Kernel-based Reconstruction of Dynamic Functions over Dynamic Graphs

---

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Vasileios Ioannidis

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Professor Georgios B. Giannakis, Advisor

August 2017

Acknowledgments

First, my sincerest thanks and deepest gratitude goes to Prof. Georgios B. Giannakis for giving me the opportunity to be a part of the prestigious SPiNCOM research group and introducing me to the world of academic research. His guidance and discussions have been of such great value, making the completion of this thesis a reality. In addition, he has and continues to aid me in developing clear scientific thought and expression.

I would also like to deeply thank Prof. Daniel Romero, whose contributions were crucial to the completion of this thesis. In a special way I would like to extend my appreciation to Prof. Jarvis Haupt and Prof. Zhi-Li Zhang who agreed to serve on my thesis committee. Thanks go to other professors in the departments of Electrical Engineering and Computer Science whose graduate level courses helped me build the necessary background to embark on this area of research.

The content of this thesis has benefited from numerous discussions with current and former members of the SPiNCOM group: Dr. A. Nikolakopoulos, Dr. B. Baingana, D. Lee, F. Sheikholeslami, D. K. Berberidis, P. A. Traganitis, G. Wang, L. Zhang, T. Chen, Y. Shen, G. V. Karanikolas, A. Sadeghi, and M. Ma, as well as outside of the group Prof. N. Sidiropoulos.

Last but not the least, I would like to express my deepest thanks to my family: my parents, Nikos and Evi, for raising me and my dear sister Eleni, and for their continued love and support throughout my life.

Vassilis N. Ioannidis, Minneapolis, July 12, 2017

Work in this thesis was supported by ARO grant W911NF-15-1-0492 and NSF grants 1343248, 1442686, and 1514056

Abstract

Graph-based methods pervade the inference toolkits of numerous disciplines including sociology, biology, neuroscience, physics, chemistry, and engineering. A challenging problem encountered in this context pertains to determining the attributes of a set of vertices given those of another subset at possibly different time instants. Leveraging spatiotemporal dynamics and prior information can drastically reduce the number of observed vertices, and hence the cost of sampling. Alleviating the limited flexibility of existing approaches, this thesis broadens the kernel-based graph function estimation framework to reconstruct time-evolving functions over possibly time-evolving topologies. This encompassing approach inherits the versatility and generality of kernel-based methods, for which no knowledge on distributions or second-order statistics is required. Efficient inference algorithms are derived that operate in an online and even data-adaptive fashion. Moreover, semi-parametric approaches capable of incorporating the structure of known graph functions without sacrificing the flexibility of the overall model are advocated. Numerical tests with real data sets corroborate the merits of the proposed methods relative to competing alternatives.

Contents

List of Figures

Chapter 1

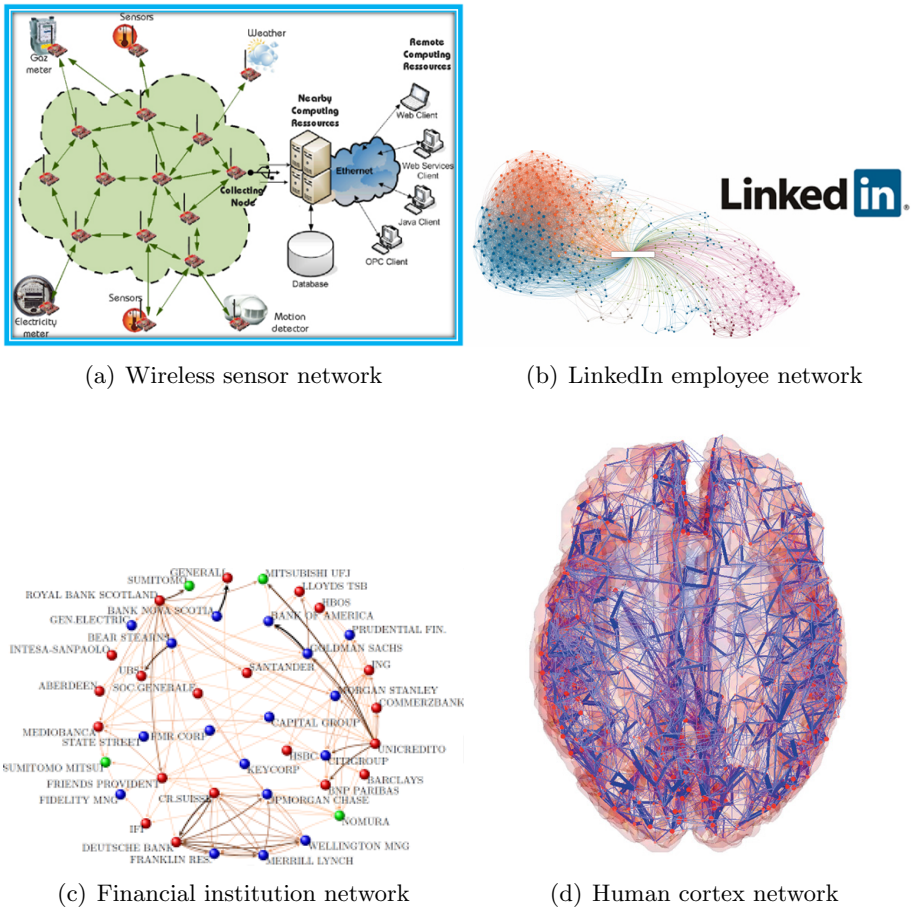
Introduction

A number of applications involve data that can be efficiently represented as node attributes over social, economic, sensor, communication, and biological networks [1, 2]. An inference problem that often emerges is to predict the attributes of all nodes in the network given the attributes of a subset of nodes. In the finance network of Fig 1.1(c) for instance, where nodes correspond to stocks and edges indicate dependence between them, one may be interested in predicting the price of all stocks in the network knowing the price of some.

Especially in real large-scale networks, often one can afford working only with limited node observations due to inherent restrictions particular to the inference task at hand. In social networks, for example, individuals may be reluctant to share personal information; in sensor networks the nodes may report observations sporadically in order to save energy, see Fig. 1.1(a); in brain networks acquiring node samples may involve invasive procedures (e.g. electrocorticography), as that depicted in Fig. 1.1(d).

Existing approaches typically formulate this problem as the reconstruction of a function or signal on a graph [1,3–7], and rely on its smoothness with respect to the graph, in the sense that neighboring vertices have similar function values. This principle suggests, for instance, estimating one person's salary by looking at their friends' salary, a task encountered in employment-oriented social networks such as LinkedIn; see also Fig. 1.1(b).

A more challenging problem involves reconstructing time-evolving functions on graphs, such as the ones describing the time-dependent activity of regions in a brain network, given



(a) Wireless sensor network (b) LinkedIn employee network

(c) Financial institution network (d) Human cortex network

Figure 1.1: Networks that model data originating from different applications

their values on a subset of vertices and time instants. Efficiently exploiting spatiotemporal dynamics can markedly impact sampling costs by reducing the number of vertices that need to be observed to attain a target performance. Such a reduction is of paramount interest in applications such as invasive electrocorticography (ECoG), where observing a vertex requires the implantation of an intracranial electrode [8].

1.1 Context and motivation

An extensive body of literature has dealt with reconstructing time-invariant graph functions. Parametric approaches – falling under the umbrella term of signal processing on graphs [5,6] – either adopt a *graph-bandlimited* model, which postulates that the signal of interest lies

in a B-dimensional subspace related to the graph topology [9–14], or they assume that the signal can be sparsely represented by an overcomplete dictionary [15]. Nonparametric techniques rely on *graph kernels* [3, 4, 7, 16–18] which allow them to also accommodate nonbandlimited signals, upon selecting an appropriate kernel [19]. The performance of algorithms for parametric models is limited by how well the signals actually adhere to the selected model. Nonparametric models on the other hand, offer flexibility and robustness but cannot readily incorporate information available a priori – a fact that could limit their performance especially in face of scarce node samples.

Although one could reconstruct a time-varying function by separately applying these time-agnostic schemes per time instant, leveraging time dynamics typically affords estimators with improved performance. Schemes tailored for time-evolving functions on graphs include [20] and [21], which predict the function values at time t given observations up to time $t-1$. However, these schemes assume that the function of interest adheres to a specific vector autoregression and all vertices are observed at previous time instances. Moreover, [20] requires Gaussianity along with a rather *ad hoc* form of stationarity.

Other works target time-invariant functions, but can afford tracking sufficiently slow variations. This is the case of the dictionary learning approach in [22] and the distributed algorithms in [23] and [24]. Unfortunately, the flexibility of these algorithms to capture spatial information is also limited since [22] focuses on Laplacian regularization, whereas [23] and [24] require the signal to be graph-bandlimited.

Different approaches investigate special instances of the reconstruction problem with domain-specific requirements and assumptions [25, 26]. Finally, it is worth mentioning that no graph-based reconstruction approach has dealt with time-evolving topologies.

1.2 Thesis contributions

This thesis builds upon the kernel-based learning framework from machine learning to develop estimators for reconstruction of time-invariant as well as time-varying graph functions. The inference algorithms exploit smoothness captured by the *graph kernels* [1, 3, 4], that relate to the topology of the graph.

Specifically, to address limitations of existing estimators of time-invariant graph functions this thesis advocates a *semi-parametric* approach whereby the function of interest is modeled as the superposition of a parametric and a nonparametric component. While the former leverages side information, the latter accounts for deviations from the parametric part, and can also promote smoothnes using *graph kernels*.

Next, to account for time-varying settings, the existing kernel-based learning framework is naturally extended to incorporate time-evolving functions over possibly dynamic graphs through the notion of *graph extension*, by which the time dimension receives the same treatment as the spatial dimension. The versatility of kernel-based methods to leverage spatial information [19] is thereby inherited and broadened to account for temporal dynamics as well. Incidentally, this vantage point also accommodates time-varying sampling sets and topologies. Moreover, systematic guidelines are provided to construct two families of *space-time kernels* with complementary strengths. The first facilitates judicious control of regularization on a space-time frequency plane, whereas the second can afford time-varying topologies. Batch and online estimators are also put forth, and a novel kernel Kalman filter (KKF) is developed to obtain these estimates at a affordable computational cost.

Finally, to accommodate cases where the wanted function exhibits markedly different behaviors over space and time a kernel kriged Kalman filter (KKrKF) is introduced. The novel deterministic estimator is derived from a KRR criterion, and is capable of promoting smoothness over time and space through judicious use of *graph kernels*. Choosing the appropriate kernel is an application-dependent art, and affects significantly the performance of the inference algorithms [19]. Data-driven techniques for selecting the pertinent kernel are known as multi-kernel learning (MKL) algorithms [27]. To cope with the challenging problem of MKL this thesis develops a novel data-driven approach, that dynamically explores a pool of multiple kernels. The time-varying setting of the problem calls for an online multi-kernel learning approach that adapts to the observed data on-the-fly. The complexity of the proposed algorithm is linear in the number of time samples, rendering it attractive for online data applications.

Results from this thesis are reported in journal and conference publications [28–31].

1.3 Thesis outline

The remainder of this thesis is organized as follows.

- Chapter 2 reviews the graph kernels, and the KRR framework [4, 19].
- Chapter 3 introduces the novel semi-parametric estimator for time-invariant graph functions, along with extensive numerical tests.
- Chapter 4 generalizes the KRR framework of space and time, develops algorithms for reconstruction of time-varying functions and showcases their superior performance over existing methods in real data applications.
- Chapter 5 introduces the KKrKF estimator as well as the online MKL approach and reports numerical experiments that demonstrate the benefits of the proposed method.
- Chapter 6 presents a concluding summary of the kernel-based reconstruction approaches, as well as a brief discussion on future research directions.

1.4 Notational conventions

Scalars are denoted by lowercase letters, vectors by bold lowercase, and matrices by bold uppercase, while $(\mathbf{A})_{m,n}$ is the (m,n)-th entry of matrix $\mathbf{A}$. Superscripts ^{T} and † respectively denote transpose and pseudo-inverse. If $\mathbf{A} := [\mathbf{a}_1, \ldots, \mathbf{a}_N]$, then $\text{vec}\{\mathbf{A}\} := [\mathbf{a}_1^T, \ldots, \mathbf{a}_N^T]^T := \mathbf{a}$ and $\text{vec}^{-1}\{\mathbf{a}\} := \mathbf{A}$. With $N \times N$ matrices $\{\mathbf{A}_t\}_{t=1}^T$ and $\{\mathbf{B}_t\}_{t=2}^T$ with $\mathbf{A}_t = \mathbf{A}_t^T \ \forall t$, $\text{btridiag}\{\mathbf{A}_1, \ldots, \mathbf{A}_T; \mathbf{B}_2, \ldots, \mathbf{B}_T\}$ represents the symmetric block tridiagonal

matrix

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_2^T & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_2 & \mathbf{A}_2 & \mathbf{B}_3^T & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_3 & \mathbf{A}_3 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_{T-1} & \mathbf{B}_T^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}_T & \mathbf{A}_T \end{bmatrix}.$$

Similarly, bdiag $\{\mathbf{A}_1, \dots, \mathbf{A}_N\} :=$ btridiag$\{\mathbf{A}_1, \dots, \mathbf{A}_N; \mathbf{0}, \dots, \mathbf{0}\}$ is a block diagonal matrix. Symbols $\odot$, $\otimes$, and $\oplus$ respectively denote element-wise (Hadamard) matrix product, Kronecker product, and Kronecker sum, the latter being defined for $\mathbf{A} \in \mathbb{R}^{M \times M}$ and $\mathbf{B} \in \mathbb{R}^{N \times N}$ as $\mathbf{A} \oplus \mathbf{B} := \mathbf{A} \otimes \mathbf{I}_N + \mathbf{I}_M \otimes \mathbf{B}$. The n-th column of the identity matrix $\mathbf{I}_N$ is represented by $\boldsymbol{i}_{N,n}$. If $\mathbf{A}$ is a matrix and $\mathbf{x}$ a vector, then $||\mathbf{x}||_{\mathbf{A}}^2 := \mathbf{x}^T \mathbf{A}^{-1} \mathbf{x}$ and $||\mathbf{x}||_2 := ||\mathbf{x}||_{\mathbf{I}}$. $\mathbb{S}_+^N$ represents the cone of $N \times N$ positive definite matrices. Finally, $\delta[\cdot]$ stands for the Kronecker delta, and E for expectation.

Chapter 2

Graph Kernels

The present chapter reviews the existing framework for kernel-based reconstruction of time-invariant functions, that will be used as a building block for the estimators developed in the subsequent chapters.

Consider an undirected graph $\mathcal{G} := (\mathcal{V}, \mathbf{A})$, where $\mathcal{V} := \{v_1, \ldots, v_N\}$ is the vertex set, and $\mathbf{A}$ is the symmetric entry-wise nonegative $N \times N$ adjacency matrix, whose (n, n')-th entry denotes the edge weight between vertices v_n and $v_{n'}$. A real-valued signal on a graph is a function $f : \mathcal{V} \to \mathbb{R}$ that can be compactly represented by the $N \times 1$ vector $\mathbf{f} := [f(v_1), \ldots, f(v_N)]^T$. At each sampled node v_{n_s}, a measurement $y_s = f(v_{n_s}) + e_s$, $s = 1, \ldots, S$ is collected, where $\{e_s\}_{s=1}^S$ represents noise, and $1 \leq n_1 < \ldots < n_S \leq N$ are the indices of the observed vertices. Upon defining $\mathbf{e} := [e_1, \ldots, e_S]^T$, and $\mathbf{y} := [y_1, \ldots, y_S]^T$ it follows that $\mathbf{y} = \mathbf{S}\mathbf{f} + \mathbf{e}$, where $\mathbf{S}$ is an $S \times N$ sampling matrix with all zeros except for the entries (s, n_s), $s = 1, \ldots, S$, which contain ones. The nonparametric approach that will be presented aims at reconstructing a graph function $\mathbf{f}$, that takes values over the vertices of $\mathcal{G}$ given $\mathbf{A} \in \mathbb{R}_+^{N \times N}$, $\mathbf{S} \in \{0, 1\}^{S \times N}$, and $\mathbf{y}$.

2.1 Background on kernel-based reconstruction

At first, one may feel tempted to seek a least-squares estimate

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} ||\mathbf{y} - \mathbf{S}\mathbf{f}||_2^2, \tag{2.1}$$

but noting that the N unknowns in $\mathbf{f}$ cannot be generally identified from the $S \leq N$ samples in $\mathbf{y}$ dismisses such an approach. This underdeterminacy prompts estimates of the form

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} ||\mathbf{y} - \mathbf{S}\mathbf{f}||_2^2 + \mu\rho(\mathbf{f}), \tag{2.2}$$

where $\mu > 0$ and the regularizer $\rho(\mathbf{f})$ promotes a certain structure in $\mathbf{f}$. A customary $\rho(\mathbf{f})$ encourages smooth estimates by penalizing functions that exhibit pronounced variations among neighboring vertices, for instance by means of the so-called Laplacian regularizer

$$\rho_{\text{LR}}(\mathbf{f}) := \frac{1}{2}\sum_{n=1}^{N}\sum_{n'=1}^{N} A_{n,n'}(f_n - f_{n'})^2 \tag{2.3}$$

which heavily penalizes differences between function values at vertices connected by strong links (large $A_{n,n'}$). Expression (2.3) formalizes the notion of smoothness introduced in Sec. 1, according to which a function is smooth if it takes similar values at neighboring vertices. Since $\rho_{\text{LR}}(\mathbf{f})$ is small if $\mathbf{f}$ is smooth, and large otherwise, $\rho_{\text{LR}}(\mathbf{f})$ acts as a proxy quantifying smoothness of $\mathbf{f}$, in the sense that given two functions $\mathbf{f}$ and $\mathbf{f}'$, the former is said to be smoother than the latter iff $\rho_{\text{LR}}(\mathbf{f}) < \rho_{\text{LR}}(\mathbf{f}')$ and vice versa. More general proxies are reviewed next.

Upon defining the $N \times N$ *Laplacian* matrix $\mathbf{L} := \text{diag}\{\mathbf{A1}\} - \mathbf{A}$, the functional in (2.3) can be rewritten after some algebra as $\rho_{\text{LR}}(\mathbf{f}) = \mathbf{f}^T\mathbf{L}\mathbf{f}$; see e.g. [1, Ch. 2]. It readily follows from (2.3) that $\rho_{\text{LR}}(\mathbf{f}) \geq 0\ \forall \mathbf{f}$, which in turn implies that $\mathbf{L}$ is positive semidefinite. Therefore, $\mathbf{L}$ admits an eigenvalue decomposition $\mathbf{L} = \mathbf{U}\,\text{diag}\{\boldsymbol{\lambda}\}\,\mathbf{U}^T$, where the eigenvectors in $\mathbf{U} := [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ and the eigenvalues in $\boldsymbol{\lambda} := [\lambda_1, \ldots, \lambda_N]$ are sorted so that

$0 = \lambda_1 \leq \ldots \leq \lambda_N$. By letting $\check{f}_n := (\mathbf{u}_n)^T\mathbf{f}$, one finds that

$$\rho_{\mathrm{LR}}(\mathbf{f}) = \sum_{n=1}^{N} \lambda_n |\check{f}_n|^2 \tag{2.4}$$

which means that $\rho_{\mathrm{LR}}(\mathbf{f})$ is the weighted superposition of the magnitude of the projections of $\mathbf{f}$ onto the eigenvectors of $\mathbf{L}$ with weights given by the corresponding eigenvalues. As described next, (2.4) provides an insightful interpretation of $\rho_{\mathrm{LR}}(\mathbf{f})$ in a transformed domain. Specifically, a number of works advocate the term *graph Fourier transform* or frequency representation of f to refer to $\{\check{f}_n\}_{n=1}^N$; see e.g. [5]. The main argument resides in that $\{\mathbf{u}_n\}_{n=1}^N$ play a role analogous to complex exponentials in signal processing for time signals, in the sense that (i) complex exponentials are eigensignals of the continuous counterpart of the Laplacian operator $\mathbf{f} \mapsto \mathbf{L}\mathbf{f}$, and (ii) $\{\mathbf{u}_n\}_{n=1}^N$ are eigensignals of the so-called linear, shift-invariant filters [6], which are the graph counterparts of linear, time-invariant filters in signal processing for time signals. Thus, $\mathbf{f} = \sum_{n=1}^N \check{f}_n \mathbf{u}_n$ resembles in some sense the synthesis equation of the Fourier transform, and one can therefore interpret $\{\mathbf{u}_n\}_{n=1}^N$ as a Fourier basis. Because $\lambda_1 \leq \ldots \leq \lambda_N$, it follows from $\rho_{\mathrm{LR}}(\mathbf{u}_n) = (\mathbf{u}_n)^T\mathbf{L}\mathbf{u}_n = \lambda_n$ that $\rho_{\mathrm{LR}}(\mathbf{u}_1) \leq \ldots \leq \rho_{\mathrm{LR}}(\mathbf{u}_N)$. Hence, sorting the eigenvectors $\{\mathbf{u}_n\}_{n=1}^N$ in increasing order of their associated eigenvalue is tantamount to sorting them in decreasing order of smoothness. Similarly, the complex exponentials in the traditional Fourier basis are indexed by their frequency, which can be thought of as an (inverse) proxy of time-domain smoothness. Comparing both scenarios suggests interpreting λ_n, or the index n, as the *graph frequency* of $\mathbf{u}_n$.

Back to (2.4), it is seen that $\rho_{\mathrm{LR}}(\mathbf{f})$ penalizes high-frequency components more heavily than low-frequency ones, thus promoting estimates with a "low-pass" graph Fourier transform. A finer control of how energy is distributed across frequency can be attained upon applying a transformation $r : \mathbb{R} \to \mathbb{R}_+$ to λ_n, giving rise to regularizers of the form

$$\rho_{\mathrm{LK}}(\mathbf{f}) = \sum_{n=1}^{N} r(\lambda_n)|\check{f}_n|^2 = \mathbf{f}^T\mathbf{K}^\dagger\mathbf{f} \tag{2.5a}$$

Kernel name	**Function**	**Parameters**
Diffusion [3]	$r(\lambda) = \exp\{\sigma^2\lambda/2\}$	σ^2
p-step random walk [4]	$r(\lambda) = (a-\lambda)^{-p}$	$a \geq 2$, $p \geq 0$
Regularized Laplacian [4, 32]	$r(\lambda) = 1 + \sigma^2\lambda$	σ^2
Bandlimited [19]	$r(\lambda) = \begin{cases} 1/\beta & \lambda \leq \lambda_{\max} \\ \beta & \text{otherwise} \end{cases}$	$\beta > 0$, $\lambda_{\max}$

Table 2.1: Common spectral weight functions.

where

$$\mathbf{K}^{\dagger} := r(\mathbf{L}) := \mathbf{U}^T \operatorname{diag}\{r(\boldsymbol{\lambda})\}\mathbf{U} \tag{2.5b}$$

is referred to as *Laplacian kernel* [4]. Table 2.1 summarizes some well-known examples arising with specific choices of r.

Further broadening the scope of the generalized Laplacian kernel regularizers in (2.5), the so-called *kernel ridge regression* (KRR) estimators are given by

$$\hat{\mathbf{f}} := \arg\min_{\mathbf{f}} \frac{1}{S}||\mathbf{y} - \mathbf{S}\mathbf{f}||_2^2 + \mu \mathbf{f}^T \mathbf{K}^{\dagger}\mathbf{f} \tag{2.6}$$

for an arbitrary positive semidefinite matrix $\mathbf{K}$, not necessarily a Laplacian kernel. The user-selected parameter $\mu > 0$ balances the importance of the regularizer relative to the fitting term $S^{-1}||\mathbf{y} - \mathbf{S}\mathbf{f}||_2^2$. KRR estimators have well-documented merits and solid grounds on statistical learning theory; see e.g. [33]. Different regularizers and fitting functions lead to even more general algorithms; see e.g. [19].

Chapter 3

Semiparametric Reconstruction of Graph Functions

Signal reconstruction over graphs arises naturally in diverse science and engineering applications. Existing methods employ either parametric or nonparametric approaches based on graph kernels. Although the former are adequate when the signals of interest adhere to postulated models, their performance degrades rapidly under model mismatch. Nonparametric alternatives on the other hand are flexible, but not as parsimonious in capturing prior information. To address the aforementioned limitations, this chapter develops a *semi-parametric* approach whereby the signal of interest is modeled as the superposition of a parametric and a nonparametric component. While the former leverages side information, the latter accounts for deviations from the parametric part, and can also promote smoothness using *graph kernels*.

3.1 Preliminaries

Consider an undirected graph $\mathcal{G} := (\mathcal{V}, \mathbf{A})$, where $\mathcal{V} := \{v_1, \ldots, v_N\}$ is the vertex set, and $\mathbf{A}$ is the symmetric entry-wise nonegative $N \times N$ adjacency matrix, whose (n, n')-th entry denotes the edge weight between vertices v_n and $v_{n'}$. We assume that $\mathcal{G}$ has no self-loops, meaning $(\mathbf{A})_{n,n} = 0,\ \forall v_n \in \mathcal{V}$. The Laplacian matrix of $\mathcal{G}$ is $\mathbf{L} := \mathbf{D} - \mathbf{A}$,

with $(\mathbf{D})_{n,n} := \sum_{m=1}^{N}(\mathbf{A})_{n,m}$ and $(\mathbf{D})_{n,n'} := 0$ if $n \neq n'$; matrix $\mathbf{L}$ is known to be positive semidefinite [4].

A real-valued signal on a graph is a function $f : \mathcal{V} \to \mathbb{R}$ that can be compactly represented by the $N \times 1$ vector $\mathbf{f} := [f(v_1), \ldots, f(v_N)]^T$. At each sampled node v_{n_s}, a measurement $y_s = f(v_{n_s}) + e_s$, $s = 1, \ldots, S$ is collected, where $\{e_s\}_{s=1}^{S}$ represents noise, and $1 \leq n_1 < \ldots < n_S \leq N$ are the indices of the observed vertices. Upon defining $\mathbf{e} := [e_1, \ldots, e_S]^T$, and $\mathbf{y} := [y_1, \ldots, y_S]^T$ it follows that

$$\mathbf{y} = \mathbf{S}\mathbf{f} + \mathbf{e} \tag{3.1}$$

where $\mathbf{S}$ is an $S \times N$ sampling matrix with all zeros except for the entries (s, n_s), $s = 1, \ldots, S$, which contain ones.

Function f is modeled as the superposition $f = f_{\mathrm{P}} + f_{\mathrm{NP}}$, or, in vector form

$$\mathbf{f} = \mathbf{f}_{\mathrm{P}} + \mathbf{f}_{\mathrm{NP}} \tag{3.2}$$

where $\mathbf{f}_{\mathrm{P}} := [f_{\mathrm{P}}(v_1), \ldots, f_{\mathrm{P}}(v_N)]^T$, and $\mathbf{f}_{\mathrm{NP}} := [f_{\mathrm{NP}}(v_1), \ldots, f_{\mathrm{NP}}(v_N)]^T$. The parametric $f_{\mathrm{P}}(v) := \sum_{m=1}^{M} \beta_m b_m(v)$ captures the known signal structure via the basis $\mathcal{B} := \{b_m\}_{m=1}^{M}$, while the nonparametric term f_{NP} belongs to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, which accounts for deviations from the span of $\mathcal{B}$. The goal of this chapter is efficient and reliable estimation of $\mathbf{f}$ given $\mathbf{y}$, $\mathbf{S}$, $\mathcal{B}$, $\mathcal{H}$ and $\mathbf{A}$.

Remark 1. *Decomposing f as in (3.2) is well motivated in certain applications. Consider for instance an employment-oriented social network like LinkedIn, and let the goal be to estimate the salaries of all users given information about the salaries of a few. Clearly, besides network connections, exploiting available information regarding the users' education level and work experience could benefit the reconstruction task. Another application where this decomposition fits nicely, is in recommender systems. Inferring preference scores for every item, given the users' feedback about particular items, could be cast as a signal reconstruction problem over the item correlation graph. Exploiting side information about the items, is known to alleviate limitations of pure collaborative filtering techniques, leading to consid-*

erably improved recommendation performance [34, 35]. In our setup, the item attributes can be used to create a parametric base capturing the user's coarse level preferences.

3.2 Semi-parametric Reconstruction

This section introduces our semi-parametric approach. Specifically, Sec. 3.2.1 reviews the RKHS for graph functions and Sec. 3.2.2 presents two semi-parametric estimators.

