Kernel-Based Reconstruction of Graph Signals

Daniel Romero, *Member, IEEE*, Meng Ma, and Georgios B. Giannakis, *Fellow, IEEE*

***Abstract*—A number of applications in engineering, social sciences, physics, and biology involve inference over networks. In this context, graph signals are widely encountered as descriptors of vertex attributes or features in graph-structured data. Estimating such signals in all vertices given noisy observations of their values on a subset of vertices has been extensively analyzed in the literature of signal processing on graphs (SPoG). This paper advocates kernel regression as a framework generalizing popular SPoG modeling and reconstruction and expanding their capabilities. Formulating signal reconstruction as a regression task on reproducing kernel Hilbert spaces of graph signals permeates benefits from statistical learning, offers fresh insights, and allows for estimators that leverage richer forms of prior information than existing alternatives. A number of SPoG notions such as bandlimitedness, graph filters, and the graph Fourier transform are naturally accommodated in the kernel framework. Additionally, this paper capitalizes on the so-called representer theorem to devise simpler versions of existing Tikhonov regularized estimators, and offers a novel probabilistic interpretation of kernel methods on graphs based on graphical models. Motivated by the challenges of selecting the bandwidth parameter in SPoG estimators or the kernel map in kernel-based methods, this paper further proposes two multikernel approaches with complementary strengths. Whereas the first enables estimation of the unknown bandwidth of bandlimited signals, the second allows for efficient graph filter selection. Numerical tests with synthetic as well as real data demonstrate the merits of the proposed methods relative to state-of-the-art alternatives.**

***Index Terms*—Graph signal reconstruction, kernel regression, multi-kernel learning.**

I. Introduction

GRAPH data play a central role in analysis and inference tasks for social, brain, communication, biological, transportation, and sensor networks [1], thanks to their ability to capture relational information. Vertex attributes or features associated with vertices can be interpreted as functions or signals defined on graphs. In social networks, for instance, where a vertex represents a person and an edge corresponds to a friendship relation, such a function may denote e.g. the person's age, location, or rating of a given movie.

Research efforts over the last years are centered on estimating or processing functions on graphs; see e.g. [1]–[6]. Existing approaches rely on the premise that signals exhibit a certain form of parsimony relative to the graph topology. For example, it seems reasonable to estimate a person's age by looking at their friends' age. The present paper deals with a general version of this task, where the goal is to estimate a graph signal given noisy observations on a subset of vertices.

The machine learning community has already looked at SPoG-related issues in the context of semi-supervised learning under the term of transductive regression and classification [6]–[8]. Existing approaches rely on smoothness assumptions to infer functions over graphs using nonparametric methods [2], [3], [6], [9]. Whereas some works consider estimation of real-valued signals [7]–[10], most in this body of literature have focused on estimating binary-valued functions; see e.g. [6]. On the other hand, function estimation has also been investigated recently by the community of signal processing on graphs (SPoG) under the term *signal reconstruction* [11]–[18]. Existing approaches commonly adopt *parametric* estimation tools and rely on *bandlimitedness*, by which the signal of interest is assumed to lie in the span of B eigenvectors of the graph Laplacian or adjacency matrix [12]–[14], [16]–[18]. Different from machine learning works, SPoG research is mainly concerned with estimating real-valued functions.

The present paper cross-pollinates ideas and broadens both machine learning and SPoG perspectives under the unifying framework of kernel-based learning. The first part unveils the implications of adopting this standpoint and demonstrates how it naturally accommodates a number of SPoG concepts and tools. From a high level, this connection (i) brings to bear performance bounds and algorithms from transductive regression [8] and the extensively analyzed general kernel methods (see e.g. [19]); (ii) offers the possibility of reducing the dimension of the optimization problems involved in Tikhonov regularized estimators by invoking the so-called representer theorem [20]; and, (iii) it provides guidelines for systematically selecting parameters in existing signal reconstruction approaches by leveraging the connection with linear minimum mean-square error (LMMSE) estimation via *covariance kernels*.

Further implications of applying kernel methods to graph signal reconstruction are also explored. Specifically, it is shown that the finite dimension of graph signal spaces allows for an insightful proof of the representer theorem which, different from existing proofs relying on functional analysis, solely involves linear algebra arguments. Moreover, an intuitive probabilistic interpretation of graph kernel-based reconstruction methods is introduced based on graphical models. These findings are complemented with a technique to deploy regression with Laplacian kernels in big-data setups.

It is further established that a number of existing signal reconstruction approaches, including the least-squares (LS) estimators for bandlimited signals from [11]–[16]; the Tikhonov

Manuscript received May 23, 2016; revised September 12, 2016; accepted October 4, 2016. Date of publication October 26, 2016; date of current version November 28, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yuichi Tanaka. This work was supported in part by the ARO under Grant W911NF-15-1-0492 and in part by the NSF under Grant 1343248, Grant 1442686, and Grant 1514056. This paper was presented in part at the 2016 IEEE Statistical Signal Processing Workshop, Palma de Mallorca, Spain.

The authors are with the Department of Electrical and Computer Engineering and Digital Technology Center, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: dromero@umn.edu; maxxx971@umn.edu; georgios@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2016.2620116

regularized estimators from [4], [12], [21] and [22, eq. (27)]; and the maximum a posteriori estimator in [13], can be viewed as kernel methods on *reproducing kernel Hilbert spaces* (RKHSs) of graph signals. Popular notions in SPoG such as graph filters, the graph Fourier transform, and bandlimited signals can also be accommodated under the kernel framework. First, it is seen that the so-called graph filters [4] are essentially kernel smoothers [23]. Second, bandlimited kernels are introduced to accommodate estimation of bandlimited signals. Third, the connection between the so-called graph Fourier transform [4] (see [5], [15] for a related definition) and Laplacian kernels [2], [3] is delineated. Relative to methods relying on the bandlimited property (see e.g. [11]–[17]), kernel methods offer increased flexibility in leveraging prior information about the graph Fourier transform of the estimated signal.

The second part of the paper pertains to the challenge of model selection. On the one hand, a number of reconstruction schemes in SPoG [12]–[15], [17] require knowledge of the signal bandwidth, which is typically unknown [11], [16]. Existing approaches for determining this bandwidth rely solely on the set of sampled vertices, disregarding the observations [11], [16]. On the other hand, existing kernel-based approaches [1, Ch. 8] necessitate proper kernel selection, which is computationally inefficient to carry through cross-validation.

The present paper addresses both issues by means of two multi-kernel learning (MKL) techniques with complementary strengths. Existing MKL methods on graphs are confined to estimating binary-valued signals [24]–[26]. This paper, on the other hand, is concerned with MKL for real-valued graph signal reconstruction. The algorithms here optimally combine the kernels in a given dictionary and simultaneously estimate the graph signal by solving a single optimization problem.

The rest of the paper is structured as follows. Section II formulates the problem of graph signal reconstruction. Section III presents kernel-based learning as an encompassing framework for graph signal reconstruction, and explores the implications of adopting such a standpoint. A novel proof of the representer theorem is provided, and its role in reducing the dimensionality of kernel-based learning problems is elucidated. Section IV deals with the selection of suitable kernels. First, two families of topology-based kernels are considered: (i) Laplacian kernels, which are reviewed and interpreted from an SPoG perspective; and (ii) bandlimited kernels, which are proposed to reconstruct signals adhering to the popular bandlimited model. Second, a probabilistic interpretation of kernel-based methods is provided through the notion of vertex-covariance kernels, which constitute an appealing alternative in setups with unknown graph topology. Section V shows that graph filtering can be thought of as a form of kernel smoothing. Section VI presents two MKL schemes listed as Algorithms 1 and 2. The first pursues estimates that optimally combine signals belonging to different RKHSs from a given dictionary. The second finds an estimate within an RKHS constructed by optimally combining kernels of a given dictionary. As a byproduct, the first algorithm provides a method to estimate the bandwidth of a bandlimited graph signal. Section VII complements analytical findings with numerical tests by comparing with competing alternatives via synthetic- and real-data experiments. Finally, concluding remarks are highlighted in Section VIII.

TABLE I
SYMBOL DEFINITIONS

N	Number of vertices
$\boldsymbol{W}$	$N \times N$ weighted adjacency matrix
$\boldsymbol{D} := \text{diag}\{\boldsymbol{W}\mathbf{1}\}$	$N \times N$ diagonal degree matrix
$\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{W}$	$N \times N$ Laplacian matrix
S	Number of sampled vertices
$\mathcal{S} := \{n_1, \ldots, n_S\}$	Set of indices of sampled vertices
$\boldsymbol{f} := [f(v_1), \ldots, f(v_N)]^T$	$N \times 1$ vector with values of signal f at all vertices
$\bar{\boldsymbol{f}} := [f(v_{n_1}), \ldots, f(v_{n_S})]^T$	$S \times 1$ vector with values of signal f at sampled vertices
$\kappa(v_n, v_{n'})$	Kernel evaluated at vertices v_n and $v_{n'}$
$\boldsymbol{K} := (\kappa(v_n, v_{n'}))_{n,n'=1}^{N}$	$N \times N$ kernel matrix over all vertices
$\bar{\boldsymbol{K}} := (\kappa(v_{n_s}, v_{n_{s'}}))_{s,s'=1}^{S}$	$S \times S$ kernel matrix over sampled vertices
$\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_N]^T$	$N \times 1$ vector with RKHS expansion coefficients over all vertices (see (3))
$\bar{\boldsymbol{\alpha}} := [\bar{\alpha}_1, \ldots, \bar{\alpha}_S]^T$	$S \times 1$ vector with RKHS expansion coefficients over sampled vertices (see (10))

Notation: $(\cdot)_N$ denotes the remainder of integer division by N, $\delta[\cdot]$ the Kronecker delta, and $\mathcal{I}[C]$ the indicator of condition C, which equals 1 if C is satisfied and 0 otherwise. Scalars are denoted by lowercase letters, vectors by bold lowercase, and matrices by bold uppercase. The (i,j)-th entry of matrix $\boldsymbol{A}$ is $(\boldsymbol{A})_{i,j}$. Notation $||\cdot||_2$ and $\text{Tr}(\cdot)$ respectively represent Euclidean norm and trace; $\boldsymbol{I}_N$ denotes the $N \times N$ identity matrix; $\boldsymbol{i}_n$ is the n-th column of $\boldsymbol{I}_N$, while $\mathbf{0}$ ($\mathbf{1}$) is a vector of appropriate dimension with all zeros (ones). The span of the columns of $\boldsymbol{A}$ is denoted by $\mathcal{R}\{\boldsymbol{A}\}$, whereas $\boldsymbol{A} \succ \boldsymbol{B}$ (resp. $\boldsymbol{A} \succeq \boldsymbol{B}$) means that $\boldsymbol{A} - \boldsymbol{B}$ is positive definite (resp. semi-definite). Superscripts ^{T} and † respectively stand for transpose and pseudo-inverse, whereas $\mathbb{E}$ denotes expectation. Table I lists the most common symbols for reference purposes.

II. PROBLEM STATEMENT

A graph is a tuple $\mathcal{G} := (\mathcal{V}, w)$, where $\mathcal{V} := \{v_1, \ldots, v_N\}$ denotes the vertex set and $w : \mathcal{V} \times \mathcal{V} \to [0, +\infty)$ assigns a weight to each vertex pair. For simplicity, it is assumed that $w(v,v) = 0\, \forall v \in \mathcal{V}$. This paper focuses on undirected graphs, for which $w(v,v') = w(v',v)\ \forall v, v' \in \mathcal{V}$. A graph is said to be unweighted if $w(v,v')$ is either 0 or 1. The edge set $\mathcal{E}$ is the support of w, i.e., $\mathcal{E} := \{(v,v') \in \mathcal{V} \times \mathcal{V} : w(v,v') \neq 0\}$. Two vertices v and v' are adjacent, connected, or neighbors if $(v,v') \in \mathcal{E}$. The n-th neighborhood $\mathcal{N}_n$ is the set of neighbors of v_n, namely, $\mathcal{N}_n := \{v \in \mathcal{V} : (v, v_n) \in \mathcal{E}\}$. The information in w is compactly represented by the $N \times N$ weighted adjacency matrix $\boldsymbol{W}$, whose (n,n')-th entry is $w(v_n, v_{n'})$; the $N \times N$ diagonal *degree* matrix $\boldsymbol{D}$, whose (n,n)-th entry is $\sum_{n'=1}^{N} w(v_n, v_{n'})$; and the *Laplacian* matrix $\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{W}$, which is symmetric and positive semidefinite [1, Ch. 2]. The latter is sometimes replaced with its normalized version $\boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$, whose eigenvalues are confined to the interval $[0, 2]$.

A real-valued function (or signal) on a graph is a map $f_0 : \mathcal{V} \to \mathbb{R}$. As mentioned in Section I, the value $f_0(v)$ represents an attribute or feature of $v \in \mathcal{V}$, such as age, political alignment,

or annual income of a person in a social network. Signal values at all vertices are collected in $\boldsymbol{f}_0 := [f_0(v_1), \ldots, f_0(v_N)]^T$.

Suppose that a collection of noisy samples (or observations) $\{y_s = f_0(v_{n_s}) + e_s\}_{s=1}^S$, is available, where e_s models noise and $\mathcal{S} := \{n_1, \ldots, n_S\}$ contains the indices $1 \leq n_1 < \cdots < n_S \leq N$ of the sampled vertices. In a social network, this may be the case if a subset of persons have been surveyed about the attribute of interest (e.g. political alignment). Given $\{(n_s, y_s)\}_{s=1}^S$ and assuming knowledge of $\mathcal{G}$, the goal is to estimate f_0. This will provide estimates of $f_0(v)$ both at observed and unobserved vertices $v \in \mathcal{V}$. By defining $\bar{\boldsymbol{y}} := [y_1, \ldots, y_S]^T$, the observation model can be summarized as

$$\bar{\boldsymbol{y}} = \boldsymbol{\Phi} \boldsymbol{f}_0 + \bar{\boldsymbol{e}} \tag{1}$$

where $\bar{\boldsymbol{e}} := [e_1, \ldots, e_S]^T$ and $\boldsymbol{\Phi}$ is an $S \times N$ binary matrix with entries (s, n_s), $s = 1, \ldots, S$ set to one, and the rest set to zero.

III. Unifying the Reconstruction of Graph Signals

Kernel methods constitute the "workhorse" of machine learning for nonlinear function estimation [19]. Their popularity can be ascribed to their simplicity, flexibility, and good performance. This section presents kernel regression as a novel unifying framework for graph signal reconstruction and explores the implications of the so-called representer theorem.

Kernel regression seeks an estimate of f_0 in an RKHS $\mathcal{H}$, which is the space of functions $f : \mathcal{V} \to \mathbb{R}$ defined as

$$\mathcal{H} := \left\{ f : f(v) = \sum_{n=1}^{N} \alpha_n \kappa(v, v_n),\ \alpha_n \in \mathbb{R} \right\}. \tag{2}$$

The *kernel map* $\kappa : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is any function defining a symmetric and positive semidefinite $N \times N$ matrix with entries $(\boldsymbol{K})_{n,n'} := \kappa(v_n, v_{n'}),\ \forall n, n'$ [27]. Intuitively, $\kappa(v, v')$ in (2) is a basis function measuring similarity between the values of f_0 at v and v'. For instance, if a feature vector $\boldsymbol{x}_n \in \mathbb{R}^D$ containing attributes of the entity (e.g. a person in a social network) represented by v_n is available for $n = 1, \ldots, N$, one can employ the popular Gaussian kernel $\kappa(v_n, v_{n'}) = \exp\{-||\boldsymbol{x}_n - \boldsymbol{x}_{n'}||^2/\sigma^2\}$, where $\sigma^2 > 0$ is a user-selected parameter [19]. When such feature vectors $\boldsymbol{x}_n$ are not available, the graph topology can be leveraged to construct graph kernels as detailed in Section IV.

RKHSs of graph signals are finite dimensional since the expansion in (2) involves a finite number of terms. This property sets them apart from more general RKHSs of functions defined over infinite sets such as $\mathbb{R}^p$, which are typically infinite-dimensional since they are constructed using infinite expansions. From (2), it follows that any signal in $\mathcal{H}$ can be expressed as:

$$\boldsymbol{f} := [f(v_1), \ldots, f(v_N)]^T = \boldsymbol{K}\boldsymbol{\alpha} \tag{3}$$

for some $N \times 1$ vector $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_N]^T$. Given two functions $f(v) := \sum_{n=1}^N \alpha_n \kappa(v, v_n)$ and $f'(v) := \sum_{n=1}^N \alpha'_n \kappa(v, v_n)$, their RKHS inner product is defined as[1]

$$\langle f, f' \rangle_{\mathcal{H}} := \sum_{n=1}^{N} \sum_{n'=1}^{N} \alpha_n \alpha'_{n'} \kappa(v_n, v_{n'}) = \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}' \tag{4}$$

where $\boldsymbol{\alpha}' := [\alpha'_1, \ldots, \alpha'_N]^T$. The RKHS norm is defined by

$$||f||_{\mathcal{H}}^2 := \langle f, f \rangle_{\mathcal{H}} = \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} \tag{5}$$

and will be used as a regularizer to control overfitting. As a special case, setting $\boldsymbol{K} = \boldsymbol{I}_N$ recovers the standard inner product $\langle f, f' \rangle_{\mathcal{H}} = \boldsymbol{f}^T \boldsymbol{f}'$, and the Euclidean norm $||f||_{\mathcal{H}}^2 = ||\boldsymbol{f}||_2^2$. Note that, when $\boldsymbol{K} \succ \boldsymbol{0}$, the set of functions of the form (3) equals $\mathbb{R}^N$. Thus, two RKHSs with strictly positive definite kernel matrices contain the same functions. They differ only in their RKHS inner products and norms. Interestingly, this observation establishes that any positive definite kernel is *universal* [28] for graph signal reconstruction.

The term *reproducing kernel* stems from the reproducing property. Let $\kappa(\cdot, v_{n_0})$ denote the map $v \mapsto \kappa(v, v_{n_0})$, where $n_0 \in \{1, \ldots, N\}$. Using (4), the reproducing property can be expressed as $\langle \kappa(\cdot, v_{n_0}), \kappa(\cdot, v_{n'_0}) \rangle_{\mathcal{H}} = \boldsymbol{i}_{n_0}^T \boldsymbol{K} \boldsymbol{i}_{n'_0} = \kappa(v_{n_0}, v_{n'_0})$. Such a property is of paramount importance when dealing with an RKHS of functions defined on *infinite* spaces (thus excluding RKHSs of graph signals) since it offers an efficient alternative to the costly multidimensional integration required by inner products such as $\langle f_1, f_2 \rangle_{L^2} := \int_{\mathcal{X}} f_1(\boldsymbol{x}) f_2(\boldsymbol{x}) d\boldsymbol{x}$. Specifically, due to the linearity of inner products and the fact that all signals in such RKHSs are the superposition of basis functions $\kappa(\cdot, \boldsymbol{x})$, the reproducing property $\langle \kappa(\cdot, \boldsymbol{x}), \kappa(\cdot, \boldsymbol{x}') \rangle_{\mathcal{H}} = \kappa(\boldsymbol{x}, \boldsymbol{x}')$ enables inner product computation just by evaluating κ.

Given $\{(n_s, y_s)\}_{s=1}^S$, RKHS-based function estimators are obtained by solving functional minimization problems formulated as (see also e.g. [19], [27], [29])

$$\hat{f}_0 := \arg\min_{f \in \mathcal{H}} \mathcal{L}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{f}}) + \mu \Omega(||f||_{\mathcal{H}}) \tag{6}$$

where the loss $\mathcal{L}$ measures how the values of the estimated function f at the observed vertices $\bar{\boldsymbol{v}} := [v_{n_1}, \ldots, v_{n_S}]^T \in \mathcal{V}^S$, collected in $\bar{\boldsymbol{f}} := [f(v_{n_1}), \ldots, f(v_{n_S})]^T = \boldsymbol{\Phi}\boldsymbol{f}$, deviate from the data $\bar{\boldsymbol{y}}$. The so-called square loss $\mathcal{L}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{f}}) := (1/S) \sum_{s=1}^S [y_s - f(v_{n_s})]^2$ constitutes a popular choice for $\mathcal{L}$. The increasing function Ω is used to promote smoothness with typical choices including $\Omega(\zeta) = \zeta$ and $\Omega(\zeta) = \zeta^2$. The regularization parameter $\mu > 0$ controls overfitting.

Substituting (3) and (5) into (6) shows that $\hat{\boldsymbol{f}}_0$ can be found as

$$\boxed{\begin{aligned} &\hat{\boldsymbol{f}}_0 = \boldsymbol{K}\hat{\boldsymbol{\alpha}} \\ &\hat{\boldsymbol{\alpha}} := \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \mathcal{L}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{y}}, \boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\alpha}) + \mu \Omega((\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha})^{1/2}). \end{aligned}} \tag{7}$$

An alternative form of (7) that will be frequently used in the sequel results upon noting that $\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{K}^\dagger \boldsymbol{K} \boldsymbol{\alpha} = \boldsymbol{f}^T \boldsymbol{K}^\dagger \boldsymbol{f}$. Thus, one can rewrite (7) as

$$\hat{\boldsymbol{f}}_0 := \arg \min_{\boldsymbol{f} \in \mathcal{R}\{\boldsymbol{K}\}} \mathcal{L}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{y}}, \boldsymbol{\Phi}\boldsymbol{f}) + \mu \Omega((\boldsymbol{f}^T \boldsymbol{K}^\dagger \boldsymbol{f})^{1/2}). \tag{8}$$

[1] Whereas f denotes a *function*, symbol $f(v)$ represents the *scalar* resulting from evaluating f at vertex v.

If $\boldsymbol{K} \succ \boldsymbol{0}$, the constraint $\boldsymbol{f} \in \mathcal{R}\{\boldsymbol{K}\}$ can be omitted and $\boldsymbol{K}^{\dagger}$ can be replaced with $\boldsymbol{K}^{-1}$. If $\boldsymbol{K}$ contains null eigenvalues, it is customary to remove the constraint by replacing $\boldsymbol{K}$ (or $\boldsymbol{K}^{\dagger}$) with a perturbed version $\boldsymbol{K} + \epsilon \boldsymbol{I}$ (respectively $\boldsymbol{K}^{\dagger} + \epsilon \boldsymbol{I}$), where $\epsilon > 0$ is a small constant. Expression (8) shows that kernel regression unifies and subsumes the Tikhonov-regularized graph signal reconstruction schemes in [4], [12], [21] and [22, eq. (27)] by properly selecting $\boldsymbol{K}$, $\mathcal{L}$, and Ω (see Section IV).

Although graph signals can be reconstructed from (7), such an approach involves optimizing over N variables. It is shown next that a solution can be obtained by solving an optimization problem in S variables, where typically $S \ll N$.

The representer theorem [20], [27] plays an instrumental role in the non-graph setting of infinite-dimensional $\mathcal{H}$, where (6) cannot be directly solved. This theorem enables a solver by providing a finite parameterization of the function $\hat{f}_0$ in (6). On the other hand, when $\mathcal{H}$ comprises graph signals, (6) is inherently finite dimensional and can be solved directly. However, the representer theorem can still be beneficial to reduce the dimension of the optimization in (7).

Theorem 1 (Representer theorem): The solution to the functional minimization in (6) can be expressed as

$$\hat{f}_0(v) = \sum_{s=1}^{S} \bar{\alpha}_s \kappa(v, v_{n_s}) \tag{9}$$

for some $\bar{\alpha}_s \in \mathbb{R}$, $s = 1, \ldots, S$.

The conventional proof for the representer theorem involves tools from functional analysis [27]. However, when $\mathcal{H}$ comprises functions defined on finite spaces, such us graph signals, an insightful proof can be obtained relying solely on linear algebra arguments (see Appendix A).

Since the solution $\hat{f}_0$ of (6) lies in $\mathcal{H}$, it can always be expressed as $\hat{f}_0(v) = \sum_{n=1}^{N} \alpha_n \kappa(v, v_n)$ for some $\{\alpha_n\}_{n=1}^{N}$. Theorem 1 asserts that the terms corresponding to unobserved vertices v_n, $n \notin \mathcal{S}$, play no role in the kernel expansion of the estimate; that is, $\alpha_n = 0$, $\forall n \notin \mathcal{S}$. Thus, whereas (7) requires optimization over N variables, Theorem 1 establishes that a solution can be found by solving a problem in S variables, where typically $S \ll N$. Clearly, this conclusion carries over to the signal reconstruction schemes in [4], [12], [21] and [22, eq. (27)], since they constitute special instances of kernel regression. The fact that the number of parameters to be estimated after applying Theorem 1 depends on (in fact, equals) the number of samples S justifies why $\hat{f}_0$ in (6) is referred to as a nonparametric estimate.

Theorem 1 shows the form of $\hat{f}_0$ but does not provide the optimal $\{\bar{\alpha}_s\}_{s=1}^{S}$, which are found after substituting (9) into (6) and solving the resulting optimization problem with respect to these coefficients. To this end, let $\bar{\boldsymbol{\alpha}} := [\bar{\alpha}_1, \ldots, \bar{\alpha}_S]^T$, and write $\boldsymbol{\alpha} = \boldsymbol{\Phi}^T \bar{\boldsymbol{\alpha}}$ to deduce that

$$\hat{\boldsymbol{f}}_0 = \boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{K}\boldsymbol{\Phi}^T \bar{\boldsymbol{\alpha}}. \tag{10}$$

From (7) and (10), the optimal $\bar{\boldsymbol{\alpha}}$ can be found as

$$\hat{\bar{\boldsymbol{\alpha}}} := \underset{\bar{\boldsymbol{\alpha}} \in \mathbb{R}^S}{\arg\min}\, \mathcal{L}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{K}}\bar{\boldsymbol{\alpha}}) + \mu \Omega((\bar{\boldsymbol{\alpha}}^T \bar{\boldsymbol{K}} \bar{\boldsymbol{\alpha}})^{1/2}) \tag{11}$$

where $\bar{\boldsymbol{K}} := \boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\Phi}^T$.

Example 1 (kernel ridge regression): For $\mathcal{L}$ chosen as the square loss and $\Omega(\zeta) = \zeta^2$, the function $\hat{f}_0$ in (6) is referred to as the kernel ridge regression estimate [19]. If $\bar{\boldsymbol{K}}$ is full rank, this estimate is given by $\hat{\boldsymbol{f}}_{\text{RR}} = \boldsymbol{K}\boldsymbol{\Phi}^T \hat{\bar{\boldsymbol{\alpha}}}$, where

$$\hat{\bar{\boldsymbol{\alpha}}} := \underset{\bar{\boldsymbol{\alpha}} \in \mathbb{R}^S}{\arg\min}\, \frac{1}{S} \left\| \bar{\boldsymbol{y}} - \bar{\boldsymbol{K}}\bar{\boldsymbol{\alpha}} \right\|^2 + \mu \bar{\boldsymbol{\alpha}}^T \bar{\boldsymbol{K}} \bar{\boldsymbol{\alpha}} \tag{12a}$$

$$= (\bar{\boldsymbol{K}} + \mu S \boldsymbol{I}_S)^{-1} \bar{\boldsymbol{y}}. \tag{12b}$$

Therefore, $\hat{\boldsymbol{f}}_{\text{RR}}$ can be expressed as

$$\hat{\boldsymbol{f}}_{\text{RR}} = \boldsymbol{K}\boldsymbol{\Phi}^T (\bar{\boldsymbol{K}} + \mu S \boldsymbol{I}_S)^{-1} \bar{\boldsymbol{y}}. \tag{13}$$

As seen in the next section, (13) generalizes a number of existing signal reconstructors upon properly selecting $\boldsymbol{K}$. Thus, Theorem 1 can also be used to simplify Tikhonov-regularized estimators such as the one in [12, eq. (15)]. To see this, just note that (13) inverts an $S \times S$ matrix, whereas [12, eq. (16)] entails the inversion of an $N \times N$ matrix.

Example 2 (support vector regression): If $\mathcal{L}$ equals the so-called ϵ-insensitive loss $\mathcal{L}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{f}}) := (1/S) \sum_{s=1}^{S} \max(0, |y_s - f(v_{n_s})| - \epsilon)$ and $\Omega(\zeta) = \zeta^2$, then (6) constitutes a support vector machine for regression (see e.g. [19, Ch. 1]).

So far, the kernel matrix $\boldsymbol{K}$ was regarded as given. In practice, kernel selection must account for prior information about f_0 and the graph topology. The next section guides this selection by describing different families of kernels.

IV. Graph Kernels for Signal Reconstruction

When estimating functions on graphs, conventional kernels such as the Gaussian kernel mentioned in Section III cannot be applied because the underlying set where graph signals are defined is not a metric space. Indeed, no vertex addition $v_n + v_{n'}$, scaling βv_n, or norm $||v_n||$ can be naturally defined on $\mathcal{V}$. An alternative is to embed $\mathcal{V}$ into Euclidean space via a feature map $\phi : \mathcal{V} \to \mathbb{R}^D$, and apply a conventional kernel afterwards. However, for a given graph, it is generally unclear how to explicitly design such a mapping or select D, which motivates the adoption of graph kernels [3].

The rest of this section elaborates on various classes of graph kernels, namely (i) topology-based kernels, which leverage the graph topology to promote "smooth" estimates; and (ii) vertex-covariance kernels, which enable kernel-based estimation even in cases where the topology is unknown but historical data are available.

A. Topology-Based Kernels

A key assumption common to many reconstruction methods is that f_0 evolves smoothly over the graph. Informally, this means that the values taken by f_0 are similar for neighboring vertices or for vertices lying close in geodesic distance [1]. This section describes how graph kernels capturing the graph topology can be built upon this notion.

1) Laplacian Kernels: The term Laplacian kernel comprises a wide family of kernels obtained by applying a certain function to the Laplacian matrix $\boldsymbol{L}$. From a theoretical perspective, Laplacian kernels are well motivated since they constitute the

graph counterpart of the so-called translation-invariant kernels in Euclidean spaces [3]. This section reviews Laplacian kernels from a signal processing standpoint, provides novel insights in terms of interpolating signals, and highlights their versatility in capturing prior information about the graph Fourier transform of the estimated signal.

Let $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$ denote the eigenvalues of the graph Laplacian matrix $\boldsymbol{L}$, and consider the eigendecomposition $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where $\boldsymbol{\Lambda} := \text{diag}\,\{\lambda_1, \ldots, \lambda_N\}$. A Laplacian kernel is a kernel map κ generating entries of the matrix

$$\boldsymbol{K} := r^{\dagger}(\boldsymbol{L}) := \boldsymbol{U} r^{\dagger}(\boldsymbol{\Lambda})\boldsymbol{U}^T \tag{14}$$

where $r(\boldsymbol{\Lambda})$ is the result of applying the user-selected nonnegative map $r : \mathbb{R} \to \mathbb{R}_+$ to the diagonal entries of $\boldsymbol{\Lambda}$. For reasons that will become clear, the map $r(\lambda)$ is typically increasing in λ. Common choices include the diffusion kernel $r(\lambda) = \exp\{\sigma^2\lambda/2\}$ [2], and the p-step random walk kernel $r(\lambda) = (a - \lambda)^{-p}$, $a \geq 2$ [3]. Laplacian regularization [3], [4], [9], [30], [31] is effected by setting $r(\lambda) = 1 + \sigma^2\lambda$ with σ^2 sufficiently large.

Observe that obtaining $\boldsymbol{K}$ generally requires an eigendecomposition of $\boldsymbol{L}$, which is computationally challenging for large graphs ($N \gg$). Two techniques to reduce complexity in these *big data* scenarios are proposed in Appendix B.

At this point, it is prudent to offer interpretations and insights into the operation of Laplacian kernels. Towards this objective, note first that the regularizer from (8) is an increasing function of

$$\boldsymbol{f}^T\boldsymbol{K}^{\dagger}\boldsymbol{f} = \boldsymbol{f}^T\boldsymbol{U}r(\boldsymbol{\Lambda})\boldsymbol{U}^T\boldsymbol{f} = \tilde{\boldsymbol{f}}^T r(\boldsymbol{\Lambda})\tilde{\boldsymbol{f}} = \sum_{n=1}^{N} r(\lambda_n)|\tilde{f}_n|^2 \tag{15}$$

where $\tilde{\boldsymbol{f}} := \boldsymbol{U}^T\boldsymbol{f} := [\tilde{f}_1, \ldots, \tilde{f}_N]^T$ comprises the projections of $\boldsymbol{f}$ onto the eigenvectors of $\boldsymbol{L}$ and is referred to as the *graph Fourier transform* of $\boldsymbol{f}$ in the SPoG parlance [4]. Before interpreting (15), it is worth elucidating that this term was coined because the role played by the eigenvectors of $\boldsymbol{L}$ in SPoG resembles that played by complex exponentials in conventional signal processing. Specifically, since $\boldsymbol{U} := [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N]$ is orthogonal, one can decompose $\boldsymbol{f}$ as

$$\boldsymbol{f} = \boldsymbol{U}\tilde{\boldsymbol{f}} = \sum_{n=1}^{N} \tilde{f}_n \boldsymbol{u}_n. \tag{16}$$

Because vectors $\{\boldsymbol{u}_n\}_{n=1}^N$, or more precisely their signal counterparts[2] $\{u_n\}_{n=1}^N$, are eigensignals of the so-called *graph shift operator* $\boldsymbol{u} \mapsto \boldsymbol{L}\boldsymbol{u}$, (16) resembles the classical Fourier transform in the sense that it expresses a signal as a superposition of eigensignals of a Laplacian operator [4].

Recalling from Section II that $w(v_n, v_{n'})$ denotes the weight of the edge between v_n and $v_{n'}$, a functional capturing the smoothness notion described at the beginning of Section IV-A is

$$\partial f := \frac{1}{2}\sum_{n=1}^{N}\sum_{n' \in \mathcal{N}_n} w(v_n, v_{n'})[f(v_n) - f(v_{n'})]^2 = \boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f} \tag{17}$$

where the last equality follows after some algebra from the definition of $\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{W}$. Eigensignals in conventional signal processing for time series, i.e. complex exponentials, are typically sorted according to their frequency, which can be thought of as a proxy measure for their smoothness in the time domain. In SPoG, one can note from (17) that $\partial u_n = \lambda_n$ and, together with $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$, it follows that $0 = \partial u_1 \leq \ldots \leq \partial u_N$. Thus, one can similarly sort the eigensignals $\{u_n\}_{n=1}^N$ according to their graph smoothness and interpret the index n, or alternatively the eigenvalue λ_n, as the "graph frequency" of u_n.

By adopting the aforementioned SPoG notions, one can intuitively interpret the role of Laplacian kernels. Indeed, it follows from (15) that the regularizer in (8) strongly penalizes those $\tilde{f}_n$ for which the corresponding $r(\lambda_n)$ is large, thus promoting a specific structure in this graph-frequency domain. Specifically, one prefers $r(\lambda_n)$ to be large for those n for which $|\tilde{f}_{0,n}|^2 := |\boldsymbol{u}_n^T\boldsymbol{f}_0|^2$ is small and vice versa. The fact that $|\tilde{f}_{0,n}|^2$ is expected to decrease with n for smooth f_0 motivates the adoption of an increasing r [3]. Observe that Laplacian kernels can capture richer forms of prior information than the signal reconstructors of bandlimited signals in [12]–[15], [17], [18], since the latter can solely capture the support of the Fourier transform whereas the generalized approach here can also leverage magnitude information.

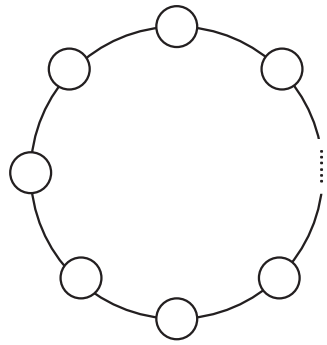


Fig. 1. An example of a circular graph.

Example 3 (circular graphs): This example capitalizes on Theorem 1 to present a novel SPoG-inspired intuitive interpretation of nonparametric regression with Laplacian kernels. To do so, a closed-form expression for the Laplacian kernel matrix of a circular graph (or ring), such as the one in Fig. 1 will be derived. This class of graphs has been commonly employed to illustrate connections between SPoG and signal processing of time-domain signals [5].

Up to vertex relabeling, an unweighted circular graph can be specified by $w(v_n, v_{n'}) = \delta[(n - n')_N - 1] + \delta[(n' - n)_N - 1]$. Therefore, its Laplacian matrix can be written as $\boldsymbol{L} = 2\boldsymbol{I}_N - \boldsymbol{R} - \boldsymbol{R}^T$, where $\boldsymbol{R}$ is the rotation matrix resulting from circularly shifting the columns of $\boldsymbol{I}_N$ one position to the right, i.e., $(\boldsymbol{R})_{n,n'} := \delta[(n' - n)_N - 1]$. Matrix $\boldsymbol{L}$ is *circulant* since its n-th row can be obtained by circularly shifting the $(n-1)$-th

[2]Recall from Section II that a graph signal f_0 can be equivalently represented by a vector $\boldsymbol{f}_0 := [f_0(v_1), \ldots, f_0(v_N)]^T \in \mathbb{R}^N$. This establishes a one-to-one correspondence between graph signals and vectors in $\mathbb{R}^N$, which implies that for any vector $\boldsymbol{u} \in \mathbb{R}^N$, there is a graph signal u such that $u(v_n) = (\boldsymbol{u})_n$.

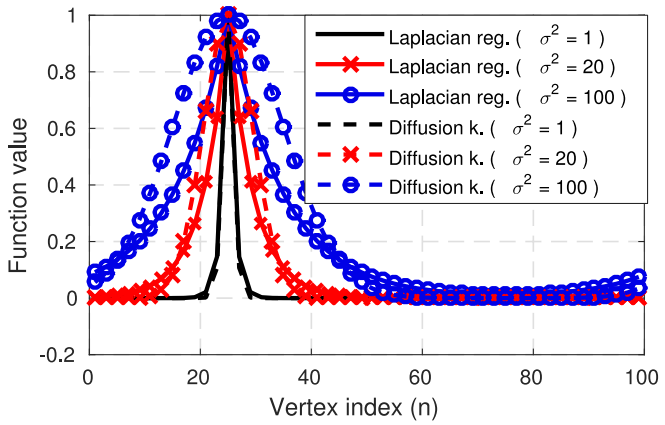


Fig. 2. 25-th column of $\boldsymbol{K}$ for a circular graph with $N = 100$ vertices. Different curves correspond to different parameter values for the Laplacian regularization and diffusion kernels.

row one position to the right. Hence, $\boldsymbol{L}$ can be diagonalized by the standard Fourier matrix [32], meaning

$$\boldsymbol{L} = \check{\boldsymbol{U}}\check{\boldsymbol{\Lambda}}\check{\boldsymbol{U}}^H \tag{18}$$

where $(\check{\boldsymbol{U}})_{m,m'} := (1/\sqrt{N})\exp\{j2\pi(m-1)(m'-1)/N\}$ is the unitary inverse discrete Fourier transform matrix and $(\check{\boldsymbol{\Lambda}})_{m,m'} := 2[1-\cos(2\pi(m-1)/N)]\delta[m-m']$. Matrices $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$ were replaced with $\check{\boldsymbol{U}}$ and $\check{\boldsymbol{\Lambda}}$ since, for notational brevity, the eigendecomposition (18) involves complex-valued eigenvectors and the eigenvalues have not been sorted in ascending order.

From (14), a Laplacian kernel matrix is given by $\boldsymbol{K} := \check{\boldsymbol{U}}r^{\dagger}(\check{\boldsymbol{\Lambda}})\check{\boldsymbol{U}}^H := \check{\boldsymbol{U}}\,\mathrm{diag}\,\{\boldsymbol{d}\}\,\check{\boldsymbol{U}}^H$, where $\boldsymbol{d} := [d_0,\ldots,d_{N-1}]^T$ has entries $d_n = r^{\dagger}(2[1-\cos(2\pi n/N)])$. It can be easily seen that $(\boldsymbol{K})_{m,m'} = D_{m-m'}$, where

$$D_m := \mathrm{IDFT}\{d_n\} := \frac{1}{N}\sum_{n=0}^{N-1} d_n e^{j\frac{2\pi}{N}mn} \tag{19}$$

is the m-th entry of the inverse discrete Fourier transform of $\{d_n\}_n$. If $r\,(2[1-\cos(2\pi n/N)]) > 0\ \forall n$, one has that

$$D_m = \frac{1}{N}\sum_{n=0}^{N-1} \frac{e^{j\frac{2\pi}{N}mn}}{r\,(2[1-\cos(2\pi n/N)])}. \tag{20}$$

This expression enables the construction of Laplacian kernel matrices for circular graphs in closed form.

Recall that Theorem 1 dictates that $\hat{\boldsymbol{f}}_0 = \sum_{s\in\mathcal{S}}\alpha_s\boldsymbol{\kappa}_s$, where $\boldsymbol{K} := [\boldsymbol{\kappa}_1,\ldots,\boldsymbol{\kappa}_N]$. Since $(\boldsymbol{K})_{m,m'} = D_{m-m'}$ and because D_m is periodic in m with period N, it follows that the vectors $\{\boldsymbol{\kappa}_n\}_{n=1}^N$ are all circularly shifted versions of each other. Moreover, since $\boldsymbol{K}$ is positive semidefinite, the largest entry of $\boldsymbol{\kappa}_s$ is precisely the s-th one, which motivates interpreting $\boldsymbol{\kappa}_s$ as an interpolating signal centered at s. This in turn suggests that the expression $\hat{\boldsymbol{f}}_0 = \sum_{s\in\mathcal{S}}\alpha_s\boldsymbol{\kappa}_s$ can be thought of as a reconstruction equation. From this vantage point, signals $\{\boldsymbol{\kappa}_s\}_{s\in\mathcal{S}}$ play an analogous role to sinc functions in signal processing of time-domain signals. Examples of these interpolating signals are depicted in Fig. 2.

2) Graph Bandlimited Kernels: A number of signal reconstruction approaches in the SPoG literature deal with graph bandlimited signals; see e.g. [11]–[18]. Here, the notion of bandlimited kernel is introduced to formally show that the LS estimator for bandlimited signals [11]–[16] is a limiting case of the kernel ridge regression estimate in (13). This notion will come handy in Sections VI and VII to estimate the bandwidth of a bandlimited signal from the observations $\{y_s\}_{s=1}^S$.

Signal f is said to be (graph) bandlimited if it admits an expansion (16) with $\tilde{f}_n$ supported on a set $\mathcal{B}\subset\{1,\ldots,N\}$; that is,

$$\boldsymbol{f} = \sum_{n\in\mathcal{B}}\tilde{f}_n\boldsymbol{u}_n = \boldsymbol{U}_{\mathcal{B}}\tilde{\boldsymbol{f}}_{\mathcal{B}} \tag{21}$$

where $\boldsymbol{U}_{\mathcal{B}}$ contains the columns of $\boldsymbol{U}$ with indices in $\mathcal{B}$, and $\tilde{\boldsymbol{f}}_{\mathcal{B}}$ is a vector stacking $\{\tilde{f}_n\}_{n\in\mathcal{B}}$. Multiple definitions of the bandwidth of f are possible. For instance, it can be defined as the cardinality $B := |\mathcal{B}|$ or as the greatest element of $\mathcal{B}$.

If f_0 is bandlimited, it follows from (1) that $\bar{\boldsymbol{y}} = \boldsymbol{\Phi}\boldsymbol{f}_0 + \bar{\boldsymbol{e}} = \boldsymbol{\Phi}\boldsymbol{U}_{\mathcal{B}}\tilde{\boldsymbol{f}}_{\mathcal{B}} + \bar{\boldsymbol{e}}$ for some $\tilde{\boldsymbol{f}}_{\mathcal{B}}$. The LS estimate of $\boldsymbol{f}_0$ is therefore given by [11]–[16]

$$\hat{\boldsymbol{f}}_{\mathrm{LS}} = \boldsymbol{U}_{\mathcal{B}}\arg\min_{\tilde{\boldsymbol{f}}_{\mathcal{B}}\in\mathbb{R}^B}||\bar{\boldsymbol{y}} - \boldsymbol{\Phi}\boldsymbol{U}_{\mathcal{B}}\tilde{\boldsymbol{f}}_{\mathcal{B}}||^2 \tag{22a}$$

$$= \boldsymbol{U}_{\mathcal{B}}[\boldsymbol{U}_{\mathcal{B}}{}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{U}_{\mathcal{B}}]^{-1}\boldsymbol{U}_{\mathcal{B}}{}^T\boldsymbol{\Phi}^T\bar{\boldsymbol{y}} \tag{22b}$$

where the second equality assumes that $\boldsymbol{U}_{\mathcal{B}}{}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{U}_{\mathcal{B}}$ is invertible, a necessary and sufficient condition for the B entries of $\tilde{\boldsymbol{f}}_{\mathcal{B}}$ to be identifiable.

The estimate $\hat{\boldsymbol{f}}_{\mathrm{LS}}$ in (22) can be accommodated in the kernel regression framework by properly constructing a bandlimited kernel. Intuitively, one can adopt a Laplacian kernel for which $r(\lambda_n)$ is large if $n\notin\mathcal{B}$ (cf. Section IV-A1). Consider the Laplacian kernel $\boldsymbol{K}_\beta$ with

$$r_\beta(\lambda_n) = \begin{cases} 1/\beta & n\in\mathcal{B} \\ \beta & n\notin\mathcal{B}. \end{cases} \tag{23}$$

For large β, this function strongly penalizes $\{\tilde{f}_n\}_{n\notin\mathcal{B}}$ (cf. (15)), which promotes bandlimited estimates. The reason for setting $r(\lambda_n) = 1/\beta$ for $n\in\mathcal{B}$ instead of $r(\lambda_n) = 0$ is to ensure that $\boldsymbol{K}_\beta$ is non-singular, a property that simplifies the statement and the proofs of some of the results in this paper.

Proposition 1: Let $\hat{\boldsymbol{f}}_{\mathrm{RR}}$ denote the kernel ridge regression estimate from (13) with kernel $\boldsymbol{K}_\beta$ as in (23) and $\mu > 0$. If $\boldsymbol{U}_{\mathcal{B}}{}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{U}_{\mathcal{B}}$ is invertible, as required by the estimator in (22b) for bandlimited signals, then $\hat{\boldsymbol{f}}_{\mathrm{RR}}\to\hat{\boldsymbol{f}}_{\mathrm{LS}}$ as $\beta\to\infty$.

Proof: See Appendix C. ■

Proposition 1 shows that the framework of kernel-based regression subsumes LS estimation of bandlimited signals. A non-asymptotic counterpart of Proposition 1 can be found by setting $r_\beta(\lambda_n) = 0$ for $n\in\mathcal{B}$ in (23), and noting that $\hat{\boldsymbol{f}}_{\mathrm{RR}} = \hat{\boldsymbol{f}}_{\mathrm{LS}}$ if $\mu = 0$. However, note that imposing $\mu = 0$ renders $\hat{\boldsymbol{f}}_{\mathrm{RR}}$ a degenerate kernel-based estimate.

Remark 1: The definition of a bandlimited signal in (21) adopts the columns of $\boldsymbol{U}$ as basis vectors. However, further definitions have been proposed involving alternative basis vectors, other than the eigenvectors of the Laplacian matrix (see

e.g. [18]). This means that a signal may be graph bandlimited or not depending on the basis adopted. Nonetheless, the approach described in this section can also be applied to construct graph bandlimited kernels for any chosen basis. Clearly, the resulting graph kernels will not in general belong to the Laplacian family.

Remark 2: Additional signal reconstructors can be interpreted as kernel-based regression methods for certain topology-based kernels. Specifically, it can be seen that [22, eq. (27)] is tantamount to kernel ridge regression with

$$\boldsymbol{K} = [(\boldsymbol{I}_N - \boldsymbol{W})^T(\boldsymbol{I}_N - \boldsymbol{W})]^{-1} \tag{24}$$

provided that the adjacency matrix $\boldsymbol{W}$ is properly scaled so that this inverse exists. Another example is the Tikhonov regularized estimate in [12, eq. (15)], which is recovered as kernel ridge regression upon setting

$$\boldsymbol{K} = [\boldsymbol{H}^T\boldsymbol{H} + \epsilon\boldsymbol{I}_N]^{-1} \tag{25}$$

and letting $\epsilon > 0$ tend to 0, where $\boldsymbol{H}$ can be viewed as a high-pass filter matrix. The role of the term $\epsilon\boldsymbol{I}_N$ is to ensure that the matrix within brackets is invertible.

B. Vertex-Covariance Kernels

So far, signal f_0 has been assumed deterministic, which precludes accommodating certain forms of prior information that probabilistic models can capture, such as domain knowledge and historical data. A probabilistic interpretation of kernel methods on graphs will be pursued here to show that: (i) vertex-covariance kernels enable signal reconstruction when the graph topology is unknown; (ii) the optimal $\boldsymbol{K}$ in the MSE sense for ridge regression is the vertex-covariance kernel; and, (iii) kernel-based ridge regression can be interpreted as a local LMMSE estimator on a Markov random field [33, Ch. 8].

Suppose without loss of generality that $\{f_0(v_n)\}_{n=1}^N$ are zero-mean random variables. The LMMSE estimator of $\boldsymbol{f}_0$ given $\bar{\boldsymbol{y}}$ is the linear estimator $\hat{\boldsymbol{f}}_{\text{LMMSE}}$ minimizing $\mathbb{E}||\boldsymbol{f}_0 - \hat{\boldsymbol{f}}_{\text{LMMSE}}||_2^2$, where the expectation is over all $\boldsymbol{f}_0$ and noise realizations. With $\boldsymbol{C} := \mathbb{E}\left[\boldsymbol{f}_0{\boldsymbol{f}_0}^T\right]$, the LMMSE estimate is given by

$$\hat{\boldsymbol{f}}_{\text{LMMSE}} = \boldsymbol{C}\boldsymbol{\Phi}^T[\boldsymbol{\Phi}\boldsymbol{C}\boldsymbol{\Phi}^T + \sigma_e^2\boldsymbol{I}_S]^{-1}\bar{\boldsymbol{y}} \tag{26}$$

where $\sigma_e^2 := (1/S)\mathbb{E}\left[||\bar{\boldsymbol{e}}||_2^2\right]$ denotes the noise variance. Comparing (26) with (13) and recalling that $\bar{\boldsymbol{K}} := \boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\Phi}^T$, it follows that $\hat{\boldsymbol{f}}_{\text{LMMSE}} = \hat{\boldsymbol{f}}_{\text{RR}}$ if $\mu S = \sigma_e^2$ and $\boldsymbol{K} = \boldsymbol{C}$. In other words, the similarity measure $\kappa(v_n, v_{n'})$ embodied in such a kernel map is just the covariance $\text{cov}[f_0(v_n), f_0(v_{n'})]$. A related observation was pointed out in [34] for general kernel methods.

In short, one can interpret kernel ridge regression as the LMMSE estimator of a signal $\boldsymbol{f}_0$ with covariance matrix equal to $\boldsymbol{K}$; see also [35] and [36] for alternative probabilistic interpretations and data-dependent kernels in the context of semi-supervised learning. This generalizes [13, Lemma 1], which requires $\boldsymbol{f}_0$ to be Gaussian, $\boldsymbol{C}$ rank-deficient, and $\sigma_e^2 = 0$. The LMMSE interpretation also suggests the usage of $\boldsymbol{C}$ as a kernel matrix, which enables signal reconstruction even when the graph topology is unknown. Although this discussion hinges on kernel ridge regression after setting $\boldsymbol{K} = \boldsymbol{C}$, any other kernel estimator of the form (7) can benefit from vertex-covariance kernels too.

Recognizing that kernel ridge regression is a linear estimator readily establishes the following result.

Proposition 2: If $\text{MSE}(\boldsymbol{K}, \mu) := \mathbb{E}[||\boldsymbol{f}_0 - \hat{\boldsymbol{f}}_{\text{RR}}(\boldsymbol{K}, \mu)||^2]$, *where* $\hat{\boldsymbol{f}}_{\text{RR}}(\boldsymbol{K}, \mu)$ *denotes the estimator in (13) with kernel matrix* $\boldsymbol{K}$ *and regularization parameter* μ, *it then holds that*

$$\text{MSE}(\boldsymbol{C}, \sigma_e^2/S) \leq \text{MSE}(\boldsymbol{K}, \mu) \tag{27}$$

for all kernel matrices $\boldsymbol{K}$ *and* $\mu > 0$.

Thus, for criteria aiming to minimize the MSE, Proposition 2 suggests choosing $\boldsymbol{K}$ "close" to $\boldsymbol{C}$. This observation may be employed for kernel selection and for parameter tuning in graph signal reconstruction methods of the kernel ridge regression family (e.g. the Tikhonov regularized estimators from [4], [12], [21] and [22, eq. (27)]) whenever an estimate of $\boldsymbol{C}$ can be obtained from historical data. For instance, the function r involved in Laplacian kernels can be chosen so that $\boldsymbol{K}$ resembles $\boldsymbol{C}$ in some sense. Investigating such approaches goes beyond the scope of this paper.

An insightful implication of the probabilistic interpretation in this section is that kernel ridge regression can be thought of as a local LMMSE estimator on a Markov random field [33, Ch. 8]. In this class of graphical models, an edge connects v_n with $v_{n'}$ if $f_0(v_n)$ and $f_0(v_{n'})$ are *not* independent given $\{f_0(v_{n''})\}_{n'' \neq n,n'}$. Thus, if $v_{n'} \notin \mathcal{N}_n$, then $f_0(v_n)$ and $f_0(v_{n'})$ are independent given $\{f_0(v_{n''})\}_{n'' \neq n,n'}$. In other words, when $f_0(v_{n''})$ is known for all neighbors $v_{n''} \in \mathcal{N}_n$ of v_n, function values at non-neighboring vertices do not provide further information about $f_0(v_n)$. This spatial Markovian property motivates the name of this class of graphical models. Real-world graphs obey this property when the topology captures direct (unmediated) "interaction", in the sense that the interaction between the entities represented by two non-neighboring vertices v_n and $v_{n'}$ is necessarily through vertices in a *path* connecting v_n with $v_{n'}$.

Proposition 3: Let $\mathcal{G}$ *be a Markov random field, let* $\text{LMMSEE}[f_0(v_n)|\{\hat{f}_{\text{RR}}(v)\}_{v\in\mathcal{N}_n}]$ *denote the LMMSE estimator of* $f_0(v_n)$ *given* $f_0(v) = \hat{f}_{\text{RR}}(v)$, $v \in \mathcal{N}_n$, *and let* $\sigma^2_{n|\mathcal{N}_n}$ *be its variance. If* $\boldsymbol{K} = \boldsymbol{C} := \mathbb{E}[\boldsymbol{f}_0{\boldsymbol{f}_0}^T]$ *and* $\mu = \sigma_e^2/S$, *the estimator in (13) satisfies*

$$\hat{f}_{\text{RR}}(v_n) = \begin{cases} \text{LMMSEE}\left[f_0(v_n)\middle|\{\hat{f}_{\text{RR}}(v)\}_{v\in\mathcal{N}_n}\right] & \text{if } n \notin \mathcal{S} \\ y_{s(n)} - \hat{e}_{s(n)} & \text{if } n \in \mathcal{S} \end{cases} \tag{28}$$

for $n = 1, \ldots, N$, *where* $s(n)$ *denotes the sample index of the observed vertex* v_n, *i.e.*, $y_{s(n)} = f_0(v_n) + e_{s(n)}$, *and*

$$\hat{e}_{s(n)} = \frac{\sigma_e^2}{\sigma^2_{n|\mathcal{N}_n}}\left[\hat{f}_{\text{RR}}(v_n) - \text{LMMSEE}\left[f_0(v_n)\middle|\{\hat{f}_{\text{RR}}(v)\}_{v\in\mathcal{N}_n}\right]\right]. \tag{29}$$

Proof: See Appendix D. ■

If a (noisy) observation of f_0 at v_n is not available, i.e. $n \notin \mathcal{S}$, then kernel ridge regression finds $\hat{f}_{\text{RR}}(v_n)$ as the LMMSE estimate of $f_0(v_n)$ given function values at the neighbors of v_n. However, since the latter are not observable, their ridge regression estimates are used instead. Conversely, when v_n is observed, implying that a sample $y_{s(n)}$ is available, the sought

estimator subtracts from this value an estimate $\hat{e}_{s(n)}$ of the observation noise $e_{s(n)}$. Therefore, the kernel estimate on a Markov random field seeks an estimate satisfying the system of *local LMMSE conditions* given by (28) for $n = 1, \ldots, N$.

Remark 3: In Proposition 3, the requirement that $\mathcal{G}$ is a Markov random field can be relaxed to that of being a *conditional correlation graph*, defined as a graph where $(v_n, v_{n'}) \in \mathcal{E}$ if $f_0(v_n)$ and $f_0(v_{n'})$ are correlated given $\{f_0(v_{n''})\}_{n'' \neq n,n'}$. Since correlation implies dependence, any Markov random field is also a conditional correlation graph. If $\boldsymbol{f}_0$ is Gaussian, a conditional correlation graph can be constructed from $\boldsymbol{C} := \mathbb{E}[\boldsymbol{f}_0 \boldsymbol{f}_0{}^T]$ by setting $\mathcal{E} = \{(v_n, v_{n'}) : (\boldsymbol{C}^{-1})_{n,n'} \neq 0,\ n \neq n'\}$ (see e.g. [37, Th. 10.2]).

Remark 4: Suppose that kernel ridge regression is adopted to estimate a function f_0 on a certain graph $\mathcal{G}$, not necessarily a Markov random field, using a kernel $\boldsymbol{K} \neq \boldsymbol{C} := \mathbb{E}[\boldsymbol{f}_0 \boldsymbol{f}_0{}^T]$. Then it can still be interpreted as a method applying (28) on a conditional correlation graph $\mathcal{G}'$ and adopting a signal covariance matrix $\boldsymbol{K}$.

Remark 5 (topology-based vis-à-vis vertex-covariance kernels): The estimators considered in this paper can be written as in (8), where the regularizer is an increasing function of $\boldsymbol{f}^T \boldsymbol{K}^\dagger \boldsymbol{f}$. If $\boldsymbol{K}$ is a vertex-covariance kernel, this term equals $\boldsymbol{f}^T \boldsymbol{C}^{-1} \boldsymbol{f}$. Alternatively, if $\boldsymbol{K}$ is a Laplacian kernel, say with $r(\lambda) = \lambda$, that term becomes $\boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f}$. This observation suggests that the roles played by $\boldsymbol{L}$ and $\boldsymbol{C}^{-1}$ are in some sense analogous, which bears important consequences for topology identification. Specifically, recall that it is customary to infer $\mathcal{E}$ from the estimate $\hat{\boldsymbol{C}}$ as $\hat{\mathcal{E}}_1 = \{(v_n, v_{n'}) : |(\hat{\boldsymbol{C}})_{n,n'}| > \tau,\ n \neq n'\}$ for some threshold τ. However, the definition of $\boldsymbol{L}$ implies that $\mathcal{E} = \{(v_n, v_{n'}) : (\boldsymbol{L})_{n,n'} \neq 0,\ \ n \neq n'\}$, which, together with the aforementioned parallelism between $\boldsymbol{L}$ and $\boldsymbol{C}^{-1}$, favors estimators of the form $\hat{\mathcal{E}}_2 = \{(v_n, v_{n'}) : |(\hat{\boldsymbol{C}}^{-1})_{n,n'}| > \tau,\ n \neq n'\}$. The graphical lasso [38], [39] is a prominent estimator of this class. If $\boldsymbol{f}_0$ is Gaussian and a sufficiently large number of realizations of $\boldsymbol{f}_0$ is given, then $\hat{\mathcal{E}}_2$ estimators can be shown to place an edge between v_n and $v_{n'}$ if $f_0(v_n)$ and $f_0(v_{n'})$ are not conditionally independent given $\{f(v_{n''})\}_{n'' \neq n,n'}$. Informally, this means that the resulting edges will correspond to unmediated interactions between vertices. This is different from what happens with $\hat{\mathcal{E}}_1$, where the resulting edges will correspond to mediated as well as unmediated interactions.

V. Kernel-Based Smoothing and Graph Filtering

When an observation y_n is available per vertex v_n for $n = 1, \ldots, N$, kernel methods can still be employed for denoising purposes. Due to the regularizer in (6), the estimate $\hat{\boldsymbol{f}}_0$ will be a smoothed version of $\bar{\boldsymbol{y}}$. This section shows how ridge regression smoothers can be thought of as graph filters, and vice versa. The importance of this two-way link is in establishing that kernel smoothers can be implemented in a decentralized fashion as graph filters [4].

Upon setting $\boldsymbol{\Phi} = \boldsymbol{I}_N$ in (13), one recovers the ridge regression smoother $\hat{\boldsymbol{f}}_{\text{RRS}} = \boldsymbol{K}(\boldsymbol{K} + \mu N \boldsymbol{I}_N)^{-1} \bar{\boldsymbol{y}}$. If $\boldsymbol{K}$ is a Laplacian kernel, then

$$\hat{\boldsymbol{f}}_{\text{RRS}} = \boldsymbol{U} \tilde{r}(\boldsymbol{\Lambda}) \boldsymbol{U}^T \bar{\boldsymbol{y}} \tag{30}$$

where $\tilde{r}(\lambda) := \mathcal{I}[r(\lambda) \neq 0]/[1 + \mu N r(\lambda)]$.

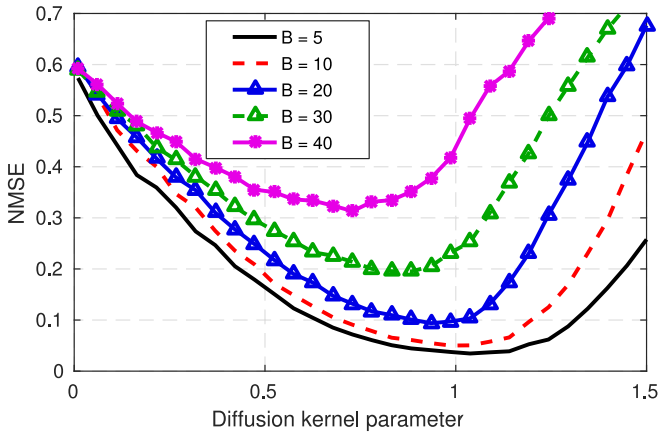


Fig. 3. Influence of the diffusion kernel parameter σ^2 on NMSE for $S = 40$ and several bandwidths B ($N = 100$, SNR $= 20$ dB, $\mu = 10^{-4}$).

To see how (30) relates to a graph filter, recall that the latter is an operator assigning $\bar{\boldsymbol{y}} \mapsto \boldsymbol{z}$, where [4]

$$\boldsymbol{z} := \left(h_0 \boldsymbol{I}_N + \sum_{n=1}^{N-1} h_n \boldsymbol{L}^n \right) \bar{\boldsymbol{y}} \tag{31a}$$

$$= \boldsymbol{U} \left(h_0 \boldsymbol{I}_N + \sum_{n=1}^{N-1} h_n \boldsymbol{\Lambda}^n \right) \boldsymbol{U}^T \bar{\boldsymbol{y}}. \tag{31b}$$

Graph filters can be implemented in a decentralized fashion since (31a) involves successive products of $\bar{\boldsymbol{y}}$ by $\boldsymbol{L}$ and these products can be computed at each vertex by just exchanging information with neighbors. Expression (31b) can be rewritten in the Fourier domain (cf. Section IV-A1) as $\tilde{\boldsymbol{z}} = [h_0 \boldsymbol{I}_N + \sum_{n=1}^{N-1} h_n \boldsymbol{\Lambda}^n] \tilde{\boldsymbol{y}}$ upon defining $\tilde{\boldsymbol{z}} := \boldsymbol{U}^T \boldsymbol{z}$ and $\tilde{\boldsymbol{y}} := \boldsymbol{U}^T \bar{\boldsymbol{y}}$. For this reason, the diagonal of $h_0 \boldsymbol{I}_N + \sum_{n=1}^{N-1} h_n \boldsymbol{\Lambda}^n$ is referred to as the frequency response of the filter.

Comparing (30) with (31b) shows that $\hat{\boldsymbol{f}}_{\text{RRS}}$ can be interpreted as a graph filter with frequency response $\tilde{r}(\boldsymbol{\Lambda})$. Thus, implementing $\hat{\boldsymbol{f}}_{\text{RRS}}$ in a decentralized fashion using (31a) boils down to solving for $\{h_n\}_{n=1}^N$ the system of linear equations $\{h_0 + \sum_{n'=1}^{N-1} h_{n'} \lambda_n^{n'} = \tilde{r}(\lambda_n)\}_{n=1}^N$. Conversely, given a filter, a Laplacian kernel can be found so that filter and smoother coincide. To this end, assume without loss of generality that $\tilde{h}_n \leq 1\ \forall n$, where $\tilde{h}_n := h_0 + \sum_{n'=1}^{N-1} h_{n'} \lambda_n^{n'}$; otherwise, simply scale $\{h_n\}_{n=0}^{N-1}$. Then, given $\{h_n\}_{n=0}^{N-1}$, the sought kernel can be constructed by setting

$$r(\lambda_n) = \frac{1 - \tilde{h}_n}{\mu N \tilde{h}_n} \mathcal{I}[\tilde{h}_n \neq 0]. \tag{32}$$

VI. Multi-Kernel Graph Signal Reconstruction

One of the limitations of kernel methods is their sensitivity to the choice of the kernel. To appreciate this, Fig. 3 depicts the normalized mean-square error (NMSE) $\mathbb{E}||\boldsymbol{f}_0 - \hat{\boldsymbol{f}}_0||_2^2 / \mathbb{E}||\boldsymbol{f}_0||_2^2$ when $\mathcal{L}$ is the square loss and $\Omega(\zeta) = \zeta$ across the parameter σ^2 of the adopted diffusion kernel (see Section IV-A1). The simulation setting is described in Section VII. At this point

though, it suffices to stress the impact of σ^2 on the NMSE and the dependence of the optimum σ^2 on the bandwidth B of f_0.

Similarly, the performance of estimators for bandlimited signals degrades considerably if the estimator assumes a frequency support $\mathcal{B}$ that differs from the actual one. Even for estimating *low-pass signals*, for which $\mathcal{B}=\{1,\ldots,B\}$, the parameter B is unknown in practice. Approaches for setting B were considered in [11], [16], but they rely solely on $\mathcal{S}$ and $\boldsymbol{L}$, disregarding the observations $\bar{\boldsymbol{y}}$.

Clearly, bandwidth selection reduces to kernel selection among a family of bandlimited kernels; see Section IV-A2. Therefore, although this paper proposes algorithms for kernel selection, it is immediate to accomplish bandwidth selection by focusing on bandlimited kernels.

This section advocates an MKL approach to kernel selection in graph signal reconstruction. Two algorithms with complementary strengths will be developed. Both select the most suitable kernels within a user-specified kernel dictionary.

A. RKHS Superposition

Since $\mathcal{H}$ in (6) is determined by κ, kernel selection is tantamount to RKHS selection. Therefore, a kernel dictionary $\{\kappa_m\}_{m=1}^M$ gives rise to an RKHS dictionary $\{\mathcal{H}_m\}_{m=1}^M$, which motivates estimates of the form[3]

$$\hat{f}=\sum_{m=1}^{M}\hat{f}_m,\quad \hat{f}_m\in\mathcal{H}_m. \tag{33}$$

Upon adopting a criterion that controls sparsity in this expansion, the "best" RKHSs will be selected. A reasonable approach is therefore to generalize (6) to accommodate multiple RKHSs. With $\mathcal{L}$ selected as the square loss and $\Omega(\zeta)=\zeta$, one can pursue an estimate $\hat{f}$ by solving

$$\min_{\{f_m\in\mathcal{H}_m\}_{m=1}^M}\frac{1}{S}\sum_{s=1}^{S}\left[y_s-\sum_{m=1}^{M}f_m(v_{n_s})\right]^2+\mu\sum_{m=1}^{M}\|f_m\|_{\mathcal{H}_m}. \tag{34}$$

Invoking Theorem 1 per f_m establishes that the minimizers of (34) can be written as

$$\hat{f}_m(v)=\sum_{s=1}^{S}\bar{\alpha}_s^m\kappa_m(v,v_{n_s}),\quad m=1,\ldots,M \tag{35}$$

for some coefficients $\bar{\alpha}_s^m$. Substituting (35) into (34) suggests obtaining these coefficients as

$$\arg\min_{\{\bar{\boldsymbol{\alpha}}_m\}_{m=1}^M}\frac{1}{S}\left\|\bar{\boldsymbol{y}}-\sum_{m=1}^{M}\bar{\boldsymbol{K}}_m\bar{\boldsymbol{\alpha}}_m\right\|^2+\mu\sum_{m=1}^{M}\left(\bar{\boldsymbol{\alpha}}_m^T\bar{\boldsymbol{K}}_m\bar{\boldsymbol{\alpha}}_m\right)^{1/2} \tag{36}$$

where $\bar{\boldsymbol{\alpha}}_m:=[\bar{\alpha}_1^m,\ldots,\bar{\alpha}_S^m]^T$ and $\bar{\boldsymbol{K}}_m:=\boldsymbol{\Phi}\boldsymbol{K}_m\boldsymbol{\Phi}^T$, with $(\boldsymbol{K}_m)_{n,n'}:=\kappa_m(v_n,v_{n'})$. If one lets $\breve{\boldsymbol{\alpha}}_m:=\bar{\boldsymbol{K}}_m^{1/2}\bar{\boldsymbol{\alpha}}_m$, solving (36) amounts to solving

$$\arg\min_{\{\breve{\boldsymbol{\alpha}}_m\}_{m=1}^M}\frac{1}{S}\left\|\bar{\boldsymbol{y}}-\sum_{m=1}^{M}\bar{\boldsymbol{K}}_m^{1/2}\breve{\boldsymbol{\alpha}}_m\right\|^2+\mu\sum_{m=1}^{M}\|\breve{\boldsymbol{\alpha}}_m\|_2. \tag{37}$$

Note that the sum in the regularizer of (37) can be interpreted as the ℓ_1-norm of $[||\breve{\boldsymbol{\alpha}}_1||_2,\ldots,||\breve{\boldsymbol{\alpha}}_M||_2]^T$, which is known to promote sparsity in its entries and therefore in (33). Indeed, (37) can be seen as a particular instance of group lasso [34].

As shown next, (37) can be efficiently solved using the alternating-direction method of multipliers (ADMM) [40]. To this end, rewrite (37) by defining $\boldsymbol{\Upsilon}:=[\bar{\boldsymbol{K}}_1^{1/2},\ldots,\bar{\boldsymbol{K}}_M^{1/2}]$ and $\breve{\boldsymbol{\alpha}}:=[\breve{\boldsymbol{\alpha}}_1^T,\ldots,\breve{\boldsymbol{\alpha}}_M^T]^T$, and introducing the auxiliary variable $\boldsymbol{\beta}:=[\boldsymbol{\beta}_1^T,\ldots,\boldsymbol{\beta}_M^T]^T$, as

$$\begin{aligned}\min_{\breve{\boldsymbol{\alpha}},\boldsymbol{\beta}}\quad & \frac{1}{2}\|\bar{\boldsymbol{y}}-\boldsymbol{\Upsilon}\boldsymbol{\beta}\|^2+\frac{S\mu}{2}\sum_{m=1}^{M}\|\breve{\boldsymbol{\alpha}}_m\|_2\\ \text{s. to}\quad & \breve{\boldsymbol{\alpha}}-\boldsymbol{\beta}=\boldsymbol{0}.\end{aligned} \tag{38}$$

ADMM iteratively minimizes the *augmented Lagrangian* of (38) with respect to $\breve{\boldsymbol{\alpha}}$ and $\boldsymbol{\beta}$ in a block-coordinate descent fashion, and updates the Lagrange multipliers associated with the equality constraint using gradient ascent (see [41] and references therein). The resulting iteration is summarized as Algorithm 1, where ρ is the augmented Lagrangian parameter, $\boldsymbol{\nu}:=[\boldsymbol{\nu}_1^T,\ldots,\boldsymbol{\nu}_M^T]^T$ is the Lagrange multiplier associated with the equality constraint, and

$$\mathcal{T}_\zeta(\boldsymbol{a}):=\frac{\max(0,||\boldsymbol{a}||_2-\zeta)}{||\boldsymbol{a}||_2}\boldsymbol{a} \tag{39}$$

is the so-called soft-thresholding operator [40].

After obtaining $\{\breve{\boldsymbol{\alpha}}_m\}_{m=1}^M$ from Algorithm 1, the wanted function estimate can be recovered as

$$\hat{\boldsymbol{f}}_0=\sum_{m=1}^{M}\boldsymbol{K}_m\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}}_m=\sum_{m=1}^{M}\boldsymbol{K}_m\boldsymbol{\Phi}^T\bar{\boldsymbol{K}}_m^{-1/2}\breve{\boldsymbol{\alpha}}_m. \tag{40}$$

It is recommended to normalize the kernel matrices in order to prevent imbalances in the kernel selection. Specifically, one can scale $\{\boldsymbol{K}_m\}_{m=1}^M$ such that $\mathrm{Tr}(\boldsymbol{K}_m)=1\ \forall m$. If $\boldsymbol{K}_m$ is a Laplacian kernel (see Section IV-A1), where $\boldsymbol{K}_m=\boldsymbol{U}r_m^\dagger(\boldsymbol{\Lambda})\boldsymbol{U}^T$, one can scale r_m to ensure $\sum_{n=1}^N r_m^\dagger(\lambda_n)=1$.

Remark 6: Although criterion (34) is reminiscent of the MKL approach of [34], the latter differs markedly because it assumes that the right-hand side of (33) is uniquely determined given $\hat{f}_0$, which allows application of (6) over a direct-sum RKHS $\mathcal{H}:=\mathcal{H}_1\oplus\cdots\oplus\mathcal{H}_N$ with an appropriately defined norm. However,

Algorithm 1: ADMM for Multi-kernel Regression.

1: Input: $\rho,\epsilon>0,\breve{\boldsymbol{\alpha}}^{(0)},\boldsymbol{\nu}^0$
2: **repeat**
3: $\breve{\boldsymbol{\alpha}}_m^{(k+1)}=\mathcal{T}_{\mu S/2\rho}(\boldsymbol{\beta}_m^{(k)}+\boldsymbol{\nu}_m^{(k)}),\quad m=1,\ldots,M$
4: $\boldsymbol{\beta}^{(k+1)}=(\boldsymbol{\Upsilon}^T\boldsymbol{\Upsilon}+\rho\boldsymbol{I})^{-1}[\boldsymbol{\Upsilon}^T\bar{\boldsymbol{y}}+\rho(\breve{\boldsymbol{\alpha}}^{(k+1)}-\boldsymbol{\nu}^{(k)})]$
5: $\boldsymbol{\nu}_m^{(k+1)}=\boldsymbol{\nu}_m^{(k)}+\boldsymbol{\beta}_m^{(k+1)}-\breve{\boldsymbol{\alpha}}_m^{(k+1)},\ \ m=1,\ldots,M$
6: $k\leftarrow k+1$
7: **until** $||\boldsymbol{\beta}^{(k+1)}-\breve{\boldsymbol{\alpha}}^{(k+1)}||\le\epsilon$

[3] A sum is chosen here for tractability, but the right-hand side of (33) could in principle combine the functions $\{\hat{f}_m\}_m$ in different forms.

Algorithm 2: Interpolated Iterative Algorithm.

Input: $\boldsymbol{\theta}^{(0)}$, $\{\bar{\boldsymbol{K}}_m\}_{m=1}^M$, μ , $\boldsymbol{\theta}_0$, R, η, ϵ.
$\bar{\boldsymbol{\alpha}}^{(0)} = (\bar{\boldsymbol{K}}(\boldsymbol{\theta}^{(0)}) + \mu S\boldsymbol{I})^{-1}\bar{\boldsymbol{y}}$
$k = 0$
repeat
  $\boldsymbol{\xi}^{(k)} = [(\bar{\boldsymbol{\alpha}}^{(k)})^T \bar{\boldsymbol{K}}_0 \bar{\boldsymbol{\alpha}}^{(k)}, \ldots, (\bar{\boldsymbol{\alpha}}^{(k)})^T \bar{\boldsymbol{K}}_M \bar{\boldsymbol{\alpha}}^{(k)}]^T$
  $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}_0 + (R/\|\boldsymbol{\xi}^{(k)}\|_2)\, \boldsymbol{\xi}^{(k)}$
  $\bar{\boldsymbol{\alpha}}^{(k+1)} = \eta\bar{\boldsymbol{\alpha}}^{(k)} + (1-\eta)[\bar{\boldsymbol{K}}(\boldsymbol{\theta}^{(k)}) + \mu S\boldsymbol{I}]^{-1}\bar{\boldsymbol{y}}$
  $k \leftarrow k+1$
until $\|\bar{\boldsymbol{\alpha}}^{(k+1)} - \bar{\boldsymbol{\alpha}}^{(k)}\| < \epsilon$

this approach cannot be pursued here since RKHSs of graph signals frequently overlap, implying that their sum is not a direct one (cf. discussion after (5)).

B. Kernel Superposition

The MKL algorithm in Section VI-A can identify the best subset of RKHSs and therefore kernels, but entails MS unknowns (cf. (36)). This section introduces an alternative approach entailing only $M + S$ variables at the price of not guaranteeing a sparse kernel expansion.

The approach is to postulate a kernel of the form $\boldsymbol{K}(\boldsymbol{\theta}) = \sum_{m=1}^M \theta_m \boldsymbol{K}_m$, where $\{\boldsymbol{K}_m\}_{m=1}^M$ is given and $\theta_m \geq 0\ \forall m$. The coefficients $\boldsymbol{\theta} := [\theta_1, \ldots, \theta_M]^T$ can be found by jointly minimizing (11) with respect to $\boldsymbol{\theta}$ and $\bar{\boldsymbol{\alpha}}$ [42]

$$(\boldsymbol{\theta}, \hat{\bar{\boldsymbol{\alpha}}}) := \arg\min_{\boldsymbol{\theta}, \bar{\boldsymbol{\alpha}}} \mathcal{L}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{K}}(\boldsymbol{\theta})\bar{\boldsymbol{\alpha}}) + \mu\Omega((\bar{\boldsymbol{\alpha}}^T \bar{\boldsymbol{K}}(\boldsymbol{\theta})\bar{\boldsymbol{\alpha}})^{1/2}) \tag{41}$$

where $\bar{\boldsymbol{K}}(\boldsymbol{\theta}) := \boldsymbol{\Phi}\boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{\Phi}^T$. Except for degenerate cases, problem (41) is not jointly convex in $\boldsymbol{\theta}$ and $\hat{\bar{\boldsymbol{\alpha}}}$, but it is separately convex in each of these vectors if $\mathcal{L}$ is convex in the last argument [42]. Criterion (41) generalizes the one in [43], which aims at combining Laplacian matrices of multiple graphs sharing the same vertex set.

A method termed interpolated iterative algorithm (IIA) was proposed in [44] to solve (41) when $\mathcal{L}$ is the square loss, $\Omega(\zeta) = \zeta^2$, and $\boldsymbol{\theta}$ is constrained to lie in a ball $\boldsymbol{\Theta} := \{\boldsymbol{\theta} : \boldsymbol{\theta} \geq \mathbf{0}$ and $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\| \leq R\}$ for some user-defined center $\boldsymbol{\theta}_0$ and radius $R > 0$. This constraint ensures that $\boldsymbol{\theta}$ does not diverge. The first-order optimality conditions for (41) yield a nonlinear system of equations, which IIA solves iteratively. This algorithm is listed as Algorithm 2, where $\eta > 0$ is the step size.

As a special case, it is worth noting that Algorithm 2 enables kernel selection in ridge regression smoothing, which is tantamount to optimal filter selection for graph signal denoising (cf. Section V). In this case, Algorithm 2 enjoys a particularly efficient implementation for Laplacian kernels since their kernel matrices share eigenvectors. Specifically, recalling that $\bar{\boldsymbol{K}}_m = \boldsymbol{K}_m = \boldsymbol{U} r_m^\dagger(\boldsymbol{\Lambda})\boldsymbol{U}^T$ for smoothing and letting $\tilde{\boldsymbol{\alpha}} := [\tilde{\alpha}_1, \ldots, \tilde{\alpha}_N]^T := \boldsymbol{U}^T\boldsymbol{\alpha} = \boldsymbol{U}^T\bar{\boldsymbol{\alpha}}$, suggests that the $\bar{\boldsymbol{\alpha}}$-update in Algorithm 2 can be replaced with its scalar version

$$\tilde{\alpha}_n^{(k+1)} = \eta\tilde{\alpha}_n^{(k)} + \frac{(1-\eta)\tilde{y}_n}{\sum_{m=1}^M \theta_m^{(k)} r_m^\dagger(\lambda_n) + \mu S},\ n = 1, \ldots, N \tag{42}$$

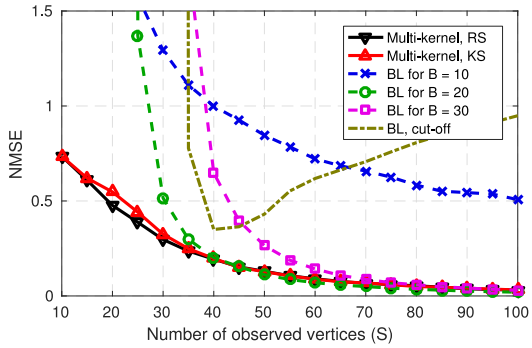


Fig. 4. Comparison of different algorithms for estimating bandlimited signals. Per Monte Carlo iteration, a bandlimited signal with $B = 20$ is generated ($N = 100$, SNR $= 10$ dB).

whereas the $\boldsymbol{\xi}$-update can be replaced with $\xi_m^{(k)} = \sum_{n=1}^N r_m^\dagger(\lambda_n)(\tilde{\alpha}_n^{(k)})^2$, where $\boldsymbol{\xi} := [\xi_1, \ldots, \xi_M]^T$.

To sum up, Section