the data variance ignored– the estimation of graph causal models becomes very scalable.

to-be-predicted signal corresponds to the x-axis coordinate of each node over time. For both datasets, we used the first half of the data (along the time dimension) for the model estimation, and for each $t = T/2+1,\ldots,T$ we computed the k-step prediction root NMSE error $\|\tilde{\boldsymbol{x}}_{t+k|t}-\boldsymbol{x}_{t+k}\|_2/\|\boldsymbol{x}_{t+k}\|_2$.

We compare the performance of the proposed GC-VARMA model (4) with the (i) independent ARMA models, which model and predict each time-series independently (using N ARMA series), (ii) with the GRC-VAR model from [17], and (iii) with a standard vector autoregressive VAR model [2]. Since we here focus on constructing a process model given a graph, we do not attempt to identify the graph from the data as in [17]. We expect however that, when combined with graph identification, the prediction accuracy could be improved. Though the above models were not overly sensitive to parameterization in our experiments, the reported results we illustrate the prediction only for the best orders identified via exhaustive search[5].

Fig. 1 (a) shows the prediction error up to 5 future steps for all methods in the Molene weather dataset. We note that the proposed GC-VARMA model achieves the best performance, though the prediction error is reasonable only up to two future steps. On the other hand, the GRC-VAR approach is suitable only for one step in the future. In Fig. 1 (b) on the other hand, we show the prediction error for the dog dynamic mesh. We first note that the classic multivariate VAR model was not included in the results as it ran into numerical instabilities; this is common issue with VAR and VARMA models when the length of the timeseries is not much larger than the number of timeseries. This issue is also present for the GRC-VAR, which achieves a reasonable performance up to 2 prediction steps, but produces poor long-term predictions. The proposed GC-VARMA model achieves the best performance and significantly outperforms the other multivariate

[5]For Molene, we selected: GC-VARMA $P = 3$ and $Q = 2$, GRC-VAR $P = 10$, ARMA $P = 4$ and $Q = 4$, VAR $P = 3$. For the dog dynamic mesh, we set: GC-VARMA $P = 3$ and $Q = 2$, GRC-VAR $P = 10$, ARMA $P = 3$ and $Q = 2$.

approaches. However, contrarily to the Molene weather dataset, the disjoint ARMA method gives also a reasonable performance close to the GC-VARMA. Further, as the walking dog scenario is more regular than the temperature variations we can predict reasonably well up to five future steps.

Lastly, Fig. 2 examines the effect of low-rank prediction on the 2-steps prediction error and model estimation time on the Molene dataset. In this experiment, we tested low-rank predictors which ignored a given percentage of the data variance (as estimated by the training data). Moreover, we considered the full dataset of $T = 31 \times 24$ hours. As supported by our theoretical results, performing the approximation in the graph spectral domain (GC-VARMA) far outperforms a naive approximation in the native graph domain, where one chooses to model only the timeseries with the most variance, yielding a significant computational benefit at the expense of only a small decrease in prediction accuracy.

5. CONCLUSIONS

This work proposed GC-VARMA, a model that relies on the assumption of joint (time-vertex) wide-sense stationarity and can be used to optimally forecast the future outcomes of processes evolving on graphs. The joint stationarity assumption allowed us deal with the high-dimensionality of the data, while also reducing the computational complexity of model estimation to that of fitting univariate ARMA models to a set of time-series. In our experiments, GC-VARMA was found to outperform a vector autoregressive (VAR) model, a state-of-the-art graph causal model, as well as a purely time-based model. However, we remark that the proposed model requires the eigendecomposition of the graph Laplacian, which renders it suitable for graphs consisting of up to a few thousand nodes.

6. REFERENCES

[1] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[2] H. Lütkepohl, *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[3] N. Shahid, N. Perraudin, V. Kalofolias, G. Puy, and P. Vandergheynst, "Fast robust pca on graphs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 740–756, 2016.

[4] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering." in *NIPS*, vol. 14, 2001, pp. 585–591.

[5] U. Von Luxburg, M. Belkin, and O. Bousquet, "Consistency of spectral clustering," *The Annals of Statistics*, pp. 555–586, 2008.

[6] B. Girault, "Stationary graph signals using an isometric graph translation," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 1516–1520.

[7] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3462–3477, 2017.

[8] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *arXiv preprint arXiv:1603.04667*, 2016.

[9] M. Belkin and P. Niyogi, "Semi-supervised learning on riemannian manifolds," *Machine learning*, vol. 56, no. 1-3, pp. 209–239, 2004.

[10] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning theory and kernel machines*. Springer, 2003, pp. 144–158.

[11] A. Sandryhaila and J. M. Moura, "Big Data Analysis with Signal Processing on Graphs: Representation and Processing of Massive Data Sets with Irregular Structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, 2014.

[12] A. Loukas and D. Foucard, "Frequency Analysis of Temporal Graph Signals," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. Washington D.C, USA, 2016.

[13] F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud, "A time-vertex signal processing framework," *IEEE Transactions on Signal Processing*, 2017.

[14] 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, 2017.

[15] ——, "Separable Autoregressive Moving Average Graph-Temporal Filters," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 200–204.

[16] E. Isufi, G. Leus, and P. Banelli, "2-dimensional finite impulse response graph-temporal filters," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. Washington DC, USA: IEEE, December 2016.

[17] J. Mei and J. M. Moura, "Signal processing on graphs: Modeling (causal) relations in big data," *arXiv preprint*, 2015.

[18] A. Loukas and N. Perraudin, "Stationary time-vertex signal processing," *arXiv preprint arXiv:1611.00255*, 2016.

[19] V. N. Ioannidis, D. Romero, and G. B. Giannakis, "Inference of spatiotemporal processes over graphs via kernel kriged kalman filtering," in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 1679–1683.

[20] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, and G. Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *arXiv preprint arXiv:1709.03726*, 2017.

[21] N. Perraudin, A. Loukas, F. Grassi, and P. Vandergheynst, "Towards stationary time-vertex signal processing," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3914–3918.

[22] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[23] L. Ljung, "System identification," in *Signal Analysis and Prediction*. Springer, 1998, pp. 163–173.

[24] A. Wills and B. Ninness, "On gradient-based search for multivariable system estimates," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 298–306, 2008.

[25] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[26] I. Markovsky, "Structured low-rank approximation and its applications," *Automatica*, vol. 44, no. 4, pp. 891–909, 2008.