3.2.1 Reproducing kernel Hilbert spaces on graphs

An RKHS is a space of functions $h : \mathcal{V} \to \mathbb{R}$ expressed in terms of a kernel function $\kappa : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ as

$$\mathcal{H} := \left\{ h : h(v) = \sum_{n=1}^{N} \alpha_n \kappa(v, v_n),\ \alpha_n \in \mathbb{R} \right\} \tag{3.3}$$

where $\kappa(v_n, v_{n'})$ captures the similarity between vertices v_n and $v_{n'}$ [4]. Upon defining the $N \times N$ positive definite matrix with entries $(\mathbf{K})_{n,n'} := \kappa(v_n, v_{n'})$, and $\mathbf{h} := [h(v_1), h(v_2), \ldots, h(v_N)]$, we can write

$$\mathbf{h} = \mathbf{K}\boldsymbol{\alpha} \tag{3.4}$$

where $\boldsymbol{\alpha} := [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$. The RKHS norm of a function h is given by $\|h\|^2_{\mathcal{H}} := \sum_{n=1}^{N} \sum_{n'=1}^{N} \alpha_n \alpha_{n'} \kappa(v_n, v_{n'})$ or in vector form by

$$\|h\|^2_{\mathcal{H}} = \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \tag{3.5}$$

and is usually employed as a regularization term to control overfitting [4, 19, 36].

Laplacian kernels have been widely used [3, 4, 19, 28, 31] to promote *smoothness* with respect to the underlying graph topology, by penalizing functions that exhibit pronounced variations among neighboring vertices (cf. Sec. 2.1). For a given Laplacian matrix with

eigendecomposition $\mathbf{L} = \mathbf{U}\,\mathrm{diag}\{\boldsymbol{\lambda}\}\mathbf{U}^T$, a family of graph kernels is defined as [4]

$$\mathbf{K} := r^{-1}(\mathbf{L}) := \mathbf{U}\,\mathrm{diag}\{r^{-1}(\boldsymbol{\lambda})\}\mathbf{U}^T \tag{3.6}$$

where $r : \mathbb{R} \to \mathbb{R}_+$ is chosen to be a monotonically increasing function. Table 2.1 summarizes common choices of $r(\cdot)$ which can be selected to promote a certain structure in the so-called graph Fourier transform of $\mathbf{h}$ [4, 5, 19].

3.2.2 Kernel-based semi-parametric reconstruction

Since $f_{\mathrm{NP}} \in \mathcal{H}$, vector $\mathbf{f}_{\mathrm{NP}}$ can be represented as in (3.4). By defining $\boldsymbol{\beta} := [\beta_1, \ldots, \beta_M]^T$, and the $N \times M$ matrix $\mathbf{B}$ with entries $(\mathbf{B})_{n,m} := b_m(v_n)$, the parametric term can be written in vector form as $\mathbf{f}_{\mathrm{P}} := \mathbf{B}\boldsymbol{\beta}$. The semi-parametric estimates can be found as the solution of the following optimization problem

$$\begin{aligned} \{\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}\} = \arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & \frac{1}{S}\sum_{s=1}^{S} \mathcal{L}(y_s, f(v_{n_s})) + \mu \|f_{\mathrm{NP}}\|_{\mathcal{H}}^2 \\ \text{s.t.} \quad & \mathbf{f} = \mathbf{f}_{\mathrm{P}} + \mathbf{f}_{\mathrm{NP}} \\ & \mathbf{f}_{\mathrm{P}} = \mathbf{B}\boldsymbol{\beta} \\ & \mathbf{f}_{\mathrm{NP}} = \mathbf{K}\boldsymbol{\alpha} \end{aligned} \tag{3.7}$$

where the fitting loss $\mathcal{L}$ quantifies the deviation of f from the data, and $\mu > 0$ is the regularization scalar that controls overfitting the nonparametric term. Using (3.7), we can express our semi-parametric estimate as $\hat{\mathbf{f}} = \mathbf{B}\hat{\boldsymbol{\beta}} + \mathbf{K}\hat{\boldsymbol{\alpha}}$.

Solving (3.7) entails minimization over $N + M$ variables. Clearly, when dealing with large-scale graphs this could lead to prohibitively large computational cost. To ensure applicability in big-data scenaria we leverage the dimensionality reduction effected through the semi-parametric version of the representer theorem [33, 36], which establishes that

$$\hat{\mathbf{f}} = \mathbf{B}\hat{\boldsymbol{\beta}} + \mathbf{K}\mathbf{S}^T\hat{\tilde{\boldsymbol{\alpha}}} \tag{3.8}$$

where $\hat{\bar{\boldsymbol{\alpha}}} := [\hat{\bar{\alpha}}_1, \ldots, \hat{\bar{\alpha}}_S]^T$. Estimates $\hat{\bar{\boldsymbol{\alpha}}}, \hat{\boldsymbol{\beta}}$ are found solving the optimization problem

$$\begin{aligned} \{\hat{\bar{\boldsymbol{\alpha}}}, \hat{\boldsymbol{\beta}}\} = \arg\min_{\bar{\boldsymbol{\alpha}}, \boldsymbol{\beta}} \quad & \frac{1}{S}\sum_{s=1}^{S} \mathcal{L}(y_s, f(v_{n_s})) + \mu \|f_{\mathrm{NP}}\|_{\mathcal{H}}^2 \\ \text{s.t.} \quad & \mathbf{f} = \mathbf{f}_{\mathrm{P}} + \mathbf{f}_{\mathrm{NP}} \\ & \mathbf{f}_{\mathrm{P}} = \mathbf{B}\boldsymbol{\beta} \\ & \mathbf{f}_{\mathrm{NP}} = \mathbf{K}\mathbf{S}^T\bar{\boldsymbol{\alpha}} \end{aligned} \tag{3.9}$$

where $\bar{\boldsymbol{\alpha}} := [\bar{\alpha}_1, \ldots, \bar{\alpha}_S]^T$. The RKHS norm in (3.9) is expressed as $\|f_{\mathrm{NP}}\|_{\mathcal{H}}^2 = \bar{\boldsymbol{\alpha}}^T\bar{\mathbf{K}}\bar{\boldsymbol{\alpha}}$, with $\bar{\mathbf{K}} := \mathbf{S}\mathbf{K}\mathbf{S}^T$. Relative to (3.7) the number of optimization variables in (3.9) is reduced to the more affordable $S + M$, with $S \ll N$.

We will consider two loss functions with complementary benefits: the *square* loss and the ϵ-*insensitive* loss. The square loss function is

$$\mathcal{L}(y_s, f(v_{n_s})) := \|y_s - f(v_{n_s}))\|_2^2 \tag{3.10}$$

and (3.9) then admits the following closed-form solution

$$\hat{\bar{\boldsymbol{\alpha}}} = (\mathbf{P}\bar{\mathbf{K}} + \mu\mathbf{I}_S)^{-1}\mathbf{P}\mathbf{y} \tag{3.11a}$$

$$\hat{\boldsymbol{\beta}} = (\bar{\mathbf{B}}^T\bar{\mathbf{B}})^{-1}\bar{\mathbf{B}}^T(\mathbf{y} - \bar{\mathbf{K}}\hat{\bar{\boldsymbol{\alpha}}}) \tag{3.11b}$$

where $\bar{\mathbf{B}} := \mathbf{S}\mathbf{B}$ and $\mathbf{P} := \mathbf{I}_S - \bar{\mathbf{B}}(\bar{\mathbf{B}}^T\bar{\mathbf{B}})^{-1}\bar{\mathbf{B}}^T$. The complexity of (3.11) is $\mathcal{O}(S^3 + M^3)$.

The ϵ-*insensitive* loss function is given by

$$\mathcal{L}(y_s, f(v_{n_s})) = \max(0, |y_s - f(v_{n_s})| - \epsilon) \tag{3.12}$$

where ϵ is tuned, e.g. via cross-validation, to minimize the generalization error and has well-documented merits in signal estimation from quantized data [37]. Substituting (3.12) into (3.9) yields a convex non-smooth quadratic problem that can be solved efficiently for $\bar{\boldsymbol{\alpha}}$ and $\boldsymbol{\beta}$ using e.g. interior-point methods [33].

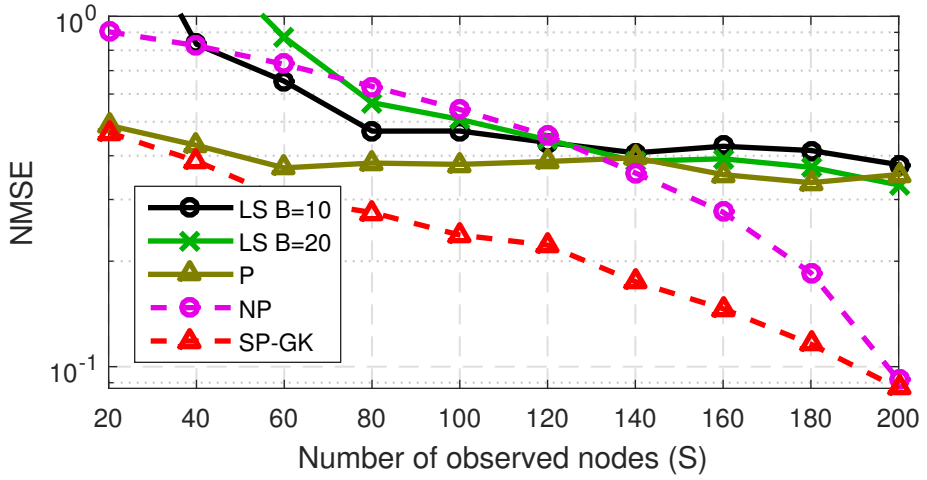


Figure 3.1: NMSE of the synthetic signal estimates. ($\mu = 5 \times 10^{-4}$, $\sigma = 5 \times 10^{-4}$, $\text{SNR}_e = 5\text{dB}$).

3.3 Numerical Tests

This section describes tests on synthetic and real graph functions to illustrate the effective reconstruction performance of our semi-parametric graph kernel estimators, SP-GK and SP-GK(ϵ) resulting from using (3.10) and (3.12) in (3.9) respectively.

Our approach is compared against the *parametric* (P) that considers only the parametric term in (3.2); the *nonparametric* (NP) [3, 4] that considers only the nonparametric term in (3.2); and the *least-squares* estimators (LS) from [9, 13], which assume a bandlimited model with bandwidth B. For all the experiments we use the diffusion kernel (cf. Table 2.1) with parameter σ.

We assess the performance of the proposed estimators via Monte Carlo simulation by comparing the normalized mean-square error (NMSE)

$$\text{NMSE} = \mathbb{E}\left[\frac{\|\hat{\mathbf{f}} - \mathbf{f}\|^2}{\|\mathbf{f}\|^2}\right] \tag{3.13}$$

averaged over choices of sample indices $\{n_s\}_{s=1}^{S}$ and, for synthetic data experiments, also over noise and signal realizations.

3.3.1 Synthetic signals

An Erdős-Rènyi graph with probability of edge presence 0.6 and $N = 200$ nodes was generated, and $\mathbf{f}$ was formed by superimposing a bandlimited [9, 13] with a piecewise constant signal [38]; that is

$$\mathbf{f} = \sum_{i=1}^{10} \gamma_i \mathbf{u}_i + \sum_{i=1}^{6} \delta_i \mathbf{1}_{\mathcal{V}_c} \tag{3.14}$$

where $\{\gamma_i\}_{i=1}^{10}$ and $\{\delta_i\}_{i=1}^{6}$ are standardized Gaussian distributed, $\{\mathbf{u}_i\}_{i=1}^{10}$ are the eigenvectors associated with the 10 smallest eigenvalues of the Laplacian matrix, $\{\mathcal{V}_i\}_{i=1}^{6}$ are the vertex sets of 6 clusters obtained via spectral clustering [39], and $\mathbf{1}_{\mathcal{V}_i}$ is the indicator vector with entries $(\mathbf{1}_{\mathcal{V}_i})_n := 1$ if $v_n \in \mathcal{V}_i$, and 0 otherwise. The parametric basis $\mathcal{B} = \{\mathbf{1}_{\mathcal{V}_i}\}_{i=1}^{6}$ was used by the estimators capturing the prior knowledge, and S vertices were sampled uniformly at random.

In the first experiment, white Gaussian noise e_s of variance σ_e^2 is added to each sample f_s to yield signal-to-noise ratio $\text{SNR}_e := \|\mathbf{f}\|_2^2/(N\sigma_e^2)$. Fig. 3.1 reports the NMSE of all competing methods and showcases the benefits of our semiparametric estimator. Observe that the limited flexibility of the parametric approaches, LS and P, affects their ability to capture the true signal structure. The nonparametric approach (NP) is performing better, but only when the amount of available samples increases. Both our semi-parametric estimators were found to outperform all competing approaches, exhibiting reliable reconstruction even with few samples.

Note here that since the performance of SP-GK(ϵ) and SP-GK was very close, we have chosen to include only SP-GK in Fig. 3.1, to avoid "clotting" the plot. To illustrate the differences of our semi-parametric estimators, we conduct a second experiment which compares the performance of SP-GK and SP-GK(ϵ) in the presence of outlying noise. Each sample f_s is contaminated with Gaussian noise o_s of large variance σ_o^2 with probability $p = 0.1$. Fig. 3.2 demonstrates the robustness of SP-GK(ϵ) which is attributed to the $\epsilon-$insensitive loss function (3.12).

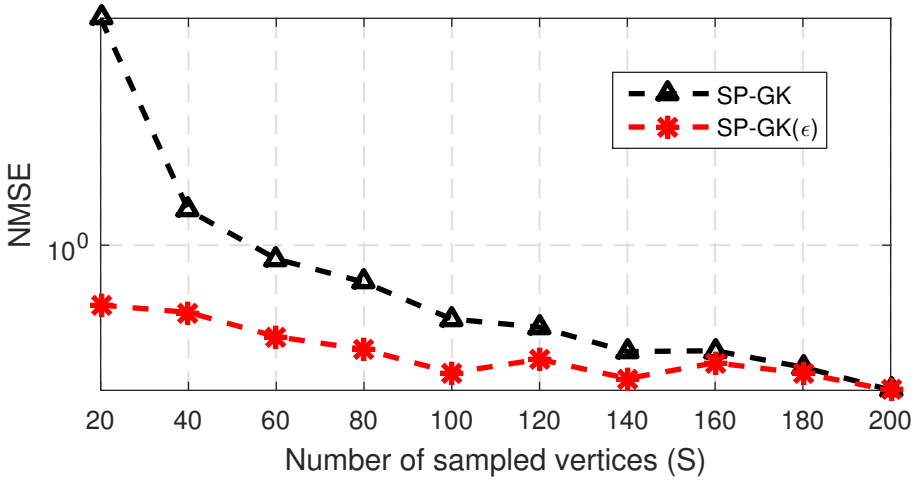


Figure 3.2: NMSE of the synthetic signal estimates. ($\mu = 5 \times 10^{-4}$, $\sigma = 5 \times 10^{-4}$, $\epsilon = 10^{-4}$, and $\text{SNR}_o = -5\text{dB}$).

3.3.2 Real signals

The second dataset is provided by the National Climatic Data Center [40], and comprises temperature measurements at $N = 109$ stations across the continental United States in 2010. The geographical coordinates of the measuring stations have been used to construct a graph

$$(\mathbf{A})_{n,n'} = \frac{\exp\{-d^2_{n,n'}\}}{\sqrt{\sum_{j\in\mathcal{N}_n^k}\exp\{-d^2_{n,j}\}\sum_{l\in\mathcal{N}_{n'}^k}\exp\{-d^2_{n',l}\}}} \tag{3.15}$$

where $\mathcal{N}_n^k$ is the set containing the $k = 7$ nearest neighbors of station n, and $d_{n,n'}$ is the geographical distance between stations n and n'. The neighborhoods are defined based on $d_{n,n'}$, which is justified since geographically close stations tend to measure similar temperature values. To illustrate the benefits of leveraging side information, we cluster the stations into C vertex sets $\{\mathcal{V}_c\}_{c=1}^C$ according to their altitude, and we construct $\mathcal{B}$ using the indicator vectors $\{\mathbf{1}_{\mathcal{V}_c}\}_{c=1}^C$. We sample the temperatures at S stations, chosen uniformly at random, and we reconstruct the signal across all N nodes.

Fig. 3.3 reports the performance of the different graph inference methods, and illustrates the advantage of the proposed approach. SP-GK leverages the altitude information and

SP-GK(ϵ) performed similarly to SP-GK and was not included in the plot.

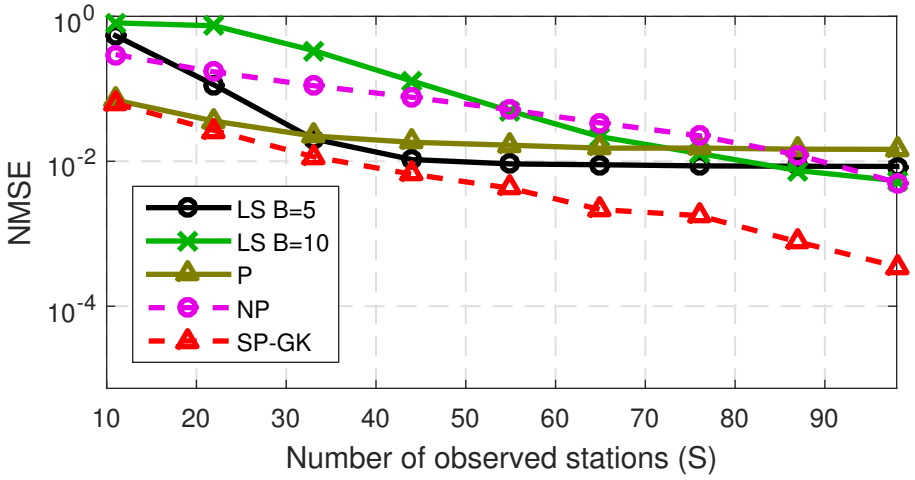


Figure 3.3: NMSE of the mean temperature estimates over 2010. ($\mu = 5 \times 10^{-5}$, $\sigma = 1.3$, and $C = 4$).

achieves NMSE$\leq 10^{-2}$ with as few as $S = 35$ samples, whereas NP requires at least $S = 85$ for the same NMSE. Moreover, we observe that the performance of the pure parametric method LS – which assumes a bandlimited model – does not improve after a certain number of samples. This was expected since the actual signal does not adhere to the modeling assumptions.

Chapter 4

Reconstruction of Dynamic Functions on Dynamic Graphs using Space-time Kernels

The present chapter develops estimators for dynamic functions defined on dynamic graphs. Specifically, the contribution of this chapter is threefold. First, the existing kernel-based learning framework is naturally extended to subsume time-evolving functions over possibly dynamic graphs through the notion of *graph extension*, by which the time dimension receives the same treatment as the spatial dimension. The versatility of kernel-based methods to leverage spatial information [19] is thereby inherited and broadened to account for temporal dynamics as well. Incidentally, this vantage point also accommodates time-varying sampling sets and topologies. Second, two families of *space-time kernels* are introduced by generalizing Laplacian kernels [4]. The first family enables kernel design in a bidimensional frequency domain, whereas the second caters for time-varying topologies. The third contribution comprises two *function estimators* with complementary strengths based on the popular kernel ridge regression (KRR) criterion; see e.g. [19, 33]. Whereas the first can handle more sophisticated forms of spatiotemporal regularization, the second can afford a more efficient implementation and *online* operation, meaning that estimates are refined as

observations become available. The proposed kernel Kalman filter (KKF) finds exact online KRR estimates by implicitly operating in a (possibly) infinite-dimensional space.

The major novelty of this work is a purely deterministic methodology that obviates the need for assumptions on data distributions, stationarity, or knowledge of second-order statistics. The proposed schemes are therefore of special interest in absence of sufficient historical data, yet the latter can be incorporated if available through covariance kernels [19]. Although more complicated dynamics can be accommodated, one may simply rely on the assumption that the target function is smooth over the graph and over time, which is reasonable whenever the graph is properly constructed and the sampling interval is attuned to the temporal dynamics of the function. The novel online estimator constitutes the first fully deterministic rigorous application of the Kalman filter (KF) to kernel-based learning. Although [41] already proposed a kernel-based KF, this work heavily relies on heuristics and approximations to explicitly operate in feature space. Moreover, this algorithm involves solving the challenging preimage problem per time step, which increases inaccuracy and computational cost. Another KF was developed in [42] within the framework of kernel-based learning, but its formulation is probabilistic and requires historical data to estimate the data distribution.

4.1 Preliminaries

A time-varying graph is a tuple $\mathcal{G} := (\mathcal{V}, \{\mathbf{A}_{\mathcal{V}}[t]\}_{t=1}^{T})$, where $\mathcal{V} := \{v_1, \ldots, v_N\}$ is the vertex set and $\mathbf{A}_{\mathcal{V}}[t]$ is an $N \times N$ adjacency matrix whose (n, n')-th entry $A^{\mathcal{V}}_{n,n'}[t]$ assigns a weight to the pair of vertices $(v_n, v_{n'})$ at time t. A time-invariant graph is a special case with $\mathbf{A}_{\mathcal{V}}[t] = \mathbf{A}_{\mathcal{V}}[t']\ \forall t, t'$. As usual, see e.g. [1, Ch. 2], [5, 17], this work assumes that $\mathcal{G}$ (i) has non-negative weights ($A^{\mathcal{V}}_{n,n'}[t] \geq 0\ \forall n, n', t$); (ii) no self-links ($A^{\mathcal{V}}_{n,n}[t] = 0\ \forall n, t$); and, (iii) it is undirected ($A^{\mathcal{V}}_{n,n'}[t] = A^{\mathcal{V}}_{n',n}[t]\ \forall n, n', t$). The edge set is defined as $\mathcal{E}[t] := \{(v_n, v_{n'}) \in \mathcal{V} \times \mathcal{V} : A^{\mathcal{V}}_{n,n'}[t] \neq 0\}$, and two vertices v and v' are said to be *adjacent*, *connected*, or *neighbors* at time t if $(v, v') \in \mathcal{E}[t]$.

See [43] and references therein for alternative representations of time-varying graphs.

A time-evolving function or signal on a graph, is a map $f : \mathcal{V} \times \mathcal{T} \to \mathbb{R}$, where $\mathcal{T} := \{1, \ldots, T\}$ is the set of time indices. The value $f(v_n, t)$ of f at vertex v_n and time t, or its shorthand version $f_n[t]$, can be thought of as the value of an attribute of $v_n \in \mathcal{V}$ at time t. In a social network, $f_n[t]$ may denote the annual income of person v_n at year t. Function values at time t will be collected in $\mathbf{f}[t] := [f_1[t], \ldots, f_N[t]]^T$.

At time t, the vertices with indices in the time-dependent and arbitrary set $\mathcal{S}[t] := \{n_1[t], \ldots, n_{S[t]}[t]\}$, $1 \leq n_1[t] < \cdots < n_{S[t]}[t] \leq N$, are observed. The resulting samples can be expressed as $y_s[t] = f_{n_s[t]}[t] + e_s[t], s = 1, \ldots, S[t]$, where $e_s[t]$ models observation error. In social networks, this encompasses scenarios where a subset of persons have been surveyed about the attribute of interest; e.g. their annual income. By letting $\mathbf{y}[t] := [y_1[t], \ldots, y_{S[t]}[t]]^T$, the observations can be conveniently expressed as

$$\mathbf{y}[t] = \mathbf{S}[t]\mathbf{f}[t] + \mathbf{e}[t], \quad t = 1, \ldots, T \tag{4.1}$$

where $\mathbf{e}[t] := [e_1[t], \ldots, e_{S[t]}[t]]^T$, and the $S[t] \times N$ sampling matrix $\mathbf{S}[t]$ contains ones at positions $(s, n_s[t])$, $s = 1, \ldots, S[t]$ and zeros elsewhere.

The broad goal of this chapter is to "reconstruct" f from the observations $\{\mathbf{y}[t]\}_{t=1}^T$ in (4.1). Two formulations will be considered: in the batch formulation, one aims at finding $\{\mathbf{f}[t]\}_{t=1}^T$ given $\mathcal{G}$, the sample locations $\{\mathbf{S}[t]\}_{t=1}^T$, and all observations $\{\mathbf{y}[t]\}_{t=1}^T$. In the online formulation, one is given $\mathcal{G}$ together with $\mathbf{S}[t]$ and $\mathbf{y}[t]$ at time t. The goal is to find $\mathbf{f}[t]$, possibly based on a previous estimate of $\mathbf{f}[t-1]$, with bounded complexity per time slot t, even if $T \to \infty$. To solve these problems, no explicit parametric model for the temporal or spatial evolution of f will be adopted. For instance, one can solely rely on the assumption that f evolves smoothly over both space and time, yet more structured dynamics can also be incorporated if known.

The entire framework can naturally be extended to accommodate complex-valued functions f.

4.2 Reconstruction of time series on graphs

The framework in Sec. 2.1 cannot accommodate functions evolving over both space and time. This section generalizes this framework through the notion of *graph extension* to flexibly exploit spatial and temporal dynamics.

An immediate approach to reconstructing time-evolving functions is to apply (2.6) separately for each $t = 1, \ldots, T$, yielding the instantaneous estimator (IE)

$$\hat{\mathbf{f}}_{\text{IE}}[t] := \arg\min_{\mathbf{f}} \frac{1}{S[t]} ||\mathbf{y}[t] - \mathbf{S}[t]\mathbf{f}||_2^2 + \mu \mathbf{f}^T \mathbf{K}^\dagger[t]\mathbf{f}. \tag{4.2}$$

Unfortunately, this estimator does not account for the possible relation between e.g. $f_n[t]$ and $f_n[t-1]$. If, for instance, f varies slowly over time, an estimate of $f_n[t]$ may as well benefit from leveraging observations $y_s[\tau]$ at time instants $\tau \neq t$. Exploiting temporal dynamics potentially reduces the number of sampled vertices required to attain a target estimation performance, which in turn can markedly reduce sampling costs.

Incorporating temporal dynamics into kernel-based reconstruction, which can only handle a single snapshot (cf. Sec. 2.1), necessitates an appropriate reformulation of time-evolving function reconstruction as a problem of reconstructing a time-invariant function. An appealing possibility is to replace $\mathcal{G}$ with its *extended version* $\bar{\mathcal{G}} := (\bar{\mathcal{V}}, \bar{\mathbf{A}})$, where each vertex in $\mathcal{V}$ is replicated T times to yield the extended vertex set $\bar{\mathcal{V}} := \{v_n[t],\ n = 1, \ldots, N,\ t = 1, \ldots, T\}$, and the $(n + N(t-1), n' + N(t'-1))$-th entry of the $TN \times TN$ extended adjacency matrix $\bar{\mathbf{A}}$ equals the weight of the edge $(v_n[t], v_{n'}[t'])$. The time-varying function f can thus be replaced with its extended time-invariant counterpart $\bar{f} : \bar{\mathcal{V}} \to \mathbb{R}$ with $\bar{f}(v_n[t]) = f_n[t]$.

This work focuses on graph extensions respecting the connectivity of $\mathcal{G}$ per time slot t, that is, $\{v_n[t]\}_{n=1}^N$ are connected according to $\mathbf{A}_{\mathcal{V}}[t]$, $\forall t$:

Definition 1. *Let $\mathcal{V} := \{v_1, \ldots, v_N\}$ denote a vertex set and let $\mathcal{G} := (\mathcal{V}, \{\mathbf{A}_{\mathcal{V}}[t]\}_{t=1}^T)$ be a time-varying graph. A graph $\bar{\mathcal{G}}$ with vertex set $\bar{\mathcal{V}} := \{v_n[t],\ n = 1, \ldots, N,\ t = 1, \ldots, T\}$ and $NT \times NT$ adjacency matrix $\bar{\mathbf{A}}$ is an extended graph of $\mathcal{G}$ if the t-th $N \times N$ diagonal block of $\bar{\mathbf{A}}$ equals $\mathbf{A}_{\mathcal{V}}[t]$.*

In general, there exist multiple graph extensions for a given time-varying graph. This is because only the diagonal blocks of $\bar{\mathbf{A}}$ are dictated by $\{\mathbf{A}_{\mathcal{V}}[t]\}_{t=1}^{T}$, whereas the remaining entries of $\bar{\mathbf{A}}$ can be freely selected. In the reconstruction problem, one is interested in selecting such off-diagonal entries to capture the space-time dynamics of f. As an example, consider an extended graph with

$$\bar{\mathbf{A}} = \text{btridiag}\{\mathbf{A}_{\mathcal{V}}[1], \ldots, \mathbf{A}_{\mathcal{V}}[T]; \mathbf{B}_{\mathcal{T}}[2], \ldots, \mathbf{B}_{\mathcal{T}}[T]\} \tag{4.3}$$

where $\mathbf{B}_{\mathcal{T}}[t] \in \mathbb{R}_{+}^{N\times N}$ connects $\{v_n[t-1]\}_{n=1}^{N}$ to $\{v_n[t]\}_{n=1}^{N}$, $t = 2, \ldots, T$. For instance, one can connect each vertex to its neighbors at the previous time instant by setting $\mathbf{B}_{\mathcal{T}}[t] = \mathbf{A}_{\mathcal{V}}[t-1]$, or one can connect each vertex to its replicas at adjacent time instants by setting $\mathbf{B}_{\mathcal{T}}[t]$ to be diagonal. Fig. 4.1 pictorially illustrates the latter choice.

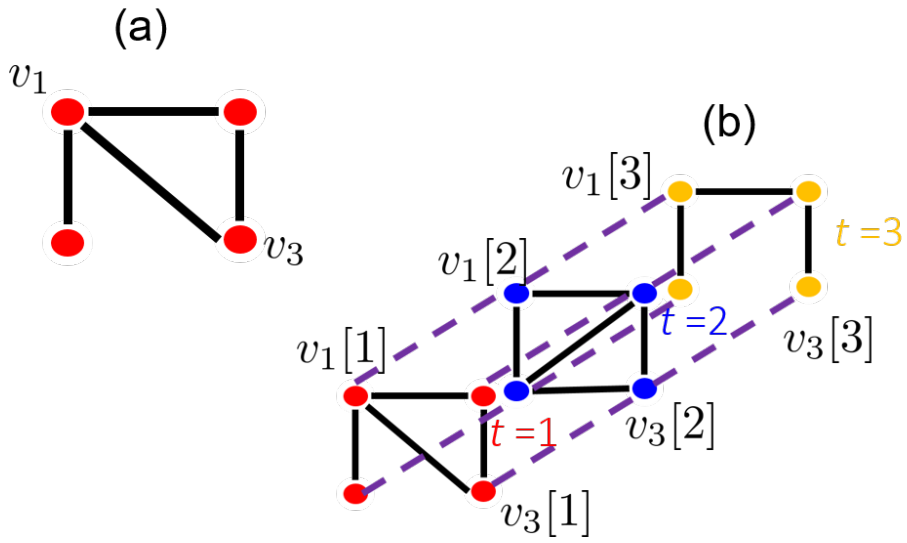


Figure 4.1: (a) Original graph $\mathcal{G}$. (b) Extended graph $\bar{\mathcal{G}}$ for diagonal $\mathbf{B}_{\mathcal{T}}[t]$. Solid lines denote the connections at a specific time instant t and dashed lines the connections between vertices at consecutive t.

Notice that the extended graph treats the time dimension just as the spatial dimension. Thus, the flexibility of graphs to convey relational information carries over to the time domain. As a major benefit, this approach lays the grounds for the design of doubly-selective kernels in Sec. 4.3.1. The extended graph also enables a generalization of the estimators in Sec. 2.1 to reconstruct time-evolving functions. The rest of this section develops two KRR

estimators along these lines.

Consider first the *batch formulation*, where all the $\bar{S} := \sum_{t=1}^{T} S[t]$ samples in $\bar{\mathbf{y}} := [\mathbf{y}^T[1], \ldots, \mathbf{y}^T[T]]^T$ are available, and the goal is to estimate $\bar{\mathbf{f}} := [\mathbf{f}^T[1], \ldots, \mathbf{f}^T[T]]^T$. Directly applying the KRR criterion in (2.6) to reconstruct $\bar{\mathbf{f}}$ on the extended graph $\bar{\mathcal{G}}$ yields

$$\hat{\bar{\mathbf{f}}} := \arg\min_{\bar{\mathbf{f}}} ||\bar{\mathbf{y}} - \bar{\mathbf{S}}\bar{\mathbf{f}}||^2_{\mathbf{D}_S} + \mu \bar{\mathbf{f}}^T \bar{\mathbf{K}}^{\dagger} \bar{\mathbf{f}} \tag{4.4a}$$

where $\bar{\mathbf{K}}$ is now a $TN \times TN$ "space-time" kernel matrix to be designed in Sec. 4.3, $\bar{\mathbf{S}} := \text{bdiag}\{\mathbf{S}[1], \ldots, \mathbf{S}[T]\}$, and $\mathbf{D}_S := \text{bdiag}\left\{S[1]\mathbf{I}_{S[1]}, \ldots, S[T]\mathbf{I}_{S[T]}\right\}$. If $\bar{\mathbf{K}}$ is invertible, (4.4a) can be solved in closed form as

$$\hat{\bar{\mathbf{f}}} = \bar{\mathbf{K}}\bar{\mathbf{S}}^T(\bar{\mathbf{S}}\bar{\mathbf{K}}\bar{\mathbf{S}}^T + \mu \mathbf{D}_S)^{-1}\bar{\mathbf{y}}. \tag{4.4b}$$

For the special $\bar{\mathbf{K}}^{\dagger} = \text{bdiag}\left\{\mathbf{K}^{\dagger}[1], \ldots, \mathbf{K}^{\dagger}[T]\right\}$, where $\mathbf{K}[t]$ is an $N \times N$ kernel matrix for $\mathcal{G}$ at time t, then (4.4a) separates into T sub-problems, each as in (4.2). This implies that only matrices $\bar{\mathbf{K}}^{\dagger}$ with non-zero entries off its block diagonal are capable of accounting for temporal dynamics.

In the *online formulation*, one aims to estimate $\mathbf{f}[t]$ after the $\bar{S}[t] := \sum_{\tau=1}^{t} S[\tau]$ samples in $\bar{\mathbf{y}}[t] := [\mathbf{y}^T[1], \ldots, \mathbf{y}^T[t]]^T$ become available. Based on these samples, the KRR estimate of $\bar{\mathbf{f}}$, denoted as $\hat{\bar{\mathbf{f}}}\Big|t]$, is clearly

$$\begin{aligned} \hat{\bar{\mathbf{f}}}\Big|t] :=& \quad \arg\min_{\bar{\mathbf{f}}} ||\bar{\mathbf{y}}[t] - \bar{\mathbf{S}}[t]\bar{\mathbf{f}}||^2_{\mathbf{D}_S[t]} + \mu \bar{\mathbf{f}}^T \bar{\mathbf{K}}^{-1} \bar{\mathbf{f}} & (4.5a) \\ =& \quad \bar{\mathbf{K}}\bar{\mathbf{S}}^T[t](\bar{\mathbf{S}}[t]\bar{\mathbf{K}}\bar{\mathbf{S}}^T[t] + \mu \mathbf{D}_S[t])^{-1}\bar{\mathbf{y}}[t]. & (4.5b) \end{aligned}$$

where $\bar{\mathbf{K}}$ is assumed invertible for simplicity, $\mathbf{D}_S[t] := \text{bdiag}\left\{S[1]\mathbf{I}_{S[1]}, \ldots, S[t]\mathbf{I}_{S[t]}\right\}$, and $\bar{\mathbf{S}}[t] := [\text{diag}\{\mathbf{S}[1], \ldots, \mathbf{S}[t]\}, \mathbf{0}_{\bar{S}[t] \times (T-t)N}] \in \{0,1\}^{\bar{S}[t] \times TN}$. The estimate in (4.5) comprises the per slot estimates $\{\hat{\mathbf{f}}[\tau|t]\}_{\tau=1}^{T}$; that is, $\hat{\bar{\mathbf{f}}}\Big|t] := [\hat{\mathbf{f}}^T[1|t], \hat{\mathbf{f}}^T[2|t], \ldots, \hat{\mathbf{f}}^T[T|t]]^T$ with $\hat{\mathbf{f}}[\tau|t] = [\hat{f}_1[\tau|t], \ldots, \hat{f}_N[\tau|t]]^T$, where $\hat{\mathbf{f}}[\tau|t]$ (respectively $\hat{f}_n[\tau|t]$) is the KRR estimate of $\mathbf{f}[\tau]$ ($f_n[\tau]$)

given the observations up to time t. Observe that, with this notation, it follows that

$$\hat{\mathbf{f}}[\tau|t] = (\boldsymbol{\imath}_{T,\tau}^T \otimes \mathbf{I}_N)\hat{\bar{\mathbf{f}}}\Big|t] \tag{4.6}$$

for all t, τ.

Regarding t as the present, (4.5) therefore provides estimates of past, present, and future values of f. The solution to the online problem formulated in Sec. 4.1 includes the sequence of present KRR estimates for all t, that is, $\{\hat{\mathbf{f}}[t|t]\}_{t=1}^T$. This can be obtained by solving (4.5a) in closed form per t as in (4.5b) and then applying (4.6). However, such an approach does not yield a desirable online algorithm since its complexity per time slot is cubic in t (see Remark 1) and therefore increasing with t. For this reason, this approach is not satisfactory since the online problem formulation in Sec. 4.1 requires the complexity per time slot of the desired algorithm to be bounded. An algorithm that does satisfy this bounded-complexity requirement yet provides the exact KRR estimate is developed next for the case where the kernel matrix is any positive definite matrix $\bar{\mathbf{K}}$ satisfying

$$\bar{\mathbf{K}}^{-1} = \text{btridiag}\{\mathbf{D}[1], \ldots, \mathbf{D}[T]; \mathbf{C}[2], \ldots, \mathbf{C}[T]\} \tag{4.7}$$

for some $N \times N$ matrices $\{\mathbf{D}[t]\}_{t=1}^T$ and $\{\mathbf{C}[t]\}_{t=2}^T$. Kernels in this important family are designed in Sec. 4.3. Broader classes of kernels can be accommodated as described in Remark 3.

The process of developing the desired online algorithm involves two steps. The first step expresses (4.5a) as a weighted least-squares problem amenable to a KF solver. In the second step, the KF is applied to solve such a problem. The first step is accomplished by the following result.

Lemma 1. *For $\bar{\mathbf{K}}$ of the form (4.7), the KRR criterion in (4.5a) boils down to the following*

regularized weighted least-squares objective

$$\hat{\hat{\mathbf{f}}}\Big|t] = \underset{\{\mathbf{f}[\tau]\}_{\tau=1}^{T}}{\arg\min} \sum_{\tau=1}^{t} \frac{1}{\sigma_e^2[\tau]} ||\mathbf{y}[\tau] - \mathbf{S}[\tau]\mathbf{f}[\tau]||^2 + \sum_{\tau=2}^{T} ||\mathbf{f}[\tau] - \mathbf{P}[\tau]\mathbf{f}[\tau-1]||^2_{\mathbf{\Sigma}[\tau]} + \mathbf{f}^T[1]\mathbf{\Sigma}^{-1}[1]\mathbf{f}[1]. \tag{4.8}$$

Proof. See Appendix A.1. □

Relative to (4.5a), matrices $\{\mathbf{D}[\tau], \mathbf{C}[\tau]\}$ in $\bar{\mathbf{K}}^{-1}$ have been replaced in (B.6) with matrices $\{\mathbf{\Sigma}[\tau], \mathbf{P}[\tau]\}$, which can be found through Algorithm 1.

Although no probabilistic assumption is required throughout the derivation of the proposed online algorithm, exploring the link between (B.6) and the conventional probabilistic setup for state estimation provides the intuition behind why (B.6) can be solved through Kalman filtering. To this end, suppose that $\mathbf{f}[\tau]$ obeys the random model $\mathbf{f}[\tau] = \mathbf{P}[\tau]\mathbf{f}[\tau-1] + \boldsymbol{\eta}[\tau]$ for $\tau = 2, \ldots, T$, initialized by $\mathbf{f}[1] = \boldsymbol{\eta}[1]$, with zero-mean noise $\boldsymbol{\eta}[\tau]$ having covariance $\mathbf{\Sigma}[\tau]$, and the observations follow the model $\mathbf{y}[\tau] = \mathbf{S}[t]\mathbf{f}[\tau] + \mathbf{e}[\tau]$ for $\tau = 1, \ldots, T$, with $\mathbf{e}[\tau]$ zero-mean noise having covariance $\sigma_e^2[\tau]\mathbf{I}$. In this state estimation problem, $\mathbf{P}[\tau]$ is referred to as the state-transition matrix. In this scenario, one can easily see that obtaining the maximum a posteriori (MAP) and the minimum mean square error (MMSE) estimators of $\hat{\hat{\mathbf{f}}}$ given the observations up to time T when $\{\boldsymbol{\eta}[\tau], \mathbf{e}[\tau]\}_{\tau=1}^{T}$ are Gaussian distributed reduces to minimizing (B.6). This link suggests that (B.6) can be minimized using the celebrated KF [44, Ch. 17].

The following result formalizes the latter claim. The resulting algorithm, termed KKF, is summarized as Algorithm 2. In the probabilistic KF terminology, step 3 yields the prediction of $\mathbf{f}[t]$, step 4 provides the covariance matrix of the prediction error, step 5 yields the Kalman gain, step 6 returns the posterior estimate upon correcting the prediction with the innovations scaled by the Kalman gain, and step 7 finds the error of this posterior estimate.

Theorem 1. *For $\bar{\mathbf{K}}$ of the form (4.7), the KKF Algorithm 2 returns the sequence $\{\hat{\mathbf{f}}[t|t]\}_{t=1}^{T}$, where $\hat{\mathbf{f}}[t|t]$ is given by (4.6).*

Algorithm 1 Recursion to set parameters of KKF

Input: $\mathbf{D}[t]$, $t = 1, \ldots, T$, $\mathbf{C}[t]$, $t = 2, \ldots, T$.

1: **Set** $\boldsymbol{\Sigma}^{-1}[T] = \mathbf{D}[T]$
2: **for** $t = T,\ T-1, \ldots, 2$ **do**
3: $\quad \mathbf{P}[t] = -\boldsymbol{\Sigma}[t]\mathbf{C}[t]$
4: $\quad \boldsymbol{\Sigma}^{-1}[t-1] = \mathbf{D}[t-1] - \mathbf{P}^T[t]\boldsymbol{\Sigma}^{-1}[t]\mathbf{P}[t]$

Output: $\boldsymbol{\Sigma}[t]$, $t = 1, \ldots, T$, $\mathbf{P}[t]$, $t = 2, \ldots, T$

Algorithm 2 Kernel Kalman filter (KKF)

Input: $\{\boldsymbol{\Sigma}[t] \in \mathbb{S}_+^N\}_{t=1}^T$, $\{\mathbf{P}[t] \in \mathbb{R}^{N\times N}\}_{t=2}^T$, $\{\mathbf{y}[t] \in \mathbb{R}^{S[t]}\}_{t=1}^T$, $\{\mathbf{S}[t] \in \{0,1\}^{S[t]\times N}\}_{t=1}^T$, $\{\sigma_e^2[t] > 0\}_{t=1}^T$.

1: **Set** $\hat{\mathbf{f}}[0|0] = \mathbf{0}$, $\mathbf{M}[0|0] = \mathbf{0}$, $\mathbf{P}[1] = \mathbf{0}$
2: **for** $t = 1, \ldots, T$ **do**
3: $\quad \hat{\mathbf{f}}[t|t-1] = \mathbf{P}[t]\hat{\mathbf{f}}[t-1|t-1]$
4: $\quad \mathbf{M}[t|t-1] = \mathbf{P}[t]\mathbf{M}[t-1|t-1]\mathbf{P}^T[t] + \boldsymbol{\Sigma}[t]$
5: $\quad \mathbf{G}[t] = \mathbf{M}[t|t-1]\mathbf{S}^T[t](\sigma_e^2[t]\mathbf{I} + \mathbf{S}[t]\mathbf{M}[t|t-1]\mathbf{S}^T[t])^{-1}$
6: $\quad \hat{\mathbf{f}}[t|t] = \hat{\mathbf{f}}[t|t-1] + \ \mathbf{G}[t](\mathbf{y}[t] - \mathbf{S}[t]\hat{\mathbf{f}}[t|t-1])$
7: $\quad \mathbf{M}[t|t] = (\mathbf{I} - \mathbf{G}[t]\mathbf{S}[t])\mathbf{M}[t|t-1]$

Output: $\hat{\mathbf{f}}[t|t]$, $t = 1, \ldots, T$; $\mathbf{M}[t]$, $t = 1, \ldots, T$.

Proof. See Appendix A.2. □

Recapitulating, given $\bar{\mathbf{K}}^{-1}$ in (4.7), one just has to run Algorithms 1 and 2 to find the online KRR estimate of f given by (4.6). Since the proposed KKF is derived within a fully deterministic framework, notions such as mean, covariance, statistical independence, or mean-square error are not required, yet they have been used to describe the connection with the classical KF. Furthermore, the proposed KKF does not explicitly involve any state-space model, which is a major novelty and indeed a surprising result of the present work.

The proposed KKF generalizes the probabilistic KF since the latter is recovered upon setting $\bar{\mathbf{K}}$ to be the covariance matrix of $\bar{\mathbf{f}}$ in the previously mentioned probabilistic setup. It is therefore natural that the assumptions required by the probabilistic KF are stronger than those involved in the KKF. Specifically, in the probabilistic KF, $\mathbf{f}[t]$ must adhere to

a linear state-space model with known transition matrix $\mathbf{P}[t]$, where the state noise $\boldsymbol{\eta}[t]$ is uncorrelated over time and has known covariance matrix $\boldsymbol{\Sigma}[t]$, and the observation noise $\mathbf{y}[t]$ must be uncorrelated over time and have known covariance matrix. Correspondingly, the performance guarantees of the probabilistic KF are also stronger: the resulting estimate is optimal in the mean-square error sense among all linear estimators. Furthermore, if $\boldsymbol{\eta}[t]$ and $\mathbf{y}[t]$ are jointly Gaussian, $t = 1, \ldots, T$, then the probabilistic KF estimate is optimal in the mean-square error sense among all (not necessarily linear) estimators. In contrast, the requirements of the proposed KKF are much weaker since it only requires f to evolve smoothly with respect to a given extended graph, but the guarantees are also weaker; see e.g. [33, Ch. 5]. However, since the KKF generalizes the probabilistic KF, the reconstruction performance of the former for judiciously selected $\bar{\mathbf{K}}$ cannot be worse than the reconstruction performance of the latter for any given criterion. The caveat, however, is that such a selection is not necessarily easy.

Remark 1. *Algorithm 2 requires $\mathcal{O}(N^3)$ operations per time slot, whereas the complexity of evaluating (4.5b) for the t-th time slot is $\mathcal{O}(\bar{S}^3[t])$, which increases with t and becomes eventually prohibitive. For large t, Algorithm 2 is computationally more efficient than a* single *plain evaluation of (4.5b): whereas the overall complexity of the former is $\mathcal{O}(tN^3)$, the latter is $\mathcal{O}(NT\bar{S}^2[t])$, which e.g. for constant $S[t] = S$ is $\mathcal{O}(NTt^2S^2)$.*

Remark 2. *Algorithm 2 provides estimates of the form $\hat{\mathbf{f}}[t|t]$ and $\hat{\mathbf{f}}[t|t-1]$. To obtain estimates $\hat{\mathbf{f}}[t|t']$ for $t > t' + 1$, one may set $\mathcal{S}[\tau] = \emptyset$ for $\tau > t' + 1$ and execute Algorithm 2 up to time t. Conversely, to obtain estimates $\hat{\mathbf{f}}[t|t']$ for which $t < t'$, one may extend Algorithm 2 by capitalizing on the notion of Kalman smoothing [45].*

Remark 3. *Similar to the probabilistic KF, which requires the inverse covariance matrix of $\bar{\mathbf{f}}$ to be block tridiagonal, the proposed KKF requires the inverse kernel matrix to be of the form (4.7). Fortunately, it is straightforward to extend both algorithms to accommodate inverse covariance or kernel matrices with any number of non-zero diagonals at the price of increasing the time interval between consecutive estimates. To illustrate such an approach, suppose that $\bar{\mathbf{K}}^{-1}$ is not block tridiagonal when blocks are of size $N \times N$, but it is block*

tridiagonal if blocks are of size $2N \times 2N$. In such a case, one can use the proposed KKF to estimate $\{\mathbf{f}'[t']\}_{t'=1}^{T/2}$, where $\mathbf{f}'[t'] := [\mathbf{f}^T[2t'-1], \mathbf{f}^T[2t']]^T \in \mathbb{R}^{2N}$, just by replacing $\mathbf{y}[t]$ with $\mathbf{y}'[t'] := [\mathbf{y}^T[2t'-1], \mathbf{y}^T[2t']]^T$, $\mathbf{S}[t]$ with $\mathbf{S}'[t'] := \text{bdiag}\,\{\mathbf{S}[2t'-1]\mathbf{S}[2t']\}$, and $\mathbf{e}[t]$ with $\mathbf{e}'[t'] := [\mathbf{e}^T[2t'-1], \mathbf{e}^T[2t']]^T$, $t' = 1, \ldots, T/2$. Note that the sampling interval associated with the index t' is twice that associated with t.

4.3 Design of space-time kernels

Sec. 4.2 assumed that the kernel matrix $\bar{\mathbf{K}}$ is given and described no methodology to address its design. An immediate approach is to mimic the Laplacian kernels of Sec. 2.1 by setting $\bar{\mathbf{K}} = r^\dagger(\bar{\mathbf{L}})$, where $\bar{\mathbf{L}} := \text{diag}\{\bar{\mathbf{A}}\mathbf{1}\} - \bar{\mathbf{A}}$ denotes the Laplacian matrix of the extended graph. Unfortunately, such a design prevents separate control of the spatial and temporal variability of the estimates, thus limiting the user's ability to flexibly account for spatial and temporal information. For instance, sampling intervals that are small relative to the time dynamics of f, meaning that f does not vary significantly between samples $t-1$ and t, favors estimates that sacrifice spatial smoothness to increase temporal smoothness.

This section proposes families of space-time kernels for which temporal and spatial smoothness can be separately tuned. Sec. 4.3.1 describes designs for time-invariant topologies, whereas Sec. 4.3.2 deals with the time-varying case.

4.3.1 Doubly-selective space-time kernels

In Sec. 2.1, the frequency interpretation of (2.4) proved decisive to interpret and design Laplacian kernels for reconstructing time-invariant functions. Introducing the time dimension in Sec. 4.2 prompts an analogous methodology, where kernels are specified in a bidimensional plane of spatio-temporal frequency; see [46] for graph filter design in this domain. This section accomplishes this task by generalizing the Laplacian kernels from Sec. 2.1. How much the regularizers $\rho(\bar{\mathbf{f}}) = \bar{\mathbf{f}}^T\bar{\mathbf{K}}^\dagger\bar{\mathbf{f}}$ associated with the proposed kernels weight each spatial and temporal frequency component of $\bar{\mathbf{f}}$ can be separately prescribed. Throughout this section, a time-invariant topology will be assumed, i.e., $\mathbf{A}_\mathcal{V}[t] = \mathbf{A}_\mathcal{V},\ t = 1, \ldots, T$.

Clearly, (2.5a) can be rewritten as $\rho_{\text{LK}}(\mathbf{f}) = \mathbf{r}^T(\check{\mathbf{f}} \odot \check{\mathbf{f}})$ for $\check{\mathbf{f}} := \mathbf{U}_{\mathcal{V}}^T\mathbf{f}$ the frequency transform of $\mathbf{f}$ and $\mathbf{r} := [r(\lambda_1^{\mathcal{V}}), \ldots, r(\lambda_N^{\mathcal{V}})]^T$. One can separately weight each frequency component by selecting $\mathbf{r}$, which can be thought of as the "frequency response" of the regularizer. For instance, one may promote low pass estimates by setting the first entries of $\mathbf{r}$ to low values and the rest to high values.

Inspired by this view, one may seek kernels $\bar{\mathbf{K}}$ for which

$$\rho(\bar{\mathbf{f}}) = \bar{\mathbf{f}}^T\bar{\mathbf{K}}^{\dagger}\bar{\mathbf{f}} = \text{Tr}\left(\mathbf{R}^T(\breve{\bar{\mathbf{F}}} \odot \breve{\bar{\mathbf{F}}})\right) \tag{4.9}$$

where $\mathbf{R}$ and $\breve{\bar{\mathbf{F}}}$ are $N \times T$ matrices to be specified later respectively containing the frequency response of the regularizer and the bidimensional transform of f. The $(\check{n}, \check{t})$-th entry of these matrices corresponds to the $\check{n}$-th spatial frequency and $\check{t}$-th temporal frequency. Kernels satisfying the second equality in (4.9) will be termed *doubly (frequency) selective*. Such kernels preserve the flexibility of their counterparts for time-invariant functions. For instance, if $\bar{\mathbf{K}}$ promotes doubly low-pass estimates, then the top left entries of $\mathbf{R}$ are small whereas the rest are large.

To determine the form of a doubly-selective kernel, let $\bar{\mathbf{F}} := [\mathbf{f}[1], \ldots, \mathbf{f}[T]]$ and recall that a linear bidimensional transform can be expressed as $\breve{\bar{\mathbf{F}}} := \mathbf{U}_{\mathcal{V}}^T\bar{\mathbf{F}}\mathbf{U}_{\mathcal{T}}$, where the $N \times N$ matrix $\mathbf{U}_{\mathcal{V}}$ and the $T \times T$ matrix $\mathbf{U}_{\mathcal{T}}$ stand for orthogonal transformations along space and time, respectively. On the other hand, vectorizing the rightmost term of (4.9) yields

$$\rho(\bar{\mathbf{f}}) = \bar{\mathbf{f}}^T\bar{\mathbf{K}}^{\dagger}\bar{\mathbf{f}} = \breve{\bar{\mathbf{f}}}^T \,\text{diag}\,\{\mathbf{r}\}\,\breve{\bar{\mathbf{f}}} \tag{4.10}$$

where $\mathbf{r} := \text{vec}\{\mathbf{R}\}$ and

$$\breve{\bar{\mathbf{f}}} := \text{vec}\{\breve{\bar{\mathbf{F}}}\} = \text{vec}\{\mathbf{U}_{\mathcal{V}}^T\bar{\mathbf{F}}\mathbf{U}_{\mathcal{T}}\} = (\mathbf{U}_{\mathcal{T}} \otimes \mathbf{U}_{\mathcal{V}})^T\bar{\mathbf{f}}. \tag{4.11}$$

Any doubly-selective kernel, or equivalently any kernel satisfying the second equality of

(4.10), is therefore of the form

$$\bar{\mathbf{K}}^{\dagger} = (\mathbf{U}_{\mathcal{T}} \otimes \mathbf{U}_{\mathcal{V}}) \operatorname{diag}\{\mathbf{r}\} (\mathbf{U}_{\mathcal{T}} \otimes \mathbf{U}_{\mathcal{V}})^T \tag{4.12}$$

for some orthogonal $N \times N$ matrix $\mathbf{U}_{\mathcal{T}}$, some orthogonal $T \times T$ matrix $\mathbf{U}_{\mathcal{V}}$, and some entrywise non-negative vector $\mathbf{r}$.

Expression (4.12) provides the general form of a doubly-selective kernel, but a specific construction for $\mathbf{U}_{\mathcal{T}}$, $\mathbf{U}_{\mathcal{V}}$, and $\mathbf{r}$ capturing the spatiotemporal dynamics of f is still required. The next procedure serves this purpose by paralleling the approach in Sec. 2.1. This involves the following two steps.

- **S1**: Since a Laplacian kernel matrix shares eigenvectors with the Laplacian matrix, one should construct an extended graph $\bar{\mathcal{G}}$ so that its Laplacian matrix $\bar{\mathbf{L}}$ is diagonalizable by a matrix of the form $\mathbf{U}_{\mathcal{T}} \otimes \mathbf{U}_{\mathcal{V}}$ for some orthogonal $\mathbf{U}_{\mathcal{T}} \in \mathbb{R}^{T \times T}$ and $\mathbf{U}_{\mathcal{V}} \in \mathbb{R}^{N \times N}$.

- **S2**: One must design a spectral weight map r to obtain the eigenvalues of $\bar{\mathbf{K}}$ from those of $\bar{\mathbf{L}}$.

Regarding S1, an explicit construction of an extended graph whose Laplacian matrix is diagonalizable by a matrix of the form $\mathbf{U}_{\mathcal{T}} \otimes \mathbf{U}_{\mathcal{V}}$ with orthogonal $\mathbf{U}_{\mathcal{T}} \in \mathbb{R}^{T \times T}$ and $\mathbf{U}_{\mathcal{V}} \in \mathbb{R}^{N \times N}$ is provided next. To this end, consider the extended adjacency matrix

$$\bar{\mathbf{A}} = \mathbf{A}_{\mathcal{T}} \oplus \mathbf{A}_{\mathcal{V}} \tag{4.13}$$

where $\mathbf{A}_{\mathcal{V}}$ is the given adjacency matrix of $\mathcal{G}$ and the $T \times T$ adjacency matrix $\mathbf{A}_{\mathcal{T}}$ is selected to capture temporal dynamics. Specifically, with $\bar{\mathbf{A}}$ as in (4.13), the definition of extended adjacency matrix in Sec. 4.2 dictates that the weight of the edge $(v_{n_1}[t], v_{n_2}[t])$ for all t is given by the (n_1, n_2)-th entry of $\mathbf{A}_{\mathcal{V}}$, whereas the weight of the edge $(v_n[t_1], v_n[t_2])$ for all n is given by the (t_1, t_2)-th entry of $\mathbf{A}_{\mathcal{T}}$. A simple choice for $\mathbf{A}_{\mathcal{T}}$ will be described later. Note that (4.13) differs from *Kronecker graphs* [47], for which $\bar{\mathbf{A}} = \mathbf{A}_{\mathcal{T}} \otimes \mathbf{A}_{\mathcal{V}}$, although it can be interpreted as the *Cartesian graph* of $\mathcal{V}$ and $\{1, \ldots, T\}$ [48, 49]. Cartesian graphs

have been considered in the graph signal processing literature for graph filtering and Fourier transforms of time-varying functions [49], but not for signal reconstruction.

With $\bar{\mathbf{A}}$ as in (4.13), it can be readily seen that $\bar{\mathbf{L}} := \operatorname{diag}\{\bar{\mathbf{A}}\mathbf{1}\} - \bar{\mathbf{A}} = \mathbf{L}_\mathcal{T} \oplus \mathbf{L}_\mathcal{V}$, where $\mathbf{L}_\mathcal{T} := \operatorname{diag}\{\mathbf{A}_\mathcal{T}\mathbf{1}\} - \mathbf{A}_\mathcal{T}$ and $\mathbf{L}_\mathcal{V} := \operatorname{diag}\{\mathbf{A}_\mathcal{V}\mathbf{1}\} - \mathbf{A}_\mathcal{V}$ are the Laplacian matrices associated with $\mathbf{A}_\mathcal{T}$ and $\mathbf{A}_\mathcal{V}$, respectively. If $\mathbf{L}_\mathcal{T} = \mathbf{U}_\mathcal{T} \operatorname{diag}\{\boldsymbol{\lambda}_\mathcal{T}\} \mathbf{U}_\mathcal{T}^T$ and $\mathbf{L}_\mathcal{V} = \mathbf{U}_\mathcal{V} \operatorname{diag}\{\boldsymbol{\lambda}_\mathcal{V}\} \mathbf{U}_\mathcal{V}^T$, then

$$\begin{aligned}\bar{\mathbf{L}} &= (\mathbf{U}_\mathcal{T} \otimes \mathbf{U}_\mathcal{V}) \left[\operatorname{diag}\{\boldsymbol{\lambda}_\mathcal{T}\} \oplus \operatorname{diag}\{\boldsymbol{\lambda}_\mathcal{V}\}\right] (\mathbf{U}_\mathcal{T} \otimes \mathbf{U}_\mathcal{V})^T \\ &= (\mathbf{U}_\mathcal{T} \otimes \mathbf{U}_\mathcal{V}) \operatorname{diag}\{\boldsymbol{\lambda}_\mathcal{T} \otimes \mathbf{1}_N + \mathbf{1}_T \otimes \boldsymbol{\lambda}_\mathcal{V}\} (\mathbf{U}_\mathcal{T} \otimes \mathbf{U}_\mathcal{V})^T.\end{aligned}$$

This expression reveals that the graph extension proposed in (4.13) indeed satisfies the objective of S1, which requires the eigenvector matrix of $\bar{\mathbf{L}}$ to be of the form $\mathbf{U}_\mathcal{T} \otimes \mathbf{U}_\mathcal{V}$. Thus, it is always possible to construct a graph extension satisfying the goal of S1.

For S2, one must construct a spectral map r that yields $\mathbf{r}$ upon entrywise application to $\boldsymbol{\lambda}_\mathcal{T} \otimes \mathbf{1}_N + \mathbf{1}_T \otimes \boldsymbol{\lambda}_\mathcal{V}$. To separately control the frequency response along the spatial and temporal frequencies $\lambda_\mathcal{V}$ and $\lambda_\mathcal{T}$, such a map must take two arguments as $r(\lambda_\mathcal{T}, \lambda_\mathcal{V})$. This results in $\mathbf{r} = r(\boldsymbol{\lambda}_\mathcal{T} \otimes \mathbf{1}_N, \mathbf{1}_T \otimes \boldsymbol{\lambda}_\mathcal{V})$ and (4.12) becomes

$$\bar{\mathbf{K}}^\dagger = (\mathbf{U}_\mathcal{T} \otimes \mathbf{U}_\mathcal{V}) \operatorname{diag}\{r(\boldsymbol{\lambda}_\mathcal{T} \otimes \mathbf{1}_N, \mathbf{1}_T \otimes \boldsymbol{\lambda}_\mathcal{V})\} (\mathbf{U}_\mathcal{T} \otimes \mathbf{U}_\mathcal{V})^T. \tag{4.14}$$

Kernels of this form will be referred to as *Kronecker space-time* kernels. The transformation r can be selected in several ways. For instance, the immediate construction at the beginning of Sec. 4.3 is recovered for $r(\lambda_\mathcal{T}, \lambda_\mathcal{V}) = r(\lambda_\mathcal{T} + \lambda_\mathcal{V})$, with $r(\lambda)$ a one-dimensional spectral weight map such as the ones in Table 2.1. Another possibility is to focus on separable maps of the form $r(\lambda_\mathcal{T}, \lambda_\mathcal{V}) = r_\mathcal{T}(\lambda_\mathcal{T}) r_\mathcal{V}(\lambda_\mathcal{V})$ where $r_\mathcal{T}$ and $r_\mathcal{V}$ denote one-dimensional spectral maps. The resulting Kronecker kernel can expressed as

$$\bar{\mathbf{K}} = \mathbf{K}_\mathcal{T} \otimes \mathbf{K}_\mathcal{V} \tag{4.15}$$

The notion of Kronecker kernels together with (4.15) shows up in the literature of pairwise classification [48], but the resemblance is merely illusional since the underlying kernel is a function of two *pairs* of vertices.

where $\mathbf{K}_{\mathcal{T}} := \mathbf{U}_{\mathcal{T}} \operatorname{diag}\{r_{\mathcal{T}}(\boldsymbol{\lambda}_{\mathcal{T}})\}\mathbf{U}_{\mathcal{T}}^T$ and $\mathbf{K}_{\mathcal{V}} := \mathbf{U}_{\mathcal{V}} \operatorname{diag}\{r_{\mathcal{V}}(\boldsymbol{\lambda}_{\mathcal{V}})\}\mathbf{U}_{\mathcal{V}}^T$. For example, doubly bandlimited estimates can be obtained by setting both $\mathbf{K}_{\mathcal{T}}$ and $\mathbf{K}_{\mathcal{V}}$ to be bandlimited kernels (Table 2.1). A further possibility is to consider maps of the form $r(\lambda_{\mathcal{T}}, \lambda_{\mathcal{V}}) = r_{\mathcal{T}}(\lambda_{\mathcal{T}}) + r_{\mathcal{V}}(\lambda_{\mathcal{V}})$, which clearly result in kernels of the form

$$\bar{\mathbf{K}}^{\dagger} = \mathbf{K}_{\mathcal{T}}^{\dagger} \oplus \mathbf{K}_{\mathcal{V}}^{\dagger}. \tag{4.16}$$

To sum up, the proposed Kronecker kernels arise from an intuitive graph extension and can afford flexible adjustment of their frequency response. Unfortunately, not any Kronecker kernel is suitable for the online algorithm in Sec. 4.2 since the latter requires the inverse of the kernel matrix $\bar{\mathbf{K}}$ to be block tridiagonal. The rest of this section describes a subfamily of Kronecker kernels that is suitable for this algorithm.

Clearly, in order for $\bar{\mathbf{K}}^{\dagger}$ as in (4.15) or (4.16) to be block tridiagonal, it is necessary that $\mathbf{K}_{\mathcal{T}}^{\dagger}$ be tridiagonal, i.e., the (t, t')-th entry of $\mathbf{K}_{\mathcal{T}}^{\dagger}$ must be zero if $|t - t'| > 1$. Such a $\mathbf{K}_{\mathcal{T}}^{\dagger}$ can be obtained if, for instance, one sets the (t, t')-th entry of $\mathbf{A}_{\mathcal{T}}$ to be 0 unless $|t - t'| = 1$. In this extended graph construction, vertex $v_n[t]$, $1 < t < T$, is connected to $v_n[t-1]$ and $v_n[t+1]$, which are its replicas in adjacent time slots. For $\mathbf{K}_{\mathcal{T}}^{\dagger}$ to be tridiagonal, one may set $r_{\mathcal{T}}(\lambda_{\mathcal{T}}) = \lambda_{\mathcal{T}} + \epsilon$, where $\epsilon > 0$ ensures that $\mathbf{K}_{\mathcal{T}}$ is invertible.

Thus, the price to be paid for an online implementation with the KKF from Sec. 4.2 is limited flexibility in specifying the temporal frequency response. Note that this is not an intrinsic limitation of the proposed algorithm, but it is inherent to the classical KF as well; just recall that the latter assumes vector autoregressive processes of order 1. In any case, the temporal frequency response of a kernel for which $(\mathbf{A}_{\mathcal{T}})_{t,t'} = \delta[|t - t'| - 1]$ can be obtained analytically by approximating the resulting Laplacian $\mathbf{L}_{\mathcal{T}}$ for sufficiently large T with a circulant matrix. This implies that (i) the eigenvectors in $\mathbf{U}_{\mathcal{T}}$ are approximately those in the conventional Fourier basis and therefore the notion of temporal frequency embodied in $\mathbf{U}_{\mathcal{T}}$ preserves its conventional meaning; and (ii), upon applying [19, Example 3], the resulting frequency response is low pass. Both (i) and (ii) are intuitively reasonable. Thus, although the KKF solves only a subset of KRR problems, this subset is of practical relevance.

Remark 4. *In this work, the rows of* $\bar{\mathbf{F}}$ *can be thought of as graph functions over a graph with adjacency matrix* $\mathbf{A}_{\mathcal{T}}$*, whereas the columns of* $\bar{\mathbf{F}}$ *can be thought of as graph functions over the graph with adjacency matrix* $\mathbf{A}_{\mathcal{V}}$*. In principle, each column of* $\bar{\mathbf{F}}$ *does not need to correspond to a different time instant, but e.g. to a different movie in a recommender system application. The estimators (4.4a)-(4.5b) can therefore be used for matrix completion upon properly creating an extended graph and graph kernel matrix. Towards this end, the space-time kernels defined in (4.15) and (4.16) readily generalize to space-space kernels that promote smoothness over both graphs.*

4.3.2 Space-time kernels for time-varying topologies

For time-invariant topologies, Sec. 4.3.1 proposed kernels that can be designed and interpreted on a two-dimensional frequency plane. This section deals with changing topologies, for which no bidimensional frequency notion can be defined.

To recognize this claim, suppose that $\mathbf{A}_{\mathcal{V}}[t] = \mathbf{A}_{\mathcal{V}}$ remains constant over t and recall that $\mathbf{u}_{\check{n}}^{\mathcal{V}}$ is the $\check{n}$-th eigenvector of $\mathbf{L}_{\mathcal{V}}$ or, equivalently, the $\check{n}$-th column of $\mathbf{U}_{\mathcal{V}}$. In this case, a bidimensional transform exists and can be expressed as $\check{\bar{\mathbf{F}}} := \mathbf{U}_{\mathcal{V}}^T \bar{\mathbf{F}} \mathbf{U}_{\mathcal{T}}$, whose $(\check{n}, \check{t})$-th entry corresponds to the $\check{n}$-th spatial frequency and $\check{t}$-th temporal frequency. Fundamentally, the precise meaning of the latter statement is that $(\check{\bar{\mathbf{F}}})_{\check{n},\check{t}}$ is the $\check{t}$-th temporal frequency component of the $\check{n}$-th spatial frequency component of f, i.e., the $\check{t}$-th temporal frequency component of the time series $\{\check{f}_{\check{n}}[t] := (\mathbf{u}_{\check{n}}^{\mathcal{V}})^T \mathbf{f}[t]\}_{t=1}^T$, which is the time evolution of the $\check{n}$-th *spatial* frequency component of f. However, for changing topologies one cannot generally conceive the temporal evolution of a specific spatial frequency component since the eigenvectors of $\mathbf{L}_{\mathcal{V}}[t]$ generally differ from those of $\mathbf{L}_{\mathcal{V}}[t']$, thus precluding any natural definition of the aforementioned sequence and therefore of a bidimensional frequency transform. Nonetheless, it is shown next that the notion of spatial frequency per slot t can still be utilized to design space-time kernels for time-varying topologies.

To this end, consider the extended graph defined by (4.3) for arbitrary $\mathbf{B}_{\mathcal{T}}[t] \in \mathbb{R}_+^{N \times N}$.

It then follows that

$$\begin{aligned}\bar{\mathbf{L}} := \operatorname{diag}\{\bar{\mathbf{A}}\mathbf{1}\} - \bar{\mathbf{A}} = \operatorname{bdiag}\{\mathbf{L}_{\mathcal{V}}[1], \ldots, \mathbf{L}_{\mathcal{V}}[T]\} \\ + \operatorname{btridiag}\Big\{ \operatorname{diag}\{\mathbf{b}_{\mathcal{T}}[1]\}, \ldots, \operatorname{diag}\{\mathbf{b}_{\mathcal{T}}[T]\}; \\ - \mathbf{B}_{\mathcal{T}}[2], \ldots, -\mathbf{B}_{\mathcal{T}}[T]\Big\} \end{aligned} \tag{4.17}$$

where

$$\mathbf{b}_{\mathcal{T}}[t] := \begin{cases} \mathbf{B}_{\mathcal{T}}^T[2]\mathbf{1} & \text{if } t = 1 \\ (\mathbf{B}_{\mathcal{T}}^T[t+1] + \mathbf{B}_{\mathcal{T}}[t])\mathbf{1} & \text{if } 1 < t < T \\ \mathbf{B}_{\mathcal{T}}[T]\mathbf{1} & \text{if } t = T. \end{cases}$$

The rationale behind this graph extension is that, for $\bar{\mathbf{L}}$ as in (4.17) and diagonal $\{\mathbf{B}_{\mathcal{T}}[t]\}_{t=1}^T$, one can show that

$$\bar{\mathbf{f}}^T \bar{\mathbf{L}} \bar{\mathbf{f}} = \sum_{t=1}^T \mathbf{f}^T[t] \mathbf{L}_{\mathcal{V}}[t] \mathbf{f}[t] + \sum_{t=2}^T (\mathbf{f}[t] - \mathbf{f}[t-1])^T \mathbf{B}_{\mathcal{T}}[t] (\mathbf{f}[t] - \mathbf{f}[t-1]). \tag{4.18}$$

Clearly, the first and second sums on the right-hand side respectively penalize spatial and temporal variations. As a special case, if one sets $\mathbf{B}_{\mathcal{T}}[t] = b_{\mathcal{T}}\mathbf{I}\ \forall t$ for some $b_{\mathcal{T}} > 0$, the second sum becomes $b_{\mathcal{T}} \sum_{t=2}^T ||\mathbf{f}[t] - \mathbf{f}[t-1]||_2^2$, which promotes estimates with small changes over time.

Applying the notion of Laplacian kernels along the spatial dimension (see Sec. 2.1), but not along time, suggests generalizing (4.18) to obtain the regularizer

$$\bar{\mathbf{f}}^T \bar{\mathbf{K}}^\dagger \bar{\mathbf{f}} = \sum_{t=1}^T \mathbf{f}^T[t] \mathbf{K}_{\mathcal{V}}^\dagger[t] \mathbf{f}[t] + \sum_{t=2}^T (\mathbf{f}[t] - \mathbf{f}[t-1])^T \mathbf{B}_{\mathcal{T}}[t] (\mathbf{f}[t] - \mathbf{f}[t-1]) \tag{4.19}$$

where $\mathbf{K}_{\mathcal{V}}^\dagger[t] = r_t(\mathbf{L}_{\mathcal{V}}[t])$, $t = 1, \ldots, T$ for $\{r_t\}_{t=1}^T$ a collection of user-selected spectral maps

such as those in Table 2.1. In that case, (4.19) corresponds to the kernel matrix

$$\begin{aligned}\bar{\mathbf{K}}^{\dagger} = &\,\mathrm{bdiag}\left\{\mathbf{K}_{\mathcal{V}}^{\dagger}[1],\ldots,\mathbf{K}_{\mathcal{V}}^{\dagger}[T]\right\}\\ &+\mathrm{btridiag}\Big\{\,\mathrm{diag}\left\{\mathbf{b}_{\mathcal{T}}[1]\right\},\ldots,\mathrm{diag}\left\{\mathbf{b}_{\mathcal{T}}[T]\right\};\\ &-\mathbf{B}_{\mathcal{T}}[2],\ldots,-\mathbf{B}_{\mathcal{T}}[T]\Big\}. \end{aligned} \tag{4.20}$$

Although kernels of this form do not offer a frequency-domain control of reconstruction along time, they still enjoy the spatial flexibility of the kernels in Sec. 4.3.1.

Remark 5. *To guarantee that $\bar{\mathbf{K}}^{\dagger}$ in (4.20) qualifies for online implementation, it suffices to guarantee that $\bar{\mathbf{K}}$ is invertible since it is already block tridiagonal. This holds e.g. if $\mathbf{K}_{\mathcal{V}}[t]$ is invertible for all t.*

4.4 Simulated tests

This section compares the performance of the proposed schemes with state-of-the-art alternatives and illustrates some of the trade-offs inherent to time-varying function reconstruction through real-data experiments. Unless otherwise stated, the compared estimators include distributed least squares reconstruction (DLSR) [23] with step size μ_{DLSR} and parameter β_{DLSR}; the least mean-square (LMS) algorithm in [24] with step size μ_{LMS}; bandlimited instantaneous estimator (BL-IE), which results from applying [9,10,13] separately per t; KRR instantaneous estimator (KRR-IE) reconstruction in (4.2) with a diffusion kernel with parameter σ; and the proposed KKF (Algorithms 1 and 2) with kernel given by (4.20) for $\mathbf{B}_{\mathcal{T}}[T] = b_{\mathcal{T}}\boldsymbol{I}$ and $\mathbf{K}_{\mathcal{V}}[t]$ a diffusion kernel with parameter σ. DLSR, LMS, and BL-IE also use a bandwidth parameter B.

The first data set comprises hourly temperature measurements at $N = 109$ stations across the continental U.S. in 2010 [40]. Temperature reconstruction has been extensively employed in the literature to analyze the performance of inference tools over graphs (see e.g. [20,21,23,31]). A time-invariant graph was constructed following the approach in [21] with 7 nearest neighbors, which relies on geographical distances. Function $f_n[t]$ represents

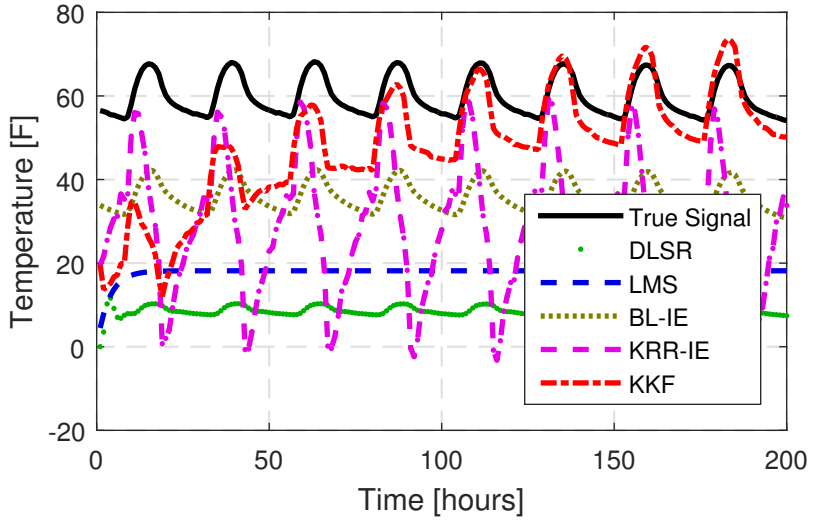


Figure 4.2: True temperature and estimates across time at a randomly picked unobserved station ($\mu = 10^{-7}$, $\sigma = 1.8$, $b_{\mathcal{T}} = 0.01$, $\mu_{\text{DLSR}} = 1.2$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 0.6$, $B = 2$).

the temperature at the n-th station and t-th sampling instant. In the first experiment, the latter corresponds to the t-th hour, whereas for the rest, it corresponds to the temperature at 12:00 PM of the t-th day.

Fig. 4.2 depicts the true temperature measured at an unobserved randomly picked station over the first 200 hours of 2010 along with its estimates for a typical realization of the time-invariant sampling set $\mathcal{S} = \mathcal{S}[t],\ \forall t$, drawn at random within all sampling sets with $S = 44$ elements. Different from instantaneous alternatives, whose error does not decrease with time, KKF is observed to successfully leverage time dynamics to track the temperature at the unobserved station. On the other hand, DLSR and LMS are unable to track the rapid variations of f since their design assumes slowly changing functions.

The next experiments compare the cumulative normalized mean-square error (NMSE), defined as

$$\text{NMSE}(t, \{\mathcal{S}[\tau]\}_{\tau=1}^{t}) := \frac{\sum_{\tau=1}^{t} ||\mathbf{S}^c[\tau](\mathbf{f}[\tau] - \hat{\mathbf{f}}[\tau|\tau])||_2^2}{\sum_{\tau=1}^{t} ||\mathbf{S}^c[\tau]\mathbf{f}[\tau]||_2^2}$$

where $\mathbf{S}^c[\tau]$ is an $N - S[\tau] \times N$ matrix comprising the rows of $\mathbf{I}_N$ whose indices are not in $\mathcal{S}[t]$.

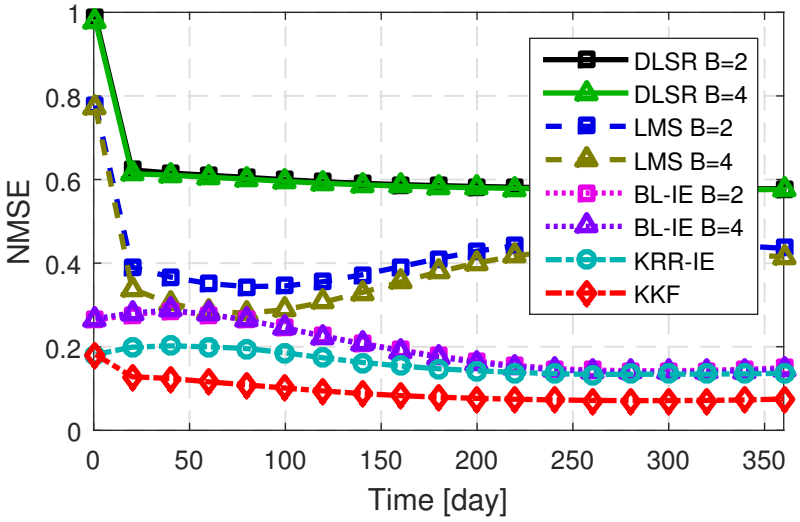


Figure 4.3: NMSE of daily temperature estimates over 2010. ($\mu = 10^{-7}$, $\sigma = 1.8$, $b_{\mathcal{T}} = 0.01$, $\mu_{\text{DLSR}} = 1.2$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 0.6$).

Fig. 4.3 shows the NMSE for $\mathcal{S}[t] = \mathcal{S}$, $\forall t$, averaged over all possible $\mathcal{S}$ with $S = 44$ elements. It is observed that the instantaneous estimators outperform DLSR and LMS, which can only cope with slow variations of f. Furthermore, the error of KKF is half the error of the nearest alternative, demonstrating the importance of exploiting time dynamics.

Fig. 4.4 shows the impact of the number of observed vertices S in NMSE$(T, \{\mathcal{S}[\tau]\}_{\tau=1}^{t})$, with $T = 365$ days, averaged over all sets $\mathcal{S}[\tau] = \mathcal{S}$ $\forall \tau$ with S elements. Observe that KKF consistently outperforms all alternatives. Still, the advantage of KKF over KRR-IE is more pronounced for small S, since in that case exploiting the time dynamics is more critical.

To illustrate the trade-off between reliance on temporal versus spatial information, the next experiment analyzes the effects of the scaling parameter $b_{\mathcal{T}}$ in the kernel adopted by KRR (cf. (4.20)). A large value of $b_{\mathcal{T}}$ leads to an estimator that relies more heavily on time dynamics and vice versa. Fig. 4.5 shows NMSE$(T, \{\mathcal{S}[\tau]\}_{\tau=1}^{t})$, with $T = 100$ days, averaged over all sets $\mathcal{S}[\tau] = \mathcal{S}$ $\forall \tau$ with $S = 44$ elements. The kernel in (4.20) is adopted with $\mathbf{K}_{\mathcal{V}}[t]$ being the regularized Laplacian (KKF-L) or diffusion kernels (KKF-DF) from Table 2.1, while $\mathbf{B}_{\mathcal{T}}[t] = b_{\mathcal{T}}\boldsymbol{I}$. It is observed that there exists an optimum value for $b_{\mathcal{T}}$ which leads to the best reconstruction performance. This corresponds to the optimal trade-off point between reliance on temporal and spatial information. The optimal NMSE is achieved by

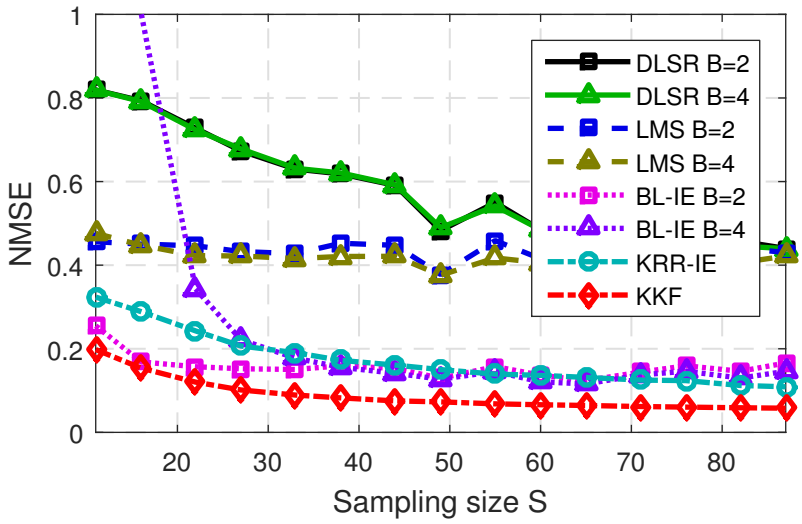


Figure 4.4: NMSE for increasing sampling size ($\mu = 10^{-7}$, $\sigma = 1.6$, $b_{\mathcal{T}} = 0.01$, $\mu_{\text{DLSR}} = 1.2$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 0.6$).

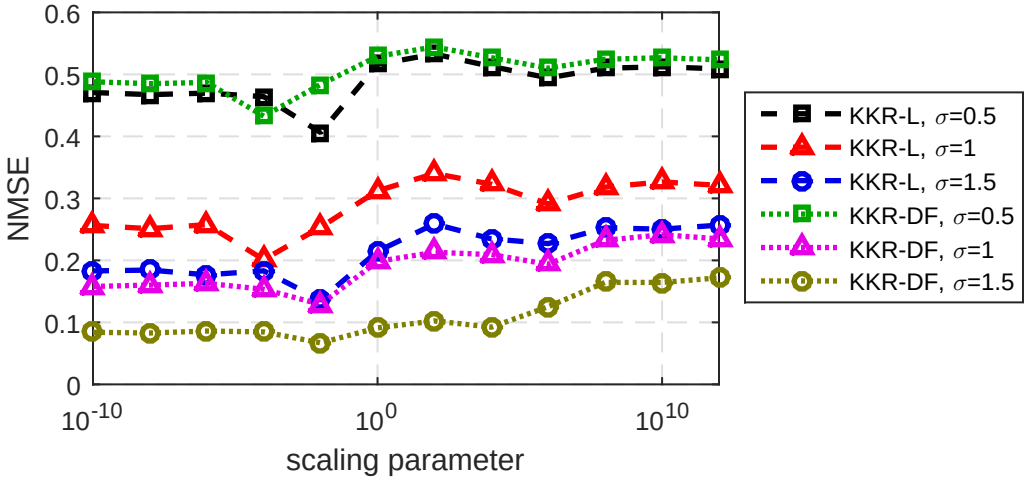


Figure 4.5: NMSE for different kernels vs. scale parameter $b_{\mathcal{T}}$ ($\mu = 10^{-7}$).

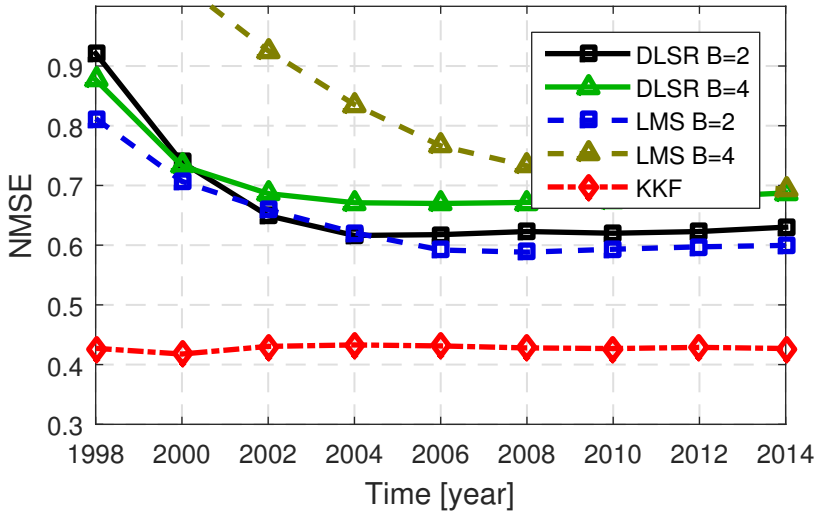


Figure 4.6: NMSE for the economic sectors data set ($\sigma = 5.2$, $\mu = 10^{-4}$, $b_{\mathcal{T}} = 0.01$, $\mu_{\text{DLSR}} = 1.2$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 0.6$).

a diffusion kernel with $\sigma = 1.5$ and $b_{\mathcal{T}} = 0.01$.

The second data set is provided by the Bureau of Economic Analysis of the U.S. Department of Commerce and contains the annual investments between each pair of sectors among $N = 61$ economic sectors in the interval 1997-2014 [50]. Each entry $A^{\mathcal{V}}_{n,n'}[t]$ of $\mathbf{A}_{\mathcal{V}}[t]$ contains the investment in trillions of dollars between sectors n and n' for the year $1995+2t$ with $t = 1, 2, \ldots, T$, where $T = 9$. DLSR and LMS adopt $\mathbf{A}_{\mathcal{V}} = (1/T)\sum_{\tau=1}^{T} \mathbf{A}_{\mathcal{V}}[\tau]$ since they cannot handle time-varying topologies. The value $f_n[t]$ corresponds to the total production of the n-th sector in year $1996 + 2t$, $t = 1, 2, \ldots, T$. The sampling interval was set to two years, so that disjoint subsets of years are used for generating the signal and constructing the graphs.

The next experiment demonstrates the ability of KKF to handle time-varying topologies. To this end, Fig. 4.6 plots NMSE$(t, \{\mathcal{S}[\tau]\}_{\tau=1}^{t})$, averaged over all sets $\mathcal{S}[t] = \mathcal{S}$, $\forall t$, with $S = 37$ elements. KKF utilizes the kernel in (4.20) with $\mathbf{K}_{\mathcal{V}}[t]$ a diffusion kernel constructed from $\mathbf{L}_{\mathcal{V}}[t]$ per t and $\mathbf{B}_{\mathcal{T}}[t] = b_{\mathcal{T}}\boldsymbol{I}$, $\forall t$. Again, Fig. 4.6 showcases the superior performance of the proposed KKF, whose error is significantly less than the error of competing alternatives.

The third data set is obtained from an epilepsy study [8]. Diagnosis of epilepsy is heavily based on analysis of ECoG data; see Sec. 1.

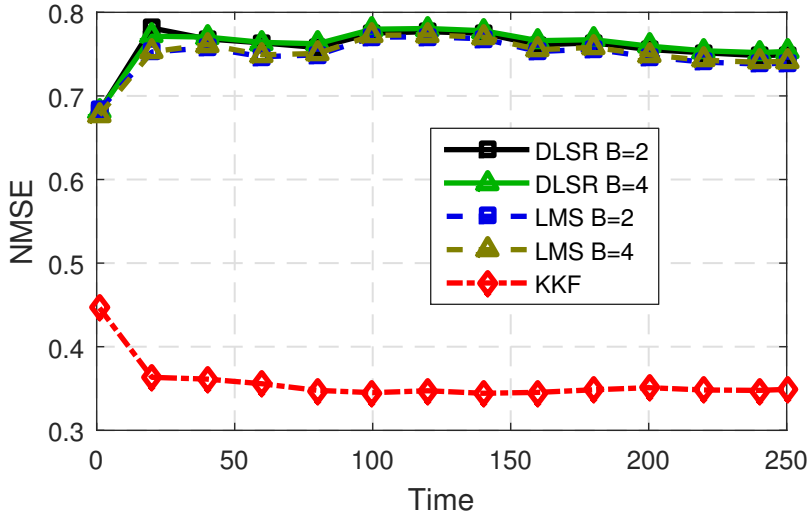


Figure 4.7: NMSE for the ECoG data set ($\sigma = 1.2$, $\mu = 10^{-4}$, $\mu_{\text{DLSR}} = 1.2$, $b_{\mathcal{T}} = 0.01$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 0.6$).

The next experiments utilize the ECoG time series obtained in [8] from $N = 76$ electrodes implanted in a patient's brain before and after the onset of a seizure.

A symmetric time-invariant adjacency matrix $\mathbf{A}_{\mathcal{V}}$ was obtained using the method in [51] with ECoG data before the onset of the seizure. Function $f_n[t]$ comprises the electrical signal at the n-th electrode and t-th sampling instant after the onset of the seizure, for a period of $T = 250$ samples. The values of $f_n[t]$ were normalized by subtracting the temporal mean of each time series before the onset of the seizure.

The goal of the experiment is to illustrate the reconstruction performance of the proposed KKF in capturing the complex spatio-temporal dynamics of brain signals.

Fig. 4.7 depicts the NMSE$(t, \{\mathcal{S}[\tau]\}_{\tau=1}^{t})$, averaged over all sets $\mathcal{S}[t] = \mathcal{S}$, $\forall t$, of size $S = 53$. For the proposed KKF, a space-time kernel was created using (4.20) with a time-invariant diffusion kernel $\mathbf{K}_{\mathcal{V}}$ generated from $\mathbf{L}_{\mathcal{V}}$, and a time-invariant $\mathbf{B}_{\mathcal{T}} = b_{\mathcal{T}}\boldsymbol{I}$. Fig. 4.7 showcases the superior reconstruction performance of the KKF among competing approaches, even with a small number of samples. This result suggests that the ECoG diagnosis technique could be efficiently conducted even with a smaller number of intracranial electrodes, which may have a great impact on the patient's experience.

Chapter 5

Reconstruction of Dynamic Functions on Dynamic Graphs using Kernel Kriged Kalman Filters

This chapter contributes a graph-aware kernel-based kriged Kalman filtering (KrKF) approach that leverages the spatio-temporal dynamics to allow for efficient online reconstruction, while also coping with dynamically evolving network topologies. Laplacian kernels are employed to promote smoothness of the estimates over the graph when second-order statistics are unknown, as is often the case. First, an encompassing model is proposed, that captures the spatio-temporal dynamics of dynamic graph functions over dynamic graphs, without any assumption on their statistics. Based on the adopted model a kernel ridge regression (KRR) problem is formulated. Next, the kernel KrKF (KKrKF) is derived, that solves the KRR optimization problem in an online fashion. The novel deterministic estimator is capable of promoting smoothness over time and space through judicious use of *Laplacian kernel* functions, that encode similarities between the nodes of the graph. Finally, to cope with the challenging problem of kernel selection this chapter develops a novel

data-driven approach, to choose the appropriate similarity measure. The time-varying setting of the problem calls for an online multi-kernel learning (MKL) approach that adapts to the observed data on-the-fly. The proposed MKL algorithm exploits the structure of Laplacian kernels to achieve a complexity linear in the number of nodes in the network, that renders it attractive for big data applications. Combining the KKrKF with the MKL algorithm provides a novel filtering approach that adapts to the observed data, selects the optimal kernel, and improves its future predictions. The computational complexity of the overall algorithm is linear in the number of time samples, and scales favorably for online data applications.

5.1 Preliminaries

A time-varying graph is a tuple $\mathcal{G}[t] := (\mathcal{V}, \mathbf{A}[t])$ $t = 1, 2, \ldots$, where $\mathcal{V} := \{v_1, \ldots, v_N\}$ denotes the vertex set and $\mathbf{A}[t]$ the $N \times N$ adjacency matrix, whose (n, n')-th entry $A_{n,n'}[t]$, is the nonnegative edge weight connecting vertices v_n and $v_{n'}$ at time t. The graphs in this chapter are undirected and have no self-loops, which respectively imply that $\mathbf{A}[t] = \mathbf{A}^T[t]$ and $A_{n,n}[t] = 0$, $\forall t, n$. The Laplacian matrix is defined as $\mathbf{L}[t] := \text{diag}\{\mathbf{A}[t]\mathbf{1}_N\} - \mathbf{A}[t]$, and is known to be positive semidefinite [5].

A time-varying graph function (or signal) is a map $f : \mathcal{V} \times \mathcal{T} \to \mathbb{R}$, where $\mathcal{T} = \{1, 2, \ldots\}$ is the set of time indices. Specifically, $f_n[t] := f(v_n, t)$ represents an attribute value at node n and time t, e.g. the closing price of the n-th stock on the t-th day. Vector $\mathbf{f}[t] := [f_1[t], \ldots, f_N[t]]^T \in \mathbb{R}^N$ collects the function values at time t. At time t, $f_n[t]$ is observed at a subset of $S[t]$ nodes $\mathcal{S}[t] \subset \mathcal{V}$. The observations $\mathbf{y}[t] \in \mathbb{R}^{S[t]}$ can be compactly arranged as

$$\mathbf{y}[t] = \mathbf{S}[t]\mathbf{f}[t] + \mathbf{e}[t], \quad t = 1, 2, \ldots \tag{5.1}$$

where $\mathbf{S}[t] \in \{0, 1\}^{S[t] \times N}$ selects the rows of $\mathbf{f}[t]$ with indices in $\mathcal{S}[t]$, and $\mathbf{e}[t] \in \mathbb{R}^{S[t]}$ represents the observation error.

Per slot t, $\mathbf{f}[t]$ will be modeled as the superposition

$$\mathbf{f}[t] = \mathbf{f}_\chi[t] + \mathbf{f}_\nu[t] \tag{5.2}$$

where $\{\mathbf{f}_\nu[t]\}_t$ capture only spatial dependencies, while $\{\mathbf{f}_\chi[t]\}_t$ accounts for spatio-temporal dynamics obeying the state equation

$$\mathbf{f}_\chi[t] = \mathbf{P}[t]\mathbf{f}_\chi[t-1] + \boldsymbol{\eta}[t], \quad t = 1, 2, \ldots \tag{5.3}$$

where $\mathbf{P}[t]$ is an $N \times N$ transition matrix, and $\mathbf{f}_\chi[0] = \mathbf{0}$, and $\boldsymbol{\eta}[t] \in \mathbb{R}^N$ captures the state error. The model in (5.3) is widely used and offers flexibility in tracking multiple forms of temporal dynamics [52, Ch. 3].

The goal of this chapter is to reconstruct $\mathbf{f}[t]$ online, given $\mathbf{y}[\tau]$, $\mathbf{S}[\tau]$, $\mathbf{P}[\tau]$ and $\mathbf{A}[\tau]$ for $\tau = 1, \ldots, t$. Based on the proposed model, (5.1)-(5.3), an online estimator is derived in Sec. 5.3 that obviates the need for assumptions on data distributions or knowledge of second-order statistics.

Remark 6. *In the field of geostatistics,* $\mathbf{f}_\nu[t]$ *models the so-termed small-scale* spatial fluctuations*, while* $\mathbf{f}_\chi[t]$ *corresponds to the so-called* trend*. The decomposition (5.2) is often dictated by the sampling interval: whereas* $\mathbf{f}_\chi[t]$ *captures slow dynamics relative to the sampling interval, fast variations are modeled with* $\mathbf{f}_\nu[t]$*. Examples motivating (5.2) include network delay prediction [25], where* $\mathbf{f}_\chi[t]$ *represents the queuing delay while* $\mathbf{f}_\nu[t]$ *the propagation, transmission, and processing delays. Likewise, when predicting prices across different stocks,* $\mathbf{f}_\chi[t]$ *captures the daily evolution of the stock market, which is correlated across stocks and time samples, while* $\mathbf{f}_\nu[t]$ *describes unexpected changes, such as the daily drop of the stock market due to political statements, which are considered uncorrelated over time.*

Remark 7. *The state transition matrix* $\mathbf{P}[t]$ *can be selected in accordance with the prior information available. Simplicity in estimation motivates the random walk model, where* $\mathbf{P}[t] = \alpha \boldsymbol{I}_N$ *with* $\alpha > 0$*. On the other hand, adherence to the graph, prompts the selection* $\mathbf{P}[t] = \alpha(\mathbf{A}[t-1] + \mathbf{I}_N)$*, in which case (5.3) amounts to a graph-constrained vector autoregressive model; see e.g. [53].*

5.2 Background

This section familiarizes the reader with the traditional KrKF and Laplacian kernels. Specifically, Sec. 5.2.1 reviews the KrKF for general functions that is derived using the statistics of the signals and LMMSE optimality criteria. Sec. 5.2.2 reviews the RKHS's of graph functions and establishes that KRR in [19, 33] generalizes clairvoyant kriging in [54].

5.2.1 Kriged Kalman Filter

The KrKF algorithm was first introduced for prediction of processes evolving over continuous fields, as typically occurs in geostatistics [55]. This section reviews the KrKF for general processes $\mathbf{f}[t]$ that do not need to evolve over a graph.

The KrKF algorithm adopts the model described by (5.1)-(5.3), but furthermore assumes knowledge of the statistics of the time-varying processes. Specifically, $\mathbf{f}_\nu[t]$ is assumed zero mean, and has covariance matrix $\mathrm{E}[\mathbf{f}_\nu[t]\mathbf{f}_\nu{}^T[\tau]] = \mathbf{\Sigma}_\nu[t]\delta[t-\tau]$. Moreover, $\mathbf{e}[t]$ and $\boldsymbol{\eta}[t]$ are assumed uncorrelated, meaning that $\mathrm{E}[\mathbf{e}[t]\boldsymbol{\eta}^T[\tau]] = \mathbf{0}_{S[t],N}$, and also uncorrelated with $\mathbf{f}_\nu[t]$, i.e., $\mathrm{E}[\mathbf{e}[t]\mathbf{f}_\nu{}^T[\tau]] = \mathbf{0}_{S[t],N}$, $\mathrm{E}[\boldsymbol{\eta}[t]\mathbf{f}_\nu{}^T[\tau]] = \mathbf{0}_N \ \forall t,\tau$. Next, $\mathbf{e}[t]$ is postulated zero mean $\mathrm{E}[\mathbf{e}[t]] = \mathbf{0}$, and uncorrelated over time and space, meaning that $\mathrm{E}[\mathbf{e}[t]\mathbf{e}^T[\tau]] = \sigma_e^2\boldsymbol{I}_{S[t]}$, if $t=\tau$, and $\mathbf{0}_{S[t],S[\tau]}$ otherwise. Finally, for $\boldsymbol{\eta}[t]$ is assumed that $\mathrm{E}[\boldsymbol{\eta}[t]] = \mathbf{0}$ and $\mathrm{E}[\boldsymbol{\eta}[t]\boldsymbol{\eta}^T[\tau]] = \mathbf{\Sigma}_\eta[t]\delta[t-\tau]$.

The estimation of $\mathbf{f}[t]$ is performed in two steps. In the first step, an estimate $\hat{\mathbf{f}}_\chi[t|t]$ is obtained from the measurements $\{\mathbf{y}[\tau]\}_{\tau=1}^t$ using the traditional Kalman filter (KF) [52, Ch. 3] with the unknown $\mathbf{f}_\nu[t]$ lumped in the observation noise. In the second step, $\mathbf{f}_\nu[t]$ is estimated through the kriging predictor [54], which is given by the LMMSE estimator

$$\begin{aligned}\hat{\mathbf{f}}_\nu[t|t] =& \mathrm{E}\left\{\mathbf{f}_\nu[t]\middle|\mathbf{f}_\chi[t]=\hat{\mathbf{f}}_\chi[t|t],\mathbf{y}[t]\right\}\\ =&\mathbf{\Sigma}_\nu[t]\mathbf{S}^T[t](\mathbf{S}[t]\mathbf{\Sigma}_\nu[t]\mathbf{S}^T[t]+\sigma_e^2\boldsymbol{I}_{S[t]})^{-1}\boldsymbol{\psi}[t]\end{aligned} \tag{5.4}$$

where $\boldsymbol{\psi}[t] := \mathbf{y}[t]-\mathbf{S}[t]\hat{\mathbf{f}}_\chi[t|t]$. Finally, combining the component estimates yields [cf. (5.2)]

$$\hat{\mathbf{f}}[t|t] = \hat{\mathbf{f}}_\chi[t|t] + \hat{\mathbf{f}}_\nu[t|t]. \tag{5.5}$$

A challenge associated with traditional KrKF is that KF algorithm as well as the kriging predictor (5.4) require knowledge of the statistics of the time-evolving functions. The next subsection reviews graph Laplacian kernels, that will be employed to cope with uncertainty about the second order statistics of the functions of interest.

5.2.2 Reproducing Kernel Hilbert Spaces on Graphs

Kernel based methods assume that the function of interest $h : \mathcal{V} \to \mathbb{R}$, that takes values over the vertices of $\mathcal{G} := (\mathcal{V}, \mathbf{A})$, belongs to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ that is expressed as

$$\mathcal{H} := \left\{ h : h(v) = \sum_{n=1}^{N} \alpha_n \kappa(v, v_n),\ \alpha_n \in \mathbb{R} \right\}. \tag{5.6}$$

The function $\kappa : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is termed *kernel map* and encodes some notion of similarity between the values $h(v_n)$ and $h(v_{n'})$. Interestingly, a vertex v_n for functions over graphs can be thought of as a feature vector $\mathbf{x}_n \in \mathbb{R}^D$ for functions over continuous fields $g : \mathbb{R}^D \to \mathbb{R}$ [56]. The matrix defined by the kernel map, $\mathbf{K}$, with entries $K_{n,n'} = \kappa(v_n, v_{n'})$ is an $N \times N$ positive definite matrix and is termed as *Laplacian kernel* [3,4,19,33]. After defining $\mathbf{h} := [h(v_1), h(v_2), \ldots, h(v_N)]$ it follows [cf. (5.6)]

$$\mathbf{h} = \mathbf{K}\boldsymbol{\alpha} \tag{5.7}$$

where $\boldsymbol{\alpha} := [\alpha_1, \alpha_2, \ldots, \alpha_N]$.

Laplacian kernels have been widely used [3,4,19,28] to promote *smoothness* with respect to the underlying graph topology, by penalizing functions that exhibit pronounced variations among neighboring vertices. For a given Laplacian matrix with eigendecomposition $\mathbf{L} = \mathbf{U}\,\mathrm{diag}\{\boldsymbol{\lambda}\}\mathbf{U}^T$, a family of Laplacian kernels is defined as [4]

$$\mathbf{K} := r^{-1}(\mathbf{L}) := \mathbf{U}\,\mathrm{diag}\{r^{-1}(\boldsymbol{\lambda})\}\mathbf{U}^T \tag{5.8}$$

where $r:\mathbb{R}\to\mathbb{R}_+$ is chosen to be a non-decreasing function. Table 2.1 summarizes common choices of $r(\cdot)$ which can be selected to promote a certain structure in the so-called graph Fourier transform of $\mathbf{h}$ [4, 5, 19].

A common complexity measure of the functions in $\mathcal{H}$ is the RKHS norm defined by

$$\|h\|_{\mathcal{H}}^2 := \langle h, h\rangle_{\mathcal{H}} = \boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} \tag{5.9a}$$

$$\|h\|_{\mathcal{H}}^2 = \boldsymbol{\alpha}^T\mathbf{K}\mathbf{K}^{-1}\mathbf{K}\boldsymbol{\alpha} = \mathbf{h}^T\mathbf{K}^{-1}\mathbf{h} = \|\mathbf{h}\|_{\mathbf{K}}^2 \tag{5.9b}$$

that is used as a regularizer to control overfitting. If one chooses as $r(\cdot)$ the regularized Laplacian kernel function with $\sigma = 1$ (cf. Table 2.1) then $\mathbf{K} = (\mathbf{L} + \mathbf{I})^{-1}$ and the RKHS norm (5.9b) takes the following form

$$\begin{aligned}\|\mathbf{h}\|_{\mathbf{K}}^2 &= \mathbf{h}^T\mathbf{L}\mathbf{h} + \|\mathbf{h}\|_2^2 \\ &= \tfrac{1}{2}\sum_{n'=1}^{N}\sum_{n=1}^{N} A_{n,n'}(h_n - h_{n'})^2 + \|\mathbf{h}\|_2^2\end{aligned} \tag{5.10}$$

Clearly, one can use (5.10) in order to promote functions with small variations among connected vertices. In general, prior information can steer the selection of an appropriate kernel, for example if the graph function is influenced only by the functions values in the p-hop neighborhood the p-step random walk kernel is preferable.

Since vectors $\mathbf{f}_\nu[t], \boldsymbol{\eta}[t]$ are functions over the graph, we will assume that

$$\begin{aligned}&\mathbf{f}_\nu[t] \in \mathcal{H}_\nu[t] \\ &\boldsymbol{\eta}[t] \in \mathcal{H}_\eta[t] \qquad t = 1, 2, \ldots\end{aligned} \tag{5.11}$$

where $\mathcal{H}_\nu[t], \mathcal{H}_\eta[t]$ are associated with the Laplacian kernels $\mathbf{K}_\nu[t], \mathbf{K}_\eta[t]$ respectively. The kernels are obtained from (5.8) using the given $\mathbf{L}[t]$ and suitable weighting functions $r_\nu(\cdot), r_\eta(\cdot)$. Laplacian kernels (5.8) span a rich family of kernels, that includes smoothness promoting kernels as well as others that not necessary promote smoothness as $s\mathbf{I}$. To cope with lack of prior information about the appropriate $\mathbf{K}_\nu[t]$ and $\mathbf{K}_\eta[t]$, a multi-kernel learning approach

will be developed in Sec. 5.4.

Next, it will be shown that kriging is a special case of KRR. Towards this goal, consider the single time instant formulation -drop the argument $[t]$- where $\boldsymbol{\psi} = \mathbf{S}\mathbf{f}_\nu + \mathbf{e}$, with $\boldsymbol{\psi} := \mathbf{y} - \mathbf{S}\mathbf{f}_\chi$. Kernel ridge regression seeks an estimate of a graph function $\mathbf{f}_\nu$ given the observations $\boldsymbol{\psi}$. The KRR estimate of $\mathbf{f}_\nu$ is given by [19]

$$\begin{aligned}\hat{\mathbf{f}}_\nu &= \arg\min_{\mathbf{f}_\nu} \frac{1}{S}\|\boldsymbol{\psi} - \mathbf{S}\mathbf{f}_\nu\|_2^2 + \mu\|\mathbf{f}_\nu\|_{\mathbf{K}_\nu}^2 \\ &= \mathbf{K}_\nu\mathbf{S}^T(\mathbf{S}\mathbf{K}_\nu\mathbf{S}^T + \mu S\mathbf{I}_S)^{-1}\boldsymbol{\psi} \end{aligned} \tag{5.12}$$

where μ is a user-selected regularization parameter. Notice that the KRR estimator (5.12) reduces to the kriging predictor (5.4) if $\mu S = \sigma_e^2$ and $\boldsymbol{\Sigma}_\nu = \mathbf{K}_\nu$. As a result, (5.12) generalizes (5.4) in the sense that $\mathbf{f}_\nu$ can be deterministic, so long as it belongs to a reproducing kernel Hilbert space generated by the prescribed $\mathbf{K}_\nu$. Rather than minimizing the LMMSE criterion, the resulting KRR can account for the underlying graph through a judicious selection of $\mathbf{K}_\nu$. Next, the KRR framework is applied to our time-varying problem so that the deterministic KrKF is derived.

5.3 Kernel Kriged Kalman Filter

Naturally, one can think of extending the KrKF whose derivation follows a probabilistic framework [25, 55, 57, 58] to the Kernel KrKF (KKrKF) an online algorithm that is derived based on a kernel-based learning framework. The resulting estimator does not make any assumptions on the statistics of the functions of interest. Formulating the generalized kernel

ridge regression that emerges from (5.1), (5.2), (5.3), (5.11) it follows

$$\begin{aligned}\underset{\{\mathbf{f}_\chi[\tau],\mathbf{f}_\nu[\tau]\}_{\tau=1}^t}{\arg\min} & \sum_{\tau=1}^{t} \tfrac{1}{S[\tau]}\|\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau]-\mathbf{S}[\tau]\mathbf{f}_\nu[\tau]\|^2 \\ & +\mu_1\sum_{\tau=1}^{t}\|\mathbf{f}_\chi[\tau]-\ \mathbf{P}[\tau]\mathbf{f}_\chi[\tau-1]\|^2_{\mathbf{K}_\eta[\tau]} \\ & +\mu_2\sum_{\tau=1}^{t}\|\mathbf{f}_\nu[\tau]\|^2_{\mathbf{K}_\nu[\tau]} \end{aligned} \tag{5.13}$$

where $\mu_1, \mu_2 \geq 0$ are regularization parameters controlling the effect of RKHS norms of $\{\mathbf{f}_\nu[\tau], \boldsymbol{\eta}[\tau]\}_{\tau=1}^t$, which are employed to cope with the under-determined least squares problem, that arises since $S[t] \leq N$ for $t = 1, 2, \ldots$. The optimization problem in (5.13) is a batch problem, at each t gives estimates for $\{\mathbf{f}_\chi[\tau], \mathbf{f}_\nu[\tau]\}_{\tau=1}^t$. The next theorem establishes that (5.13) can be solved for the online estimates $\{\hat{\mathbf{f}}_\nu[\tau|\tau], \hat{\mathbf{f}}_\chi[\tau|\tau]\}_{\tau=1}^t$, by an online algorithm.

Theorem 2. *The solution to (5.13) follows by an online algorithm, the kernel kriged Kalman filter Algorithm 3, for the online estimates* $\{\hat{\mathbf{f}}_\nu[\tau|\tau], \hat{\mathbf{f}}_\chi[\tau|\tau]\}_{\tau=1}^t$.

Proof. See Appendix B.1. □

One iteration of the proposed KKrKF is summarized as Algorithm 3. This online estimator with computational complexity $\mathcal{O}(N^3)$ per t, tracks the temporal variations of the signal of interest through (5.3), and promotes desired properties such as smoothness over the graph, using $\mathbf{K}_\nu[t]$. Different from existing approaches, our KKrKF takes into account the underlying graph structure in estimating $\mathbf{f}_\nu[t]$ as well as $\mathbf{f}_\chi[t]$. Furthermore, by using $\mathbf{L}[t]$ in (5.8), it can also accommodate dynamic graph topologies. Finally, it should be noted that other derivations of KrKF [25, 55, 57, 58] are based on probabilistic arguments, whereas KKrKF is derived using kernel ridge regression.

Lack of prior information necessitates the development of data-driven approaches that efficiently learn the appropriate kernel matrix. Towards this end a novel online multi-kernel learning approach will be developed in the next section.

Algorithm 3 Kernel Kriged Kalman filter (KKrKF)

Input: $\mathbf{K}_\eta[t]; \mathbf{K}_\nu[t] \in \mathbb{S}_+^N$; $\mathbf{P}[t] \in \mathbb{R}^{N\times N}$; $\mathbf{y}[t] \in \mathbb{R}^{S[t]}$; $\mathbf{S}[t] \in \{0,1\}^{S[t]\times N}$.

$\bar{\mathbf{K}}_\chi[t] = \frac{1}{\mu_2}\mathbf{S}[t]\mathbf{K}_\nu[t]\mathbf{S}^T[t] + S[t]\mathbf{I}_{S[t]}$
$\hat{\mathbf{f}}_\chi[t|t-1] = \mathbf{P}[t]\hat{\mathbf{f}}_\chi[t-1|t-1]$
$\mathbf{M}[t|t-1] = \mathbf{P}[t]\mathbf{M}[t-1|t-1]\mathbf{P}^T[t] + \frac{1}{\mu_1}\mathbf{K}_\eta[t]$
$\mathbf{G}[t] = \mathbf{M}[t|t-1]\mathbf{S}^T[t](\bar{\mathbf{K}}_\chi[t] + \mathbf{S}[t]\mathbf{M}[t|t-1]\mathbf{S}^T[t])^{-1}$
$\mathbf{M}[t|t] = (\mathbf{I} - \mathbf{G}[t]\mathbf{S}[t])\mathbf{M}[t|t-1]$

$\hat{\mathbf{f}}_\chi[t|t] = \hat{\mathbf{f}}_\chi[t|t-1] + \mathbf{G}[t](\mathbf{y}[t] - \mathbf{S}[t]\hat{\mathbf{f}}_\chi[t|t-1])$

$\hat{\mathbf{f}}_\nu[t|t] = \mathbf{K}_\nu[t]\mathbf{S}^T[t]{\bar{\mathbf{K}}_\chi}^{-1}[t](\mathbf{y}[t] - \mathbf{S}[t]\hat{\mathbf{f}}_\chi[t|t])$

Output: $\hat{\mathbf{f}}_\chi[t|t]$; $\hat{\mathbf{f}}_\nu[t|t]$; $\mathbf{M}[t|t]$.

5.4 Online Multi-kernel learning

This section proposes an online MKL algorithm tailored for the optimal selection of suitable kernels from a pool of kernels on-the-fly. Specifically, Sec. 5.4.1 formulates the MKL objective that has to be solved per t, Sec. 5.4.2 introduces an efficient solver for the subproblem of *kernel matching* , that exploits the common eigenspace of Laplacian kernel, and Sec. 5.4.3 presents the desired online algorithm.

5.4.1 Multi-kernel Learning

Choosing the appropriate kernel is an application-dependent art, and affects significantly the performance of the inference algorithms [19]. Data-driven techniques for selecting the pertinent kernel are known as multi-kernel learning (MKL) algorithms [27]. Most of the existing MKL approaches on graphs focus in inference of binary-valued functions [59, 60]. MKL algorithms for real-valued graph function inference have been recently proposed [19], but solve a batch learning problem, which is not appropriate when data arrive sequentially. Online MKL approaches [61, 62] are typically employed when the optimal kernel changes over time, and usually solve a challenging non-convex problem at a high computational complexity, but do not consider Laplacian kernels. To overcome limitations of previous

approaches, the present chapter introduces an online MKL technique for inference of graph functions, that dynamically explores a pool of multiple kernels and exploits the structure of the Laplacian kernels to provide efficient algorithms.

Since the suitable $r(\cdot)$ is not known a priori a linear combination of kernels will be employed $\mathbf{K} = \sum_m^M \theta^{(m)}\mathbf{K}^{(m)}$, where $\{\mathbf{K}^{(m)}\}_{m=1}^M$ is given and $\theta^{(m)} \geq 0\ \forall m$ are unknown. For the following assume that $\mathbf{K}_\nu[\tau] = \mathbf{K}_\nu$, $\mathbf{K}_\eta[\tau] = \mathbf{K}_\eta$ and $\mathbf{S}[\tau] = \mathbf{S}$, $\forall \tau$. Specifically, assume that $\mathbf{K}_\nu = \mathbf{K}_\nu(\boldsymbol{\theta}_\nu) = \sum_m^M \theta_\nu{}^{(m)}\mathbf{K}_\nu^{(m)}$ and $\mathbf{K}_\eta = \mathbf{K}_\eta(\boldsymbol{\theta}_\eta) = \sum_m^M \theta_\eta{}^{(m)}\mathbf{K}_\eta^{(m)}$, where $\boldsymbol{\theta}_\eta, \boldsymbol{\theta}_\nu \geq \mathbf{0}$. Next, (5.13) will be reformulated as

$$\begin{aligned}\underset{\substack{\{\mathbf{f}_\chi[\tau],\mathbf{f}_\nu[\tau]\}_{\tau=1}^t,\\ \boldsymbol{\theta}_\nu\geq\mathbf{0},\boldsymbol{\theta}_\eta\geq\mathbf{0}}}{\arg\min} & \sum_{\tau=1}^t \tfrac{1}{S}\|\mathbf{y}[\tau] - \mathbf{S}\mathbf{f}_\chi[\tau] - \mathbf{S}\mathbf{f}_\nu[\tau]\|^2 \\ &+\mu_1 \sum_{\tau=1}^t \|\mathbf{f}_\chi[\tau] - \ \mathbf{P}[\tau]\mathbf{f}_\chi[\tau-1]\|^2_{\mathbf{K}_\eta(\boldsymbol{\theta}_\eta)} \\ &+\mu_2 \sum_{\tau=1}^t \|\mathbf{f}_\nu[\tau]\|^2_{\mathbf{K}_\nu(\boldsymbol{\theta}_\nu)} + \rho_\nu\|\boldsymbol{\theta}_\nu\|_2^2 + \rho_\eta\|\boldsymbol{\theta}_\eta\|_2^2 \end{aligned} \tag{5.14}$$

where $\rho_\nu, \rho_\eta \geq 0$ are regularization parameters. The optimization problem in (5.14) is not jointly convex in $\{\mathbf{f}_\chi[\tau], \mathbf{f}_\nu[\tau]\}_{\tau=1}^t, \boldsymbol{\theta}_\nu, \boldsymbol{\theta}_\eta$, but it is separately convex in these variables. To solve (5.14) alternating minimization (AM) strategies will be employed, that suggest optimizing with respect to one variable, while keeping the other variables fixed [63].

Interestingly, if $\boldsymbol{\theta}_\nu, \boldsymbol{\theta}_\eta$ are considered fixed (5.13) is recovered, which is solved by Algorithm 3 for the estimates $\{\hat{\mathbf{f}}_\nu[\tau|\tau], \hat{\mathbf{f}}_\chi[\tau|\tau]\}_{\tau=1}^t$ in an online fashion. The following theorem derives the optimization problem to be solved for obtaining time-varying estimates for $\boldsymbol{\theta}_\nu, \boldsymbol{\theta}_\eta$.

Theorem 3. *Let $\hat{\mathbf{f}}_\chi[\tau|\tau]$, $\hat{\mathbf{f}}_\nu[\tau|\tau]$, for $\tau = 1 \ldots t$ be the estimates obtained from Algorithm 3 and let $\hat{\boldsymbol{\eta}}[\tau|\tau] := \hat{\mathbf{f}}_\chi[\tau|\tau] - \mathbf{P}[\tau]\hat{\mathbf{f}}_\chi[\tau-1|\tau-1]$, $\tau = 2 \ldots t$, and the matrices $\mathbf{R}_\nu[t] = \sum_{\tau=1}^t \hat{\mathbf{f}}_\nu[\tau|\tau]\hat{\mathbf{f}}_\nu^T[\tau|\tau]$ and $\mathbf{R}_\eta[t] = \sum_{\tau=1}^t \hat{\boldsymbol{\eta}}[\tau|\tau]\hat{\boldsymbol{\eta}}^T[\tau|\tau]$. The following convex op-*

timization objectives result after applying alternating minimization to (5.14)

$$\hat{\boldsymbol{\theta}}_\nu[t] = \underset{\boldsymbol{\theta}_\nu \geq \mathbf{0}}{\arg\min}\, \mathrm{Tr}\left(\mathbf{R}_\nu[t]\mathbf{K}_\nu^{-1}(\boldsymbol{\theta}_\nu)\right) + \tfrac{\rho_\nu}{\mu_2}\|\boldsymbol{\theta}_\nu\|_2^2 \tag{5.15a}$$

$$\hat{\boldsymbol{\theta}}_\eta[t] = \underset{\boldsymbol{\theta}_\eta \geq \mathbf{0}}{\arg\min}\, \mathrm{Tr}\left(\mathbf{R}_\eta[t]\mathbf{K}_\eta^{-1}(\boldsymbol{\theta}_\eta)\right) + \tfrac{\rho_\eta}{\mu_1}\|\boldsymbol{\theta}_\eta\|_2^2 \tag{5.15b}$$

Proof. First consider (5.15a). After keeping the terms in (5.14) corresponding to $\boldsymbol{\theta}_\nu$ and replacing $\{\mathbf{f}_\nu[\tau]\}_{\tau=1}^t$ with the estimates obtained from Algorithm 3 the objective reduces to $\sum_{\tau=1}^t \hat{\mathbf{f}}_\nu^T[\tau|\tau]\mathbf{K}_\nu^{-1}(\boldsymbol{\theta}_\nu)\hat{\mathbf{f}}_\nu[\tau|\tau] + \frac{\rho_\nu}{\mu_2}\|\boldsymbol{\theta}_\nu\|_2^2$. Next, using the linearity and cyclic invariance of the trace it follows $\mathrm{Tr}(\sum_{\tau=1}^t \hat{\mathbf{f}}_\nu^T[\tau|\tau]\mathbf{K}_\nu^{-1}(\boldsymbol{\theta}_\nu)\hat{\mathbf{f}}_\nu[\tau|\tau]) = \mathrm{Tr}(\sum_{\tau=1}^t \hat{\mathbf{f}}_\nu[\tau|\tau]\hat{\mathbf{f}}_\nu^T[\tau|\tau]\mathbf{K}_\nu^{-1}(\boldsymbol{\theta}_\nu)) = \mathrm{Tr}\left(\mathbf{R}_\nu[t]\mathbf{K}_\nu^{-1}(\boldsymbol{\theta}_\nu)\right)$, which concludes the proof. The derivation for (5.15b) follows along the same lines. □

To accommodate non-stationary environments instead of using $\mathbf{R}_\nu[t]$ and $\mathbf{R}_\eta[t]$ the following matrices will be considered

$$\tilde{\mathbf{R}}_\nu[t] = \sum_{\tau=1}^t \gamma_\nu^{t-\tau}\hat{\mathbf{f}}_\nu[\tau|\tau]\hat{\mathbf{f}}_\nu^T[\tau|\tau] + \gamma_\nu^t\mathbf{I} \tag{5.16a}$$

$$\tilde{\mathbf{R}}_\eta[t] = \sum_{\tau=1}^t \gamma_\eta^{t-\tau}\hat{\boldsymbol{\eta}}[\tau|\tau]\hat{\boldsymbol{\eta}}^T[\tau|\tau] + \gamma_\eta^t\mathbf{I} \tag{5.16b}$$

where the forgetting factors $\gamma_\eta, \gamma_\nu \in (0,1)$, control the stability of matrices $\tilde{\mathbf{R}}_\nu[t], \tilde{\mathbf{R}}_\eta[t]$, and weigh exponentially less past observations [64]. Note that the correlation matrices $\tilde{\mathbf{R}}_\nu[t], \tilde{\mathbf{R}}_\eta[t]$ can be updated in an streaming way

$$\tilde{\mathbf{R}}_\nu[t] = \gamma_\nu\tilde{\mathbf{R}}_\nu[t-1] + \hat{\mathbf{f}}_\nu[t|t]\hat{\mathbf{f}}_\nu^T[t|t] \tag{5.17a}$$

$$\tilde{\mathbf{R}}_\eta[t] = \gamma_\eta\tilde{\mathbf{R}}_\eta[t-1] + \hat{\boldsymbol{\eta}}[t|t]\hat{\boldsymbol{\eta}}^T[t|t] \tag{5.17b}$$

which significantly reduces the storage complexity of estimating the sample matrices. We will refer to problems (5.15a) and (5.15b) as *kernel matching*, since the formulation is similar to the *covariance matching* [65].

5.4.2 PGD solver for kernel matching

Next, an efficient solver for (5.15a), and (5.15b), that exploits the structure of the Laplacian matrices, will be introduced. Consider the general problem of

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\geq\mathbf{0}}{\arg\min}\ \mathrm{Tr}\left(\mathbf{R}\mathbf{K}^{-1}(\boldsymbol{\theta})\right) + \rho\|\boldsymbol{\theta}\|_2^2 \quad \text{s.t.}\ \ \mathbf{K}(\boldsymbol{\theta}) = \sum_m^M \theta^{(m)}\mathbf{K}^{(m)} \tag{5.18}$$

The ensuing propositions will come handy in developing efficient solvers of (5.18).

Proposition 1. *Let* $\{\mathbf{K}^{(m)}\}_{m=1}^M$ *be Laplacian kernels [cf. Sec. 5.2.2] with corresponding eigenvalue decompositions* $\{\mathbf{U}\,\mathrm{diag}\left\{\boldsymbol{\lambda}^{(m)}\right\}\mathbf{U}^T\}_{m=1}^M$ *and* $\mathbf{T} := \mathbf{U}^T\mathbf{R}\mathbf{U}$*. Then the objective in (5.18) can be equivelantly written as*

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\geq\mathbf{0}}{\arg\min}\ \ \phi(\boldsymbol{\theta}) := \mathrm{Tr}\left(\mathbf{T}\boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta})\right) + \rho\|\boldsymbol{\theta}\|_2^2 \quad s.t.\ \ \boldsymbol{\Lambda}(\boldsymbol{\theta}) := \mathrm{diag}\left\{\sum_m^M \theta^{(m)}\boldsymbol{\lambda}^{(m)}\right\} \tag{5.19}$$

Proof. Since the Laplacian kernels have common eigenspace it holds that $\mathbf{K}(\boldsymbol{\theta}) = \sum_m^M \theta^{(m)} \times \mathbf{U}\,\mathrm{diag}\left\{\boldsymbol{\lambda}^{(m)}\right\}\mathbf{U}^T = \mathbf{U}\boldsymbol{\Lambda}(\boldsymbol{\theta})\mathbf{U}^T$. The reformulation of (5.18) follows using the invariance of the trace under cyclic permutations, that suggest $\mathrm{Tr}(\mathbf{R}\mathbf{K}^{-1}(\boldsymbol{\theta})) = \mathrm{Tr}(\mathbf{R}\mathbf{U}\boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta})\mathbf{U}^T) = \mathrm{Tr}(\boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta})\mathbf{U}^T\mathbf{R}\mathbf{U}) = \mathrm{Tr}(\boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta})\mathbf{T})$ □

Proposition 2. *Cost* $\phi(\boldsymbol{\theta})$ *is convex and differentiable with gradient*

$$\nabla\phi(\boldsymbol{\theta}) = \mathbf{v} + 2\rho\boldsymbol{\theta} \tag{5.20}$$

where $\mathbf{v} := -[\mathrm{Tr}(\mathrm{diag}\left\{\tilde{\boldsymbol{\lambda}}^{(1)}\right\}\mathbf{T}), \ldots, \mathrm{Tr}(\mathrm{diag}\left\{\tilde{\boldsymbol{\lambda}}^{(M)}\right\}\mathbf{T})]$*, with* $\tilde{\boldsymbol{\lambda}}^{(m)} := [\tilde{\lambda}_1^{(m)}, \ldots, \tilde{\lambda}_N^{(m)}]^T$ *and* $\tilde{\lambda}_n^{(m)} := \frac{\lambda_n^{(m)}}{(\sum_{\mu=1}^M \theta^{(\mu)}\lambda_n^{(\mu)})^2}$.

Proof. The convexity of (5.19) follows since it can be reformulated to a semidefinite program by minimizing the auxiliary variable w subject to the constraint $w \geq \mathrm{Tr}\left(\mathbf{Z}\boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta})\mathbf{Z}\right)$, where

$\mathbf{Z} := \sqrt{\mathbf{T}}$. The latter is expressed as a linear matrix inequality using Shur's complement [66]

$$\begin{aligned} \hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\geq\mathbf{0}, w>0}{\arg\min} \;\; & w + \rho\|\boldsymbol{\theta}\|_2^2 \\ \text{s.t.} \;\; & \begin{bmatrix} \operatorname{diag}\left\{\sum_m^M \theta^{(m)}\boldsymbol{\lambda}^{(m)}\right\} & \mathbf{Z} \\ \mathbf{Z} & \frac{w}{N}\mathbf{I} \end{bmatrix} \succcurlyeq \mathbf{0} \end{aligned} \tag{5.21}$$

which is a convex objective with convex constraints. To obtain the gradient observe that

$$\frac{\partial \phi}{\partial \theta^{(m)}} = -\operatorname{Tr}(\boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta}) \operatorname{diag}\left\{\boldsymbol{\lambda}^{(m)}\right\} \boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta})\mathbf{T}) + 2\rho\theta^{(m)} \tag{5.22}$$

and $\boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta}) \operatorname{diag}\left\{\boldsymbol{\lambda}^{(m)}\right\} \boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta}) = \operatorname{diag}\left\{\tilde{\boldsymbol{\lambda}}^{(M)}\right\}$ which follows since matrix $\boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta})$ is diagonal with entries $\left\{\frac{1}{\sum_{\mu=1}^M \theta^{(\mu)}\lambda_n^{(\mu)}}\right\}_{n=1}^N$ on the diagonal. □

The calculation of the gradient in (5.20) exploits the common eigenspace of $\{\mathbf{K}^{(m)}\}_{m=1}^M$ and avoids the inversion of an $N \times N$ matrix, $\mathbf{K}(\boldsymbol{\theta})$, that is required when calculating the gradient for the general formulation (5.18), [67, 68]. The complexity of evaluating the gradient is reduced from a prohibitive $\mathcal{O}(N^3M)$ for general kernels to an affordable $\mathcal{O}(NM)$ for Laplacian kernels, which is essential especially in the case of large-scale networks.

The latter small complexity motivates projected gradient descent (PGD) algorithms [69] for finding the global minimum in (5.19). The general PGD iteration follows

$$\boldsymbol{\theta}^{k+1} = \left[\boldsymbol{\theta}^k - s^k \nabla\phi(\boldsymbol{\theta}^k)\right]^+ \tag{5.23}$$

where s^k is the stepsize chosen by Armijo rule [69], $\boldsymbol{\theta}^0$ is a feasible initial step, and $\left[\cdot\right]^+$ denotes projection on the set $\Theta = \{\boldsymbol{\theta} : \theta^{(m)} \geq 0,\ m = 1 \dots M\}$. The convergence analysis of PGD is well studied [69] and in the case of strongly convex smooth functions as $\phi(\boldsymbol{\theta})$ PGD guarantees linear convergence rate. For faster implementation the constrained Newton's method can be considered, that guarantees superlinear convergence rate [69].

Algorithm 4 Online Multi-kernel learning (OMK)

Input: $\{\mathbf{K}^{(m)} \in \mathbb{S}_+^N\}_{m=1}^M$; $\mathbf{R}[t-1] \in \mathbb{S}_+^N$;
$\hat{\boldsymbol{\theta}}[t-1] \in \mathbb{R}_+^M$; $\boldsymbol{\xi}[t] \in \mathbb{R}^N$.
$\boldsymbol{\theta}^0 = \hat{\boldsymbol{\theta}}[t-1]$
$\mathbf{R}[t] = \gamma \mathbf{R}[t-1] + \boldsymbol{\xi}[t]\boldsymbol{\xi}^T[t]$

while stopping_criterion not met **do**

$$\boldsymbol{\theta}^{k+1} = \left[\boldsymbol{\theta}^k - s^k \nabla \phi(\boldsymbol{\theta}^k)\right]^+$$

end while
$\mathbf{K}[t] = \sum_m^M \hat{\theta}^{(m)}[t]\mathbf{K}^{(m)}$

Output: $\hat{\boldsymbol{\theta}}[t]; \mathbf{R}[t]; \mathbf{K}[t]$.

5.4.3 Online Multi-kernel Learning Algorithm

This section present the complete multi-kernel KrKF algorithm. The optimization problems in (5.15a), and (5.15b) need to be solved at every t. The PGD solver developed for (5.19) can be applied sequentially at each t to obtain a time-varying estimate $\hat{\boldsymbol{\theta}}[t]$. Since the rank one update in (5.17b), (5.17a) will not change significantly the objectives in (5.15a), and (5.15b) the PGD solver at t can be *warm started*, with $\boldsymbol{\theta}^0 = \hat{\boldsymbol{\theta}}[t-1]$. It has been observed that projected gradient descent can perform just as well as accelerated version when using warm starts [70]. The online algorithm for multi-kernel learning (OMK) is summarized with Algorithm 4. Notice that since the estimates $\hat{\boldsymbol{\theta}}[t]$ change over time, the combined kernel matrix $\mathbf{K}[t]$ changes as well.

Next, consider the dictionaries of kernels $\mathcal{D}_\nu := \{\mathbf{K}_\nu^{(m)} \in \mathbb{S}_+^N\}_{m=1}^{M_\nu}$ and $\mathcal{D}_\eta := \{\mathbf{K}_\eta^{(m)} \in \mathbb{S}_+^N\}_{m=1}^{M_\eta}$. Algorithm 5 describes the novel multi-kernel KrKF (MKrKF) algorithm that solves (5.14) in an online and data-adaptive fashion. The algorithm is initialized with $\mathbf{K}_\nu[0] = \mathbf{K}_\nu^{(1)}$, $\mathbf{K}_\eta[0] = \mathbf{K}_\eta^{(1)}$ and by $\hat{\mathbf{f}}_\chi[0|0] = \mathbf{0}$, $\mathbf{M}[0|0] = \frac{1}{\mu_1}\mathbf{K}_\eta^{(1)}$.

Remark 8. *The derivation of the algorithms in this section considered a fixed kernel dictionary over time $\mathcal{D} = \{\mathbf{K}^{(m)} \in \mathbb{S}_+^N\}_{m=1}^M$. If the topology changes over time, the kernel matrices in $\mathcal{D}$ will change as well (5.8) . To accommodate this scenario one can restart Algorithm 5, whenever the topology change, at time t_c, and initialize $\hat{\mathbf{f}}_\chi[0|0] = \hat{\mathbf{f}}_\chi[t_c|t_c]$,*

Algorithm 5 Multi-kernel KrKF(MKrKF)

Input: $\mathcal{D}_\nu$; $\mathcal{D}_\eta$.

for $t = 1, \ldots$ **do**

Input: $\mathbf{P}[t] \in \mathbb{R}^{N\times N}$; $\mathbf{y}[t] \in \mathbb{R}^{S[t]}$; $\mathbf{S}[t] \in \{0,1\}^{S[t]\times N}$.

KKrKF($\mathbf{K}_\eta[t-1], \mathbf{K}_\nu[t-1], \mathbf{P}[t], \mathbf{y}[t], \mathbf{S}[t]$,
$\hat{\mathbf{f}}_\chi[t-1|t-1], \mathbf{M}[t-1|t-1]$)

OMK($\mathcal{D}_\nu, \tilde{\mathbf{R}}_\nu[t-1], \hat{\boldsymbol{\theta}}_\nu[t-1], \hat{\mathbf{f}}_\nu[t|t]$
OMK($\mathcal{D}_\eta, \tilde{\mathbf{R}}_\eta[t-1], \hat{\boldsymbol{\theta}}_\eta[t-1], \hat{\boldsymbol{\eta}}[t|t]$)

Output: $\hat{\mathbf{f}}_\chi[t|t]$; $\hat{\mathbf{f}}_\nu[t|t]$; $\mathbf{M}[t|t]$.

end for

$\mathbf{M}[0|0] = \mathbf{M}[t_c|t_c]$, *as well as replace the kernels in the dictionary with ones resulting from the new topology.*

5.5 Numerical tests

This section describes tests on synthetic and real graph functions over dynamic graphs which demonstrate the superior performance of KKrKF and MKrKF over competing alternatives.

The following competing reconstruction algorithms are considered: (i) The least mean-squares (LMS) algorithm in [24] with step size μ_{LMS}; (ii) the distributed least-squares reconstruction (DLSR) algorithm [23] with step sizes μ_{DLSR} and β_{DLSR} (both LMS and DLSR can track slowly time-varying B-bandlimited graph signals); (iii) The B-bandlimited instantaneous estimator (BL-IE) which uses the estimator in [10, 13] per slot t.

The synthetic experiments will evaluate the perfomance of the KKrKF algorithm. Specifically, algorithm 3 admits the following configuration: $\mathbf{K}_\nu[t]$ is a diffusion kernel with parameter σ in the first experiment and $\mathbf{K}_\nu[t]$ is a bandlimited kernel with parameters β, $\lambda_{\max}$(cf. Table 2.1) in the second experiment; $\mathbf{K}_\eta[t] = s_\eta \mathbf{I}_N$ with parameter $s_\eta > 0$; and a transition matrix $\mathbf{P}[t] = \alpha(\mathbf{A}[t-1] + \mathbf{I}_N)$ with parameter $\alpha > 0$.

The real data experiments will test the muli-kernel KrKF that is configured as follows:

$\mathcal{D}_\nu$ contains M_ν diffusion kernels with parameters $\{\sigma^{(m)}\}_{m=1}^{M_\nu}$ drawn from a Gaussian distribution with mean μ_ν and variance r_ν; $\mathcal{D}_\eta$ contains $\{{s_\eta}^{(m)}\mathbf{I}_N\}_{m=1}^{M_\eta}$ with ${s_\eta}^{(m)}, \forall m$ drawn from a Gaussian distribution with mean μ_η and variance r_η.

The performance of the aforementioned approaches is evaluated in terms of the normalized mean-square error

$$\text{NMSE}(\{\mathcal{S}[\tau]\}_{\tau=1}^{t}) := \frac{\text{E}\left[\sum_{\tau=1}^{t}\|\mathbf{S}^c[\tau](\mathbf{f}[\tau]-\hat{\mathbf{f}}[\tau|\tau])\|_2^2\right]}{\text{E}\left[\sum_{\tau=1}^{t}\|\mathbf{S}^c[\tau]\mathbf{f}[\tau]\|_2^2\right]}$$

where the expectation is taken over the sample locations, and $\mathbf{S}^c[\tau]$ is an $(N-S[\tau])\times N$ matrix comprising the rows of $\mathbf{I}_N$ whose indices are not in $\mathcal{S}[t]$. For all the tests, the sampling set is chosen uniformly at random without replacement over $\mathcal{V}$ and kept constant over time; that is $\mathcal{S}[t]=\mathcal{S},\ \forall t$.

5.5.1 Numerical tests on synthetic data

The first real dataset contains timestamped messages between students at the University of California, Irvine, exchanged over a social network [71], for a period of 90 days corresponding to 3 months. The sampling interval t is one day. A network was created where $\{A_{n,n'}[t]\}_{t=30(k-1)+1}^{t=30k}$ counts the number of messages exchanged between student n and n' in the k-th month. The resulting topology changes across months. A subset of $N=310$ users for which $\mathbf{A}[t]$ corresponds to a connected graph $\forall t$ was selected. At each time t, $\mathbf{f}[t]$ was generated by adding a temporally uncorrelated B-bandlimited component with $B=5$ and a spatio-temporally correlated component. Specifically, $\mathbf{f}[t]=\sum_{i=1}^{5}\gamma_i[t]\boldsymbol{u}_i[t]+\mathbf{f}_\chi[t]$, where $\mathbf{f}_\chi[t]$ follows (5.3), $\{\gamma_i[t]\}_{i=1}^{5}$ are standardized Gaussian distributed for all t, and $\{\boldsymbol{u}_i[t]\}_{i=1}^{5}$ are the eigenvectors associated with the 5 smallest eigenvalues of the Laplacian matrix at time t.

The first experiment justifies the proposed decomposition by assessing the impact of dropping each term on the right hand side of (5.2). Fig. 5.1 depicts the NMSE over the time index for KKrKF; the Kalman filter (KF) estimator, which results from setting $\mathbf{f}_\nu[t]=\mathbf{0}$ for all t in the KKrKF, as well as kernel Kriging (KKr), which the KKrKF reduces to

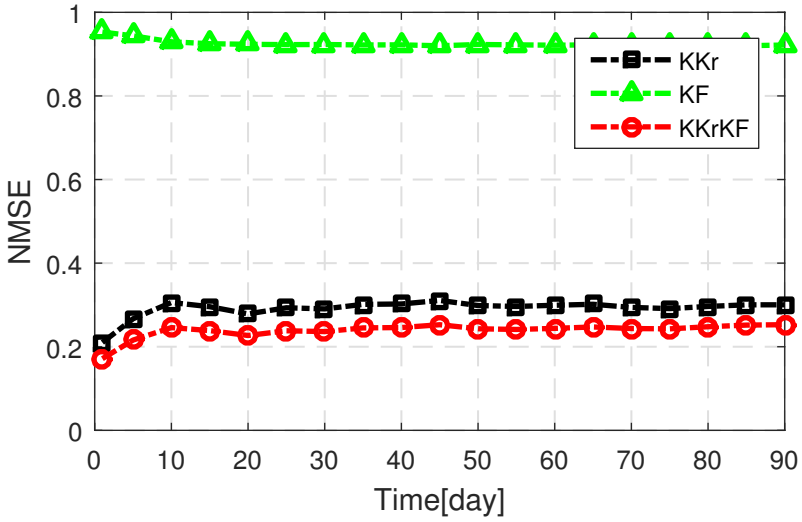


Figure 5.1: NMSE of function estimates. ($\mu_1 = 1$, $\mu_2 = 1$, $\sigma = 1.5$, $\alpha = 0.028$, $s_\eta = 0.05$)

if $\mathbf{f}_\chi[t] = \mathbf{0}$ for all t. As observed, the novel algorithm, which accounts for both terms, is capable of efficiently capturing the spatial as well as the temporal dynamics over time-evolving topologies.

The next experiment aims at evaluating the robustness of the reconstruction performance of the kernel KrKF, in the case of abrupt changes of the underlying network connections. Synthetic time-evolving networks of size $N = 81$ were generated using the *Kronecker product* model, that effectively captures properties of real graphs [72, 73]. Towards this end, consider the prescribed "seed matrix"

$$\mathbf{D}_0 := \begin{bmatrix} 1 & 0.1 & 0.7 \\ 0.3 & 0.1 & 0.5 \\ 0 & 1 & 0.1 \end{bmatrix}$$

that produces the binary-valued $N \times N$ matrix using Kronecker products, $\mathbf{D} := \mathbf{D}_0 \otimes \mathbf{D}_0 \otimes \mathbf{D}_0 \otimes \mathbf{D}_0$. The entries of the initial adjacency were selected as $A_{n,n'}[0] \sim \text{Bernulli}(D_{n,n'})\ \forall n, n'$. Next, the following time-evolving graph model is considered: periodically at each $t_p = 10\kappa$, $\kappa = 1, 2, \ldots$, each entry of $\mathbf{A}[t_p]$ changes with probability $p = 0.1$ as follows, $A_{n,n'}[t_p] = A_{n,n'}[t_p + 1] + \xi$, where the additive noise is sampled as $\xi \sim \mathcal{N}(0, \sigma_A)$. Different time-

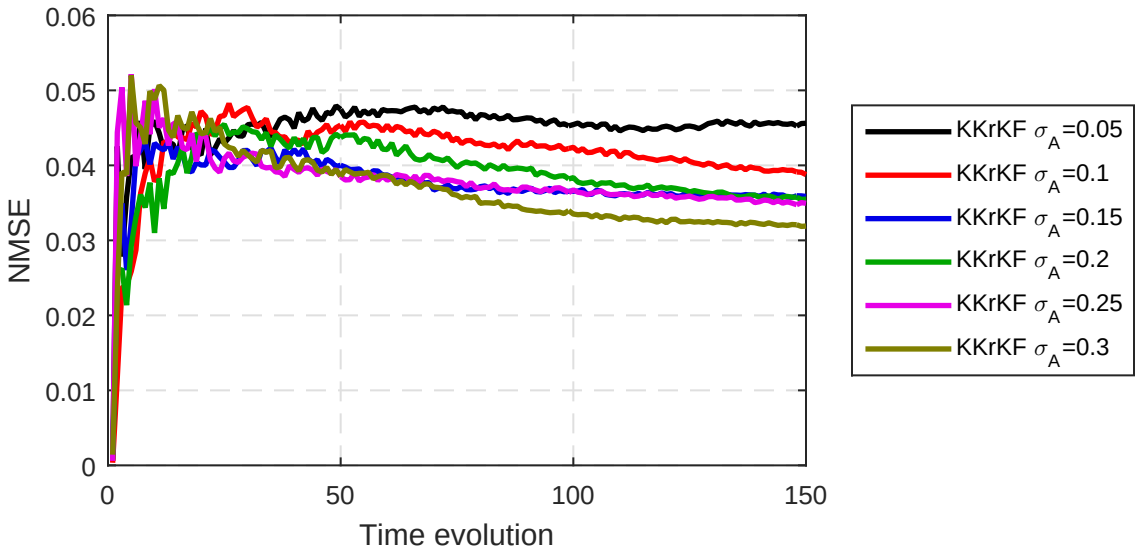


Figure 5.2: NMSE of function estimates. ($\mu_1 = 1$, $\mu_2 = 1$, $\beta = 1000$, $\lambda_{\max} = 10$ $\alpha = 10^{-3}$, $s_\eta = 10^{-4}$)

evolving graphs were generated using different σ_A. One graph function was generated for each time-evolving graph using the following model

$$\mathbf{f}[t] = \delta \mathbf{A}[t]\mathbf{f}[t-1] + \sum_{i=1}^{10} \gamma_i[t]\boldsymbol{u}_i[t] \tag{5.24}$$

where $\delta = 10^{-2}$ is a forgetting factor controlling the effect of the autoregressive part [53] and $\sum_{i=1}^{10} \gamma_i[t]\boldsymbol{u}_i[t]$ is a graph-bandlimited component as in the previous experiment and $\boldsymbol{u}_i[t]$ represent the time-evolving eigenvectors of the corresponding graph. Fig. 5.2 shows the NMSE reconstruction perfomance of the proposed algorithm, under different σ_A. Clearly, KKrKF exhibits robustness under diffent time-varying graph models and achieves low NMSE.

5.5.2 Numerical tests on real data

The second dataset is provided by the National Climatic Data Center [40], and comprises hourly temperature measuments at $N = 109$ measuring stations across the continental United States in 2010. A time-invariant graph was constructed as in [28], based on ge-

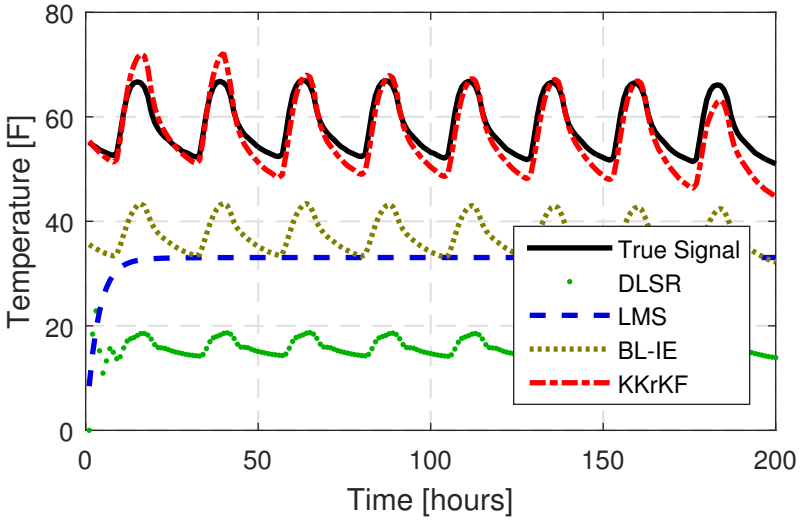


Figure 5.3: True temperature values along with the estimated ones. ($\mu_1 = 1$, $\mu_2 = 1$, $\sigma = 1.8$, $B = 5$, $\mu_{\text{DLSR}} = 1.2$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 0.6$, $\alpha = 10^{-3}$, $s_\eta = 10^{-5}$)

ographical distances. The value $f_n[t]$ represents the temperature recorded at the n-th station and t-th sample. The sampling interval is one hour for the first experiment and one day for the second.

Next, the performance of the different reconstruction algorithms is evaluated in tracking the temperature values. Fig. 5.3 depicts the true temperature value along with the estimates of the different algorithms for a station n that is not sampled, i.e. $n \notin \mathcal{S}$, with $S = 40$. Clearly, KKrKF accurately tracks the temperature by exploiting spatial and temporal information. On the other hand, DLSR and LMS cannot capture the fast signal variations.

Finally, Fig. 5.4 compares the NMSE of all considered approaches for $S = 40$, the KKrKF with a diffusion kernel for $\mathbf{K}_\nu$ with $\sigma = 1.8$ and for $\mathbf{K}_\eta = s_\eta \mathbf{I}_N$ with $s_\eta = 10^{-5}$, as well as its multi-kernel version the MKrKF. The kernel selections for KKrKF was known apriori that achieves optimal perfomance. As observed, MKrKF captures the spatio-temporal dynamics, successfully explores the pool of available kernels, and achieves a superior perfomance.

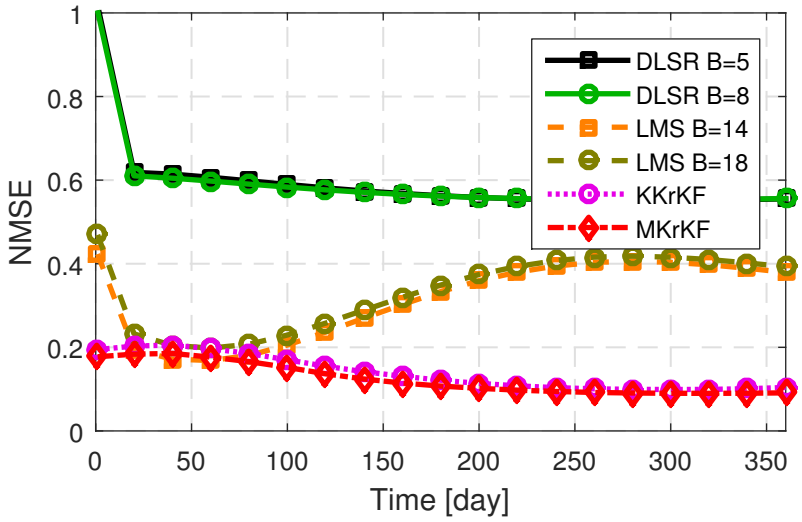


Figure 5.4: NMSE of temperature estimates. ($\mu_1 = 1$, $\mu_2 = 1$, $\mu_{\text{DLSR}} = 1.6$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 0.6$, $\alpha = 10^{-3}$, $\mu_\eta = 10^{-5}$, $r_\eta = 10^{-6}$, $\mu_\nu = 2$, $r_\nu = 0.5$, $M_\nu = 40$, $M_\eta = 40$)

Chapter 6

Concluding remarks and outlook

6.1 Summary

This thesis investigated kernel-based reconstruction methods for functions on graphs. Novel semi-parametric estimators were introduced that model the graph function as a superposition of a nonparametric component that promotes smoothness via graph kernels, and a parametric component that represents prior information captured through a known basis. The proposed algorithm outperfoms competing approaches, that can be subsumed under the encompassing semi-parametric framework.

To accommodate time evolving graph functions, this thesis introduced the notion of an extended graph, which regards the time dimension just as a spatial dimension. Several kernel designs were considered together with a batch and an online function estimators. The latter is a kernel Kalman filter developed from a purely deterministic standpoint without any need to adopt probabilistic state-space model assumptions.

Next, a model that explicitly accounts for the underlying dynamics was introduced and a novel kernel kriged Kalman filtering approach was derived that leverages graph kernels for reconstructing the function of interest. The proposed estimator is able to adapt to the observed data using an online multi-kernel technique that dynamically explores a pool of multiple kernels.

Extensive numerical tests on synthetic and real data-sets demonstrated the competi-

tive performance of the proposed algorithms, that accurately capture the spatiotemporal dynamics of the graph functions, over their state-of-the-art counterparts.

6.2 Future directions

Future research will focus on extending the kernel-based reconstruction framework, proposed in this thesis, along four directions:

1. The parametric component in Chapter 3 was represented using a general parametric base. Designing over-complete graph-aware dictionaries [15, 22], and employing them in conjunction with graph kernels has the potential to improve the performance of semi-parametric estimators.

2. The advent of large-scale networks calls for estimators with reduced computational complexity. Towards this goal, distributed versions of the kernel filtering algorithms developed in Chapters 4 and 5 will be considered by building on [74–76].

3. Nodes in many real settings share connections over multiple graphs e.g. social network users connected over Facebook, LinkedIn, and Twitter. Reconstructing attributes of these nodes, taking into account the connections corresponding to different graphs, may significantly improve the inference performance see also [77] for a relevant approach to time-invariant graphs.

4. Adversarial nodes in a network may produce carefully manipulated data aiming to compromise the performance of a machine learning algorithm, as in spam attacks in social networks or infiltration of sensor networks by malicious opponents. Exploiting graph kernels and judicious modeling of the adversaries presents an interesting possibility to identify these outlying nodes, and robustify the inference algorithms.

Bibliography

[1] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models.* Springer New York, 2009.

[2] B. Baingana, P. Traganitis, G. B. Giannakis, and G. Mateos, "Big data analytics for social networks," *Graph-Based Social Media Analysis*, vol. 39, p. 293, 2016.

[3] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *Proc. Int. Conf. Mach. Learn.*, Sydney, Australia, Jul. 2002, pp. 315–322.

[4] A. J. Smola and R. I. Kondor, "Kernels and regularization on graphs," in *Learning Theory and Kernel Machines.* Springer, 2003, pp. 144–158.

[5] 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 Sig. Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[6] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Sig. Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[7] O. Chapelle, B. Schölkopf, A. Zien *et al.*, *Semi-supervised Learning.* MIT press Cambridge, 2006.

[8] M. A. Kramer, E. D. Kolaczyk, and H. E. Kirsch, "Emergent network topology at seizure onset in humans," *Epilepsy Res.*, vol. 79, no. 2, pp. 173–186, 2008.

[9] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Trans. Sig. Process.*, vol. 64, no. 18, pp. 4845–4860, Sep. 2016.

[10] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data," in *Global Conf. Sig. Inf. Process.* Austin, Texas: IEEE, 2013, pp. 491–494.

[11] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Sig. Process.*, vol. 63, no. 9, pp. 2432–2444, May 2015.

[12] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Trans. Sig. Process.*, vol. 64, no. 7, pp. 1832–1843, Apr. 2016.

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

[14] S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, "Reconstruction of graph signals through percolation from seeding nodes," *IEEE Trans. Sig. Process.*, vol. 64, no. 16, pp. 4363–4378, Aug. 2016.

[15] D. Thanou, D. I. Shuman, and P. Frossard, "Learning parametric dictionaries for signals on graphs," *IEEE Trans. Sig. Process.*, vol. 62, no. 15, pp. 3849–3862, Aug. 2014.

[16] A. S. Zamzam, V. N. Ioannidis, and N. D. Sidiropoulos, "Coupled graph tensor factorization," in *Proc. Asilomar Conf. Sig., Syst., Comput.*, Pacific Grove, CA, 2016, pp. 1755–1759.

[17] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.

[18] D. Berberidis and G. B. Giannakis, "Active sampling for graph-cognizant classification via expected model change," *arXiv preprint arXiv:1705.07220*, 2017.

[19] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Trans. Sig. Process.*, vol. 65, no. 3, pp. 764–778, 2017.

[20] F. R. Bach and M. I. Jordan, "Learning graphical models for stationary time series," *IEEE Trans. Sig. Process.*, vol. 52, no. 8, pp. 2189–2199, 2004.

[21] J. Mei and J. M. F. Moura, "Signal processing on graphs: Causal modeling of big data," *arXiv preprint arXiv:1503.00173v3*, 2016.

[22] P. A. Forero, K. Rajawat, and G. B. Giannakis, "Prediction of partially observed dynamical processes over networks via dictionary learning," *IEEE Trans. Sig. Process.*, vol. 62, no. 13, pp. 3305–3320, Jul. 2014.

[23] X. Wang, M. Wang, and Y. Gu, "A distributed tracking algorithm for reconstruction of graph signals," *IEEE J. Sel. Topics Sig. Process.*, vol. 9, no. 4, pp. 728–740, 2015.

[24] P. D. Lorenzo, S. Barbarossa, P. Banelli, and S. Sardellitti, "Adaptive least mean squares estimation of graph signals," *IEEE Trans. Sig. Info. Process. Netw.*, vol. Early Access, 2016.

[25] K. Rajawat, E. Dall'Anese, and G. B. Giannakis, "Dynamic network delay cartography," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2910–2920, 2014.

[26] V. Kekatos, Y. Zhang, and G. B. Giannakis, "Electricity market forecasting via low-rank multi-kernel learning," *IEEE J. Sel. Topics Sig. Process.*, vol. 8, no. 6, pp. 1182–1193, 2014.

[27] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, no. Jul, pp. 2211–2268, 2011.

[28] D. Romero, V. N. Ioannidis, and G. B. Giannakis, "Kernel-based reconstruction of space-time functions on dynamic graphs," *IEEE J. Sel. Topics Sig. Process.*, vol. 11, no. 6, pp. 1–14, Sep. 2017.

[29] V. N. Ioannidis, D. Romero, and G. B. Giannakis, "Inference of spatiotemporal processes over graphs via kernel kriged kalman filtering," in *Proc. European Sig. Process. Conf. (to appear)*, Kos, Greece, Aug. 2017.

[30] V. N. Ioannidis, A. N. Nikolakopoulos, and G. B. Giannakis, "Semi-parametric graph kernel-based reconstruction," in *Global Conf. Sig. Inf. Process. (to appear)*, Montreal, Canada, Nov. 2017.

[31] V. N. Ioannidis, D. Romero, and G. B. Giannakis, "Kernel-based reconstruction of space-time functions via extended graphs," in *Proc. Asilomar Conf. Sig., Syst., Comput.*, Pacific Grove, CA, Nov. 2016, pp. 1829 – 1833.

[32] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *ICML Workshop Stat. Relational Learn. Connections Other Fields*, vol. 15, Banff, Canada, Jul. 2004, pp. 67–68.

[33] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2002.

[34] A. N. Nikolakopoulos and J. D. Garofalakis, "Ncdrec: A decomposability inspired framework for top-n recommendation," in *2014 IEEE/WIC/ACM International Joint*

Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 1, Aug 2014, pp. 183–190.

[35] A. N. Nikolakopoulos, M. A. Kouneli, and J. D. Garofalakis, "Hierarchical itemspace rank: Exploiting hierarchy to alleviate sparsity in ranking-based recommendation," *Neurocomputing*, vol. 163, pp. 126–136, Sep. 2015.

[36] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Computational Learning Theory*. Springer, 2001, pp. 416–426.

[37] V. Vapnik, *The nature of statistical learning theory*. Springer, 2013.

[38] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Signal representations on graphs: Tools and applications," arXiv preprint arXiv:1512.05406 [Online]. Available: http://arxiv.org/abs/1512.05406, 2015.

[39] U. Von Luxburg, "A tutorial on spectral clustering," vol. 17, no. 4, pp. 395–416, Dec. 2007.

[40] "1981-2010 U.S. climate normals," [Online]. Available: https://www.ncdc.noaa.gov.

[41] L. Ralaivola and F. D'Alché-Buc, "Time series filtering, smoothing and learning using the kernel kalman filter," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 3, Montreal, Canada, 2005, pp. 1449–1454.

[42] P. Zhu, B. Chen, and J. C. Príncipe, "Learning nonlinear generative models of time series with a Kalman filter in RKHS," *IEEE Trans. Sig. Process.*, vol. 62, no. 1, pp. 141–155, Jan. 2014.

[43] K. Wehmuth, A. Ziviani, and E. Fleury, "A unifying model for representing time-varying graphs," in *Int. Conf. Data Sci. Advanced Analytics*, Paris, France, Oct. 2015, pp. 1–10.

[44] G. Strang and K. Borre, *Linear algebra, geodesy, and GPS*. Siam, 1997.

[45] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *American Institute Aeronautics Astronautics J.*, vol. 3, no. 8, pp. 1445–1450, 1965.

[46] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Separable autoregressive moving average graph-temporal filters," in *Proc. European Sig. Process. Conf.*, Budapest, Hungary, 2016, pp. 200–204.

[47] P. M. Weichsel, "The Kronecker product of graphs," *Proc. American Mathematical Society*, vol. 13, no. 1, pp. 47–52, 1962.

[48] H. Kashima, S. Oyama, Y. Yamanishi, and K. Tsuda, "On pairwise kernels: An efficient alternative and generalization analysis," in *Pacific-Asia Conf. Knowledge Discovery Data Mining*, 2009, pp. 1030–1037.

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

[50] "Bureau of economic analysis," [Online]. Available: http://www.bea.gov/iTable/index_industry_io.cfm.

[51] Y. Shen, B. Baingana, and G. B. Giannakis, "Nonlinear structural vector autoregressive models for inferring effective brain network connectivity," *arXiv preprint arXiv:1610.06551*, 2016.

[52] T. W. Anderson, *An introduction to multivariate statistical analysis*. Wiley New York, 1958, vol. 2.

[53] Y. Shen, B. Baingana, and G. B. Giannakis, "Nonlinear structural vector autoregressive models for inferring effective brain network connectivity," *arXiv preprint arXiv:1610.06551v1*, 2016.

[54] N. Cressie, *Statistics for spatial data*. New York: Wiley, 1993.

[55] C. K. Wikle and N. Cressie, "A dimension-reduced approach to space-time kalman filtering," *Biometrika*, pp. 815–829, 1999.

[56] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[57] S.-J. Kim, E. Dall'Anese, and G. B. Giannakis, "Cooperative spectrum sensing for cognitive radios using kriged Kalman filtering," *IEEE J. Sel. Topics Sig. Process.*, vol. 5, no. 1, pp. 24–36, feb. 2011.

[58] K. V. Mardia, C. Goodall, E. J. Redfern, and F. J. Alonso, "The kriged kalman filter," *Test*, vol. 7, no. 2, pp. 217–282, 1998.

[59] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, 2004.

[60] X. Zhu, J. Kandola, Z. Ghahramani, and J. D. Lafferty, "Nonparametric transforms of graph kernels for semi-supervised learning," in *Proc. Advances Neural Inf. Process. Syst.*, Vancouver, Canada, 2004, pp. 1641–1648.

[61] H. Xia, S. C. Hoi, R. Jin, and P. Zhao, "Online multiple kernel similarity learning for visual search," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 36, no. 3, pp. 536–549, 2014.

[62] R. Jin, S. C. Hoi, and T. Yang, "Online multiple kernel learning: Algorithms and mistake bounds," in *Proc. 21st Int'l Conf. Algorithmic Learning Theory (ALT '10)*. Springer, 2010, pp. 390–404.

[63] I. Csiszár and G. Tusnády, "Information geometry and alternating minimization procedures," *Statistics and Decisions*, pp. 205–237, 1984.

[64] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where rls meets the ℓ_1-norm," *IEEE Trans. Sig. Process.*, vol. 58, no. 7, pp. 3436–3447, Jul. 2010.

[65] D. Romero and G. Leus, "Wideband spectrum sensing from compressed measurements using spectral prior information," *IEEE Trans. Sig. Process.*, vol. 61, no. 24, pp. 6232–6246, 2013.

[66] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[67] L. Zhang, D. Romero, and G. B. Giannakis, "Fast convergent algorithms for multi-kernel regression," in *Proc. Workshop Stat. Sig. Process.*, Palma de Mallorca, Spain, Jun. 2016.

[68] C. Cortes, M. Mohri, and A. Rostamizadeh, "Learning non-linear combinations of kernels," in *Proc. Advances Neural Inf. Process. Syst.*, 2009, pp. 396–404.

[69] D. Bertsekas, *Nonlinear Programming*. Athena scientific Belmont, 1999.

[70] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Sig. Process.*, vol. 58, no. 10, pp. 5262–5276, 2010.

[71] "Snap temporal networks: Collegemsg," [Online]. Available: http://snap.stanford.edu/data/CollegeMsg.html.

[72] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *J. Mach. Learn. Res.*, vol. 11, no. Feb, pp. 985–1042, 2010.

[73] Y. Shen, B. Baingana, and G. B. Giannakis, "Tensor decompositions for identifying directed graph topologies and tracking dynamic networks," *IEEE Trans. Sig. Process.*, 2017.

[74] E. Dall'Anese, S.-J. Kim, and G. B. Giannakis, "Channel gain map tracking via distributed kriging," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1205–1211, 2011.

[75] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc wsns with noisy links—part ii: Distributed estimation and smoothing of random signals," *IEEE Trans. Sig. Process.*, vol. 56, no. 4, pp. 1650–1666, Apr. 2008.

[76] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy links—part i: Distributed estimation of deterministic signals," *IEEE Trans. Sig. Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[77] P. A. Traganitis, Y. Shen, and G. B. Giannakis, "Topology inference for multilayer networks," in *Proc. of Intl. Workshop on Network Science for Comms*, Atlanta, GA, May 2017.

Appendix A

Space-time kernels

A.1 Proof of Lemma 1

Start by expanding the norm in the second term of (B.6) to obtain

$$\begin{aligned}
&\sum_{\tau=2}^{T} ||\mathbf{f}[\tau] - \mathbf{P}[\tau]\mathbf{f}[\tau-1]||^2_{\mathbf{\Sigma}[\tau]} \\
=&\sum_{\tau=2}^{T} (\mathbf{f}[\tau] - \mathbf{P}[\tau]\mathbf{f}[\tau-1])^T \mathbf{\Sigma}^{-1}[\tau](\mathbf{f}[\tau] - \mathbf{P}[\tau]\mathbf{f}[\tau-1]) \\
=&\sum_{\tau=2}^{T} \Bigg(\mathbf{f}^T[\tau]\mathbf{\Sigma}^{-1}[\tau]\mathbf{f}[\tau] - 2\mathbf{f}^T[\tau]\mathbf{\Sigma}^{-1}[\tau]\mathbf{P}[\tau]\mathbf{f}[\tau-1] \\
&+ \mathbf{f}^T[\tau-1]\mathbf{P}^T[\tau]\mathbf{\Sigma}^{-1}[\tau]\mathbf{P}[\tau]\mathbf{f}[\tau-1] \Bigg).
\end{aligned} \tag{A.1}$$

Noting that

$$\begin{aligned}
\sum_{\tau=2}^{T} \mathbf{f}^T[\tau]\mathbf{\Sigma}^{-1}[\tau]\mathbf{f}[\tau] = &\sum_{\tau=2}^{T} \mathbf{f}^T[\tau-1]\mathbf{\Sigma}^{-1}[\tau-1]\mathbf{f}[\tau-1] \\
&- \mathbf{f}^T[1]\mathbf{\Sigma}^{-1}[1]\mathbf{f}[1] + \mathbf{f}^T[T]\mathbf{\Sigma}^{-1}[T]\mathbf{f}[T]
\end{aligned}$$

it readily follows from (A.1) that

$$\begin{aligned}
&\sum_{\tau=2}^{T} ||\mathbf{f}[\tau] - \mathbf{P}[\tau]\mathbf{f}[\tau-1]||^2_{\mathbf{\Sigma}[\tau]} \\
=&\sum_{\tau=2}^{T} \Bigg(\mathbf{f}^T[\tau-1]\Big(\mathbf{\Sigma}^{-1}[\tau-1] + \mathbf{P}^T[\tau]\mathbf{\Sigma}^{-1}[\tau]\mathbf{P}[\tau]\Big)\mathbf{f}[\tau-1] \\
&- 2\mathbf{f}^T[\tau]\mathbf{\Sigma}^{-1}[\tau]\mathbf{P}[\tau]\mathbf{f}[\tau-1] \Bigg) \\
&- \mathbf{f}^T[1]\mathbf{\Sigma}^{-1}[1]\mathbf{f}[1] + \mathbf{f}^T[T]\mathbf{\Sigma}^{-1}[T]\mathbf{f}[T].
\end{aligned}$$

From Algorithm 1, it follows that $\mathbf{\Sigma}^{-1}[\tau-1]+\mathbf{P}^T[\tau]\mathbf{\Sigma}^{-1}[\tau]\mathbf{P}[\tau] = \mathbf{D}[\tau-1]$ and $-\mathbf{\Sigma}^{-1}[\tau]\mathbf{P}[\tau] = \mathbf{C}[\tau]$, which in turn imply that

$$\begin{aligned}
&\sum_{\tau=2}^{T} ||\mathbf{f}[\tau] - \mathbf{P}[\tau]\mathbf{f}[\tau-1]||^2_{\mathbf{\Sigma}[\tau]} \\
&= \sum_{\tau=2}^{T} \Bigg(\mathbf{f}^T[\tau-1]\mathbf{D}[\tau-1]\mathbf{f}[\tau-1] + 2\mathbf{f}^T[\tau]\mathbf{C}[\tau]\mathbf{f}[\tau-1] \Bigg) \\
&\quad - \mathbf{f}^T[1]\mathbf{\Sigma}^{-1}[1]\mathbf{f}[1] + \mathbf{f}^T[T]\mathbf{D}[T]\mathbf{f}[T] \\
&= \bar{\mathbf{f}}^T\bar{\mathbf{K}}^{-1}\bar{\mathbf{f}} - \mathbf{f}^T[1]\mathbf{\Sigma}^{-1}[1]\mathbf{f}[1] \qquad \text{(A.2)}
\end{aligned}$$

where the last equality follows from (4.7). After substituting (A.2) into (B.6) and recognizing that the first summand in (B.6) equals $||\bar{\mathbf{y}}[t] - \bar{\mathbf{S}}[t]\bar{\mathbf{f}}||^2_{\mathbf{D}_S[t]}$, expression (4.5a) is recovered and the proof is completed.

A.2 Proof of Th. 1

The first step is to simplify the objective in (B.6). To this end, note that minimizing (B.6) with respect to $\{\mathbf{f}[\tau]\}_{\tau=t'+1}^{T}$ for any t and t' such that $t' \geq t$ yields

$$\hat{\mathbf{f}}[\tau|t] = \mathbf{P}[\tau]\hat{\mathbf{f}}[\tau-1|t],\ \tau = t'+1,\ldots,T, \tag{A.3a}$$

$$\begin{aligned}\{\hat{\mathbf{f}}[\tau|t]\}_{\tau=1}^{t'} =& \underset{\{\mathbf{f}[\tau]\}_{\tau=1}^{t'}}{\arg\min} \sum_{\tau=1}^{t} \tfrac{1}{\sigma_e^2[\tau]}||\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}[\tau]||^2 \\ &+ \sum_{\tau=2}^{t'} ||\mathbf{f}[\tau] - \mathbf{P}[\tau]\mathbf{f}[\tau-1]||^2_{\mathbf{\Sigma}[\tau]} \\ &+ \mathbf{f}^T[1]\mathbf{\Sigma}^{-1}[1]\mathbf{f}[1].\end{aligned} \tag{A.3b}$$

The goal is therefore to show that the t-th iteration of Algorithm 2 returns $\hat{\mathbf{f}}[t|t]$ as given by (A.3b). To simplify notation, collect the function values up to time t as $\bar{\boldsymbol{f}}[t] := [\mathbf{f}^T[1], \mathbf{f}^T[2], \ldots, \mathbf{f}^T[t]]^T \in \mathbb{R}^{Nt}$ and their estimates given observations up to time t' as $\hat{\bar{\boldsymbol{f}}}[t|t'] := [\hat{\mathbf{f}}^T[1|t'], \hat{\mathbf{f}}^T[2|t'], \ldots, \hat{\mathbf{f}}^T[t|t']]^T \in \mathbb{R}^{Nt}$. The rest of the proof proceeds along the lines in [44, Ch. 17] by expressing $\hat{\bar{\boldsymbol{f}}}[t|t]$ and $\hat{\bar{\boldsymbol{f}}}[t|t-1]$ as the solutions to two least-squares problems. To this end, define the $Nt + \bar{S}[t] \times Nt$ matrix

$$\bar{\mathcal{A}}[t] := \begin{bmatrix} \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}[1] & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{P}[2] & \mathbf{I}_N & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}[2] & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{S}[t-1] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & -\mathbf{P}[t] & \mathbf{I}_N \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{S}[t] \end{bmatrix} \tag{A.4}$$

the $Nt + \bar{S}[t] \times Nt + \bar{S}[t]$ matrix $\bar{\mathbf{\Sigma}}[t] := \text{bdiag}\{\mathbf{\Sigma}[1], \sigma_e^2[1]\mathbf{I}_{S[1]}, \mathbf{\Sigma}[2], \sigma_e^2[2]\mathbf{I}_{S[2]}, \ldots, \sigma_e^2[t-1]\mathbf{I}_{S[t-1]},$

$\mathbf{\Sigma}[t], \sigma_e^2[t]\mathbf{I}_{S[t]}\}$, and note from (A.3b) that

$$\hat{\bar{\boldsymbol{f}}}[t|t] = \arg\min_{\bar{\boldsymbol{f}}[t]} ||\bar{\boldsymbol{\psi}}[t] - \bar{\mathcal{A}}[t]\bar{\boldsymbol{f}}[t]||^2_{\bar{\mathbf{\Sigma}}[t]} \tag{A.5}$$

where $\bar{\boldsymbol{\psi}}[t] := [\mathbf{0}_N^T, \mathbf{y}^T[1], \mathbf{0}_N^T, \mathbf{y}^T[2], \mathbf{0}_N^T, \ldots, \mathbf{0}_N^T, \mathbf{y}^T[t]]^T \in \mathbb{R}^{Nt+\bar{S}[t]}$. Indeed, expression (A.5) corresponds to the weighted least-squares solution to

$$\bar{\boldsymbol{\psi}}[t] = \bar{\mathcal{A}}[t]\bar{\boldsymbol{f}}[t] + \bar{\boldsymbol{\epsilon}}[t] \tag{A.6}$$

where $\bar{\boldsymbol{\epsilon}}[t] \in \mathbb{R}^{Nt+\bar{S}[t]}$ is an error vector, and admits the closed-form solution

$$\hat{\bar{\boldsymbol{f}}}[t|t] = (\bar{\mathcal{A}}^T[t]\bar{\mathbf{\Sigma}}^{-1}[t]\bar{\mathcal{A}}[t])^{-1}\bar{\mathcal{A}}^T[t]\bar{\mathbf{\Sigma}}^{-1}[t]\bar{\boldsymbol{\psi}}[t]. \tag{A.7}$$

Similarly, define the $Nt + \bar{S}[t-1] \times Nt$ matrix

$$\bar{\mathcal{A}}'[t] := \begin{bmatrix} \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}[1] & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{P}[2] & \mathbf{I}_N & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}[2] & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & -\mathbf{P}[t-1] & \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{S}[t-1] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & -\mathbf{P}[t] & \mathbf{I}_N \end{bmatrix} \tag{A.8}$$

which is a submatrix of $\bar{\mathcal{A}}[t]$ that results from removing the last block-row, together with the $Nt + \bar{S}[t-1] \times Nt + \bar{S}[t-1]$ matrix $\bar{\mathbf{\Sigma}}'[t] := \text{bdiag}\{\mathbf{\Sigma}[1], \sigma_e^2[1]\mathbf{I}_{S[1]}, \mathbf{\Sigma}[2], \sigma_e^2[2]\mathbf{I}_{S[2]}, \ldots, \mathbf{\Sigma}[t-1], \sigma_e^2[t-1]\mathbf{I}_{S[t-1]}, \mathbf{\Sigma}[t]\}$, which is a submatrix of $\bar{\mathbf{\Sigma}}[t]$ resulting from removing the last block-row and block-column. Now, replace t with $t-1$ and t' with t in (A.3b) to obtain

$$\hat{\bar{\boldsymbol{f}}}[t|t-1] = \arg\min_{\bar{\boldsymbol{f}}[t]} ||\bar{\boldsymbol{\psi}}'[t] - \bar{\mathcal{A}}'[t]\bar{\boldsymbol{f}}[t]||^2_{\bar{\mathbf{\Sigma}}'[t]} \tag{A.9}$$

where $\bar{\boldsymbol{\psi}}'[t] := [\mathbf{0}_N^T, \mathbf{y}^T[1], \mathbf{0}_N^T, \mathbf{y}^T[2], \mathbf{0}_N^T, \ldots, \mathbf{y}^T[t-1], \mathbf{0}_N^T]^T \in \mathbb{R}^{Nt+\bar{S}[t-1]}$ is a submatrix of $\bar{\boldsymbol{\psi}}[t]$ that results from removing its last block-row. In this case, $\hat{\bar{\boldsymbol{f}}}[t|t-1]$ in (A.9) corresponds to the least-squares solution to (A.6) after removing the last $S[t]$ equations, and can be obtained in closed form as

$$\hat{\bar{\boldsymbol{f}}}[t|t-1] = (\bar{\mathcal{A}}'^T[t]\bar{\boldsymbol{\Sigma}}'^{-1}[t]\bar{\mathcal{A}}'[t])^{-1}\bar{\mathcal{A}}'^T[t]\bar{\boldsymbol{\Sigma}}'^{-1}[t]\bar{\boldsymbol{\psi}}'[t]. \tag{A.10}$$

The rest of the proof utilizes (A.7) and (A.10) to express $\hat{\mathbf{f}}[t|t]$ in terms of $\hat{\mathbf{f}}[t|t-1]$, and $\hat{\mathbf{f}}[t|t-1]$ in terms of $\hat{\mathbf{f}}[t-1|t-1]$. To this end, define $\bar{\mathcal{J}}[t] := \boldsymbol{i}_{t,t}^T \otimes \mathbf{I}_N$, which can be used to select the last $N \times N$ block-row or block-column of a matrix, as well as

$$\mathbf{M}[t|t-1] := \bar{\mathcal{J}}[t](\bar{\mathcal{A}}'^T[t]\bar{\boldsymbol{\Sigma}}'^{-1}[t]\bar{\mathcal{A}}'[t])^{-1}\bar{\mathcal{J}}^T[t] \tag{A.11}$$

and

$$\mathbf{M}[t|t] := \bar{\mathcal{J}}[t](\bar{\mathcal{A}}^T[t]\bar{\boldsymbol{\Sigma}}^{-1}[t]\bar{\mathcal{A}}[t])^{-1}\bar{\mathcal{J}}^T[t], \tag{A.12}$$

which respectively correspond to the bottom right $N \times N$ blocks of $\bar{\mathcal{T}}'[t] := \bar{\mathcal{A}}'^T[t]\bar{\boldsymbol{\Sigma}}'^{-1}[t]\bar{\mathcal{A}}'[t]$ and $\bar{\mathcal{T}}[t] := \bar{\mathcal{A}}^T[t]\bar{\boldsymbol{\Sigma}}^{-1}[t]\bar{\mathcal{A}}[t]$. Expressions (A.12) and (A.11) will be used next to express $\mathbf{M}[t|t-1]$ in terms of $\mathbf{M}[t-1|t-1]$, and $\mathbf{M}[t|t]$ in terms of $\mathbf{M}[t|t-1]$.

Assume for simplicity that $\bar{\boldsymbol{\Sigma}}[t]$ and $\bar{\boldsymbol{\Sigma}}'[t]$ equal the identity matrices of appropriate sizes, although the proof easily carries over to arbitrary positive definite matrices $\bar{\boldsymbol{\Sigma}}[t]$ and $\bar{\boldsymbol{\Sigma}}'[t]$. Note that

$$\bar{\mathcal{T}}'[t] = \begin{bmatrix} \bar{\mathcal{T}}[t-1] + \boldsymbol{V}^T[t]\boldsymbol{V}[t] & \boldsymbol{V}^T[t] \\ \boldsymbol{V}[t] & \mathbf{I}_N \end{bmatrix} \tag{A.13}$$

where $\boldsymbol{V}[t] := -\mathbf{P}[t]\bar{\mathcal{J}}[t-1] \in \mathbb{R}^{N \times N(t-1)}$ and observe that $\mathbf{M}[t|t-1]$ is the bottom right

$N \times N$ block of $\bar{\mathcal{T}}'^{-1}[t]$. Thus, applying block matrix inversion to (A.13) yields

$$\begin{aligned}\mathbf{M}[t|t-1] =& \bar{\mathcal{J}}[t]\bar{\mathcal{T}}'^{-1}[t]\bar{\mathcal{J}}^T[t] \\ =& \Big(\mathbf{I}_N - \boldsymbol{V}[t]\Big(\bar{\mathcal{T}}[t-1] + \boldsymbol{V}^T[t]\boldsymbol{V}[t])^{-1}\boldsymbol{V}^T[t]\Big)^{-1}. \end{aligned} \tag{A.14}$$

Moreover, the matrix inversion lemma yields,

$$\begin{aligned}&(\bar{\mathcal{T}}[t-1] + \boldsymbol{V}^T[t]\mathbf{I}_N\boldsymbol{V}[t]\Big)^{-1} = \bar{\mathcal{T}}^{-1}[t-1] - \bar{\mathcal{T}}^{-1}[t-1]\times \\ &\boldsymbol{V}^T[t]\Big(\mathbf{I}_N + \boldsymbol{V}[t]\bar{\mathcal{T}}^{-1}[t-1]\boldsymbol{V}^T[t]\Big)^{-1}\boldsymbol{V}[t]\bar{\mathcal{T}}^{-1}[t-1]. \end{aligned} \tag{A.15}$$

Substituting (A.15) into (A.14), applying the definition of $\boldsymbol{V}[t]$, and using (A.12) to identify $\mathbf{M}[t-1|t-1]$ enables one to express $\mathbf{M}[t|t-1]$ in terms of $\mathbf{M}[t-1|t-1]$ as

$$\mathbf{M}[t|t-1] = \mathbf{I}_N + \mathbf{P}[t]\mathbf{M}[t-1|t-1]\mathbf{P}^T[t]. \tag{A.16}$$

On the other hand, to express $\mathbf{M}[t|t]$ in terms of $\mathbf{M}[t|t-1]$, note that $\bar{\mathcal{A}}[t] = [\bar{\mathcal{A}}'^T[t], \boldsymbol{W}^T[t]]^T$, where $\boldsymbol{W}[t] := \mathbf{S}[t]\bar{\mathcal{J}}[t] \in \mathbb{R}^{S[t]\times Nt}$. Therefore,

$$\begin{aligned}\bar{\mathcal{T}}[t] =& \bar{\mathcal{A}}^T[t]\bar{\mathcal{A}}[t] \\ =& \bar{\mathcal{A}}'^T[t]\bar{\mathcal{A}}'[t] + \boldsymbol{W}^T[t]\boldsymbol{W}[t] \\ =& \bar{\mathcal{T}}'[t] + \boldsymbol{W}^T[t]\boldsymbol{W}[t]. \end{aligned} \tag{A.17}$$

Applying the matrix inversion lemma to (A.17) yields

$$\begin{aligned}\bar{\mathcal{T}}^{-1}[t] =& \bar{\mathcal{T}}'^{-1}[t] - \bar{\mathcal{T}}'^{-1}[t]\boldsymbol{W}^T[t]\times \\ &\Big(\mathbf{I}_{S[t]} + \boldsymbol{W}[t]\bar{\mathcal{T}}'^{-1}[t]\boldsymbol{W}^T[t]\Big)^{-1}\boldsymbol{W}[t]\bar{\mathcal{T}}'^{-1}[t]. \end{aligned} \tag{A.18}$$

Substituting the definition of $\boldsymbol{W}[t]$ into (A.18) leads to

$$\begin{aligned}
&\bar{\mathcal{J}}[t]\bar{\mathcal{T}}^{-1}[t] = \bar{\mathcal{J}}[t]\bar{\mathcal{T}}'^{-1}[t] - \bar{\mathcal{J}}[t]\bar{\mathcal{T}}'^{-1}[t]\bar{\mathcal{J}}^T[t]\mathbf{S}^T[t]\times \\
&\Big(\mathbf{I}_{S[t]} + \mathbf{S}[t]\bar{\mathcal{J}}[t]\bar{\mathcal{T}}'^{-1}[t]\bar{\mathcal{J}}^T[t]\mathbf{S}^T[t]\Big)^{-1}\mathbf{S}[t]\bar{\mathcal{J}}[t]\bar{\mathcal{T}}'^{-1}[t] \\
&= \bar{\mathcal{J}}[t]\bar{\mathcal{T}}'^{-1}[t] - \mathbf{M}[t|t-1]\mathbf{S}^T[t]\times \\
&\quad\Big(\mathbf{I}_{S[t]} + \mathbf{S}[t]\mathbf{M}[t|t-1]\mathbf{S}^T[t]\Big)^{-1}\mathbf{S}[t]\bar{\mathcal{J}}[t]\bar{\mathcal{T}}'^{-1}[t] \\
&= (\mathbf{I}_N - \mathbf{G}[t]\mathbf{S}[t])\bar{\mathcal{J}}[t]\bar{\mathcal{T}}'^{-1}[t] \qquad \text{(A.19)}
\end{aligned}$$

where the second equality follows from (A.11), and the third from

$$\mathbf{G}[t] := \mathbf{M}[t|t-1]\mathbf{S}^T[t](\mathbf{I}_{S[t]} + \mathbf{S}[t]\mathbf{M}[t|t-1]\mathbf{S}^T[t])^{-1}. \qquad \text{(A.20)}$$

Finally, multiplying both sides of (A.19) with $\bar{\mathcal{J}}^T[t]$ and using (A.12) to identify $\mathbf{M}[t|t]$ enables one to express $\mathbf{M}[t|t]$ in terms of $\mathbf{M}[t|t-1]$ as

$$\mathbf{M}[t|t] = (\mathbf{I}_N - \mathbf{G}[t]\mathbf{S}[t])\mathbf{M}[t|t-1]. \qquad \text{(A.21)}$$

If $\bar{\boldsymbol{\Sigma}}[t]$ and $\bar{\boldsymbol{\Sigma}}'[t]$ are not identity matrices, then one obtains

$$\mathbf{M}[t|t-1] = \boldsymbol{\Sigma}[t] + \mathbf{P}[t]\mathbf{M}[t-1|t-1]\mathbf{P}^T[t] \qquad \text{(A.22)}$$

instead of (A.16), and

$$\mathbf{G}[t] = \mathbf{M}[t|t-1]\mathbf{S}^T[t](\sigma_e^2[t]\mathbf{I}_{S[t]} + \mathbf{S}[t]\mathbf{M}[t|t-1]\mathbf{S}^T[t])^{-1} \qquad \text{(A.23)}$$

instead of (A.20), whereas (A.21) remains the same. These equations are precisely those in steps 4, 5 and 7 of Algorithm 2.

To obtain the rest of the steps, set t to $t-1$ and τ to t in (A.3a) to obtain

$$\hat{\mathbf{f}}[t|t-1] = \mathbf{P}[t]\hat{\mathbf{f}}[t-1|t-1] \qquad \text{(A.24)}$$

which coincides with step 3 of Algorithm 2. Finally, since $\hat{\mathbf{f}}[t|t]$ is the last block vector of $\hat{\bar{\boldsymbol{f}}}[t|t]$, then

$$\begin{aligned}\hat{\mathbf{f}}[t|t] :=& \bar{\mathcal{J}}[t]\hat{\bar{\boldsymbol{f}}}[t|t] \\ =& \bar{\mathcal{J}}[t]\bar{\mathcal{T}}^{-1}[t]\bar{\mathcal{A}}^T[t]\bar{\mathbf{\Sigma}}^{-1}[t]\bar{\boldsymbol{\psi}}[t] \\ =& (\mathbf{I} - \mathbf{G}[t]\mathbf{S}[t])\bar{\mathcal{J}}[t]\bar{\mathcal{T}}'^{-1}[t]\bar{\mathcal{A}}^T[t]\bar{\mathbf{\Sigma}}^{-1}[t]\bar{\boldsymbol{\psi}}[t] \end{aligned} \tag{A.25}$$

where the second equality follows from (A.7) and the third from (A.19). From the definitions of $\bar{\mathcal{A}}[t]$, $\bar{\mathbf{\Sigma}}[t]$ and $\bar{\boldsymbol{\psi}}[t]$, one obtains that

$$\bar{\mathcal{A}}^T[t]\bar{\mathbf{\Sigma}}^{-1}[t]\bar{\boldsymbol{\psi}}[t] = \bar{\mathcal{A}}'^T[t]\bar{\mathbf{\Sigma}}'^{-1}[t]\bar{\boldsymbol{\psi}}'[t] + \tfrac{1}{\sigma_e^2[t]}\boldsymbol{W}^T[t]\mathbf{y}[t]. \tag{A.26}$$

Substituting (A.26) into (A.25) yields

$$\begin{aligned}\hat{\mathbf{f}}[t|t] =& (\mathbf{I} - \mathbf{G}[t]\mathbf{S}[t])\bar{\mathcal{J}}[t]\bar{\mathcal{T}}'^{-1}[t](\bar{\mathcal{A}}'^T[t]\bar{\mathbf{\Sigma}}'^{-1}[t]\bar{\boldsymbol{\psi}}'[t] + \tfrac{1}{\sigma_e^2[t]}\boldsymbol{W}^T[t]\mathbf{y}[t]) \\ =& (\mathbf{I} - \mathbf{G}[t]\mathbf{S}[t])(\hat{\mathbf{f}}[t|t-1] + \tfrac{1}{\sigma_e^2[t]}\mathbf{M}[t|t-1]\mathbf{S}^T[t]\mathbf{y}[t]) \\ =& \hat{\mathbf{f}}[t|t-1] + \mathbf{G}[t](\mathbf{y}[t] - \mathbf{S}[t]\hat{\mathbf{f}}[t|t-1]) \end{aligned} \tag{A.27}$$

where the second equality follows from (A.10), $\hat{\mathbf{f}}[t|t-1] = \bar{\mathcal{J}}[t]\hat{\bar{\boldsymbol{f}}}[t|t-1]$ and (A.11); whereas the third follows from

$$(\mathbf{I}_N - \mathbf{G}[t]\mathbf{S}[t])\mathbf{M}[t|t-1]\mathbf{S}^T[t] = \sigma_e^2[t]\mathbf{G}[t] \tag{A.28}$$

which results from rearranging the terms in (A.23). Noting that expression (A.27) coincides with step 6 of Algorithm 2 concludes the proof.

Appendix B

Kernel kriged Kalman filter

B.1 Proof of Th. 2

The process of developing the desired online algorithm involves two steps. The goal of the first step will be to express (5.13) in a form amenable to a low-complexity solver. The optimization problem in (5.13) is jointly convex in $\mathbf{f}_\nu[\tau]$ and $\mathbf{f}_\chi[\tau]$ for all τ. The optimality conditions suggest taking the derivative in (5.13) with respect to each component and setting it to zero. Specifically, the optimality condition for $\mathbf{f}_\nu[\tau]$ suggests

$$\begin{aligned}\mathbf{f}_\nu[\tau] =&\mathbf{K}_\nu[\tau]\mathbf{S}^T[\tau](\bar{\mathbf{K}}_\nu[\tau]+\mu_2 S[\tau]\boldsymbol{I}_{S[\tau]})^{-1}\times \\ &(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])\end{aligned} \tag{B.1}$$

where $\bar{\mathbf{K}}_\nu[\tau]=\mathbf{S}[\tau]\mathbf{K}_\nu[\tau]\mathbf{S}^T[\tau]$. Notice the $\bar{\cdot}$ notation indicates $S[\tau]\times S[\tau]$ matrices and $S[\tau]\times 1$ vectors. At this point we will substitute (B.1) to (5.13) to obtain an optimization problem that does not contain $\mathbf{f}_\nu[\tau]$ for $\tau=1,\ldots,t$ and is amenable to an online solver. To

that end the first term of (5.13) can be rewritten using (B.1) for each τ as follows

$$\begin{aligned}
&\tfrac{1}{S[\tau]}\|\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau]-\mathbf{S}[\tau]\mathbf{f}_\nu[\tau]\|^2\\
=&\tfrac{1}{S[\tau]}\|\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau]-\bar{\mathbf{K}}_\nu[\tau](\bar{\mathbf{K}}_\nu[\tau]+\mu_2 S[\tau]\mathbf{I}_{S[\tau]})^{-1}(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])\|^2\\
=&\tfrac{1}{S[\tau]}\|\Big(\mathbf{I}_{S[\tau]}-\bar{\mathbf{K}}_\nu[\tau](\bar{\mathbf{K}}_\nu[\tau]+\mu_2 S[\tau]\mathbf{I}_{S[\tau]})^{-1}\Big)(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])\|^2\\
=&\tfrac{1}{S[\tau]}\|(\tfrac{1}{\mu_2 S[\tau]}\bar{\mathbf{K}}_\nu[\tau]+\mathbf{I}_{S[\tau]})^{-1}(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])\|^2\\
=&(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])^T(\tfrac{1}{\mu_2}\bar{\mathbf{K}}_\nu[\tau]+S[\tau]\boldsymbol{I}_{S[\tau]})^{-\top}S[\tau]\boldsymbol{I}_{S[\tau]}(\tfrac{1}{\mu_2}\bar{\mathbf{K}}_\nu[\tau]+S[\tau]\boldsymbol{I}_{S[\tau]})^{-1}(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])
\end{aligned}\tag{B.2}$$

where the third equation follows from the matrix inversion lemma that establishes that

$$\begin{aligned}
&\Big(\mathbf{I}_{S[\tau]}-\bar{\mathbf{K}}_\nu[\tau](\bar{\mathbf{K}}_\nu[\tau]+\mu_2 S[\tau]\mathbf{I}_{S[\tau]})^{-1}\Big)^{-1}\\
=&\mathbf{I}_{S[\tau]}+\bar{\mathbf{K}}_\nu[\tau](\bar{\mathbf{K}}_\nu[\tau]+\mu_2 S[\tau]\mathbf{I}_{S[\tau]}-\bar{\mathbf{K}}_\nu[\tau])^{-1}\\
=&\mathbf{I}_{S[\tau]}+\tfrac{1}{\mu_2 S[\tau]}\bar{\mathbf{K}}_\nu[\tau].
\end{aligned}\tag{B.3}$$

Next, the third term of (5.13) will be rewritten using (B.1) for each τ as follows

$$\begin{aligned}
&\mu_2\|\mathbf{f}_\nu[\tau]\|^2_{\mathbf{K}_\nu[\tau]}\\
=&\mu_2\|\mathbf{K}_\nu[\tau]\mathbf{S}^T[\tau](\bar{\mathbf{K}}_\nu[\tau]+\mu_2 S[\tau]\boldsymbol{I}_{S[\tau]})^{-1}(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])\|^2_{\mathbf{K}_\nu[\tau]}\\
=&\mu_2(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])^T(\bar{\mathbf{K}}_\nu[\tau]+\mu_2 S[\tau]\boldsymbol{I}_{S[\tau]})^{-\top}\mathbf{S}[\tau]\mathbf{K}_\nu^T[\tau]\mathbf{K}_\nu^{-1}[\tau]\times\\
&\mathbf{K}_\nu[\tau]\mathbf{S}^T[\tau](\bar{\mathbf{K}}_\nu[\tau]+\mu_2 S[\tau]\boldsymbol{I}_{S[\tau]})^{-1}(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])\\
=&(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])^T(\tfrac{1}{\mu_2}\bar{\mathbf{K}}_\nu[\tau]+S[\tau]\boldsymbol{I}_{S[\tau]})^{-\top}\tfrac{1}{\mu_2}\bar{\mathbf{K}}_\nu[\tau](\tfrac{1}{\mu_2}\bar{\mathbf{K}}_\nu[\tau]+S[\tau]\boldsymbol{I}_{S[\tau]})^{-1}(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])
\end{aligned}\tag{B.4}$$

where the third equation follows from the definition of $\bar{\mathbf{K}}_\nu[\tau]$. Next, the summation of the terms in (B.2) and (B.4) yields

$$\begin{aligned}
&\tfrac{1}{S[\tau]}\|\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau]-\mathbf{S}[\tau]\mathbf{f}_\nu[\tau]\|^2+\mu_2\|\mathbf{f}_\nu[\tau]\|^2_{\mathbf{K}_\nu[\tau]}\\
=&(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])^T(\tfrac{1}{\mu_2}\bar{\mathbf{K}}_\nu[\tau]+S[\tau]\boldsymbol{I}_{S[\tau]})^{-\top}\times\\
&(\tfrac{1}{\mu_2}\bar{\mathbf{K}}_\nu[\tau]+S[\tau]\boldsymbol{I}_{S[\tau]})(\tfrac{1}{\mu_2}\bar{\mathbf{K}}_\nu[\tau]+S[\tau]\boldsymbol{I}_{S[\tau]})^{-1}(\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau])\\
=&\|\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau]\|^2_{\bar{\mathbf{K}}_\chi[\tau]}
\end{aligned}\tag{B.5}$$

where $\bar{\mathbf{K}}_\chi[\tau]=\frac{1}{\mu_2}\bar{\mathbf{K}}_\nu[\tau]+S[\tau]\mathbf{I}_{S[\tau]}$. Notice that the analysis so far carries for every τ. Using the least-squares term in (B.5), the optimization problem in (5.13) for the variables $\{\mathbf{f}_\chi[\tau]\}_{\tau=1}^t$ can be written as follows

$$\underset{\{\mathbf{f}_\chi[\tau]\}_{\tau=1}^t}{\arg\min}\sum_{\tau=1}^t\|\mathbf{y}[\tau]-\mathbf{S}[\tau]\mathbf{f}_\chi[\tau]\|^2_{\bar{\mathbf{K}}_\chi[\tau]}+\mu_1\sum_{\tau=1}^t\|\mathbf{f}_\chi[\tau]-\ \mathbf{P}[\tau]\mathbf{f}_\chi[\tau-1]\|^2_{\mathbf{K}_\eta[\tau]}\tag{B.6}$$

The sequence of estimates $\{\hat{\mathbf{f}}_\chi[\tau|\tau]\}_{\tau=1}^t$ for (B.6) is known to be found sequentially by the Kalman filter [28, 44]. After substituting $\{\hat{\mathbf{f}}_\chi[\tau|\tau]\}_{\tau=1}^t$ in (B.1) one obtains $\{\hat{\mathbf{f}}_\nu[\tau|\tau]\}_{\tau=1}^t$ which concludes the proof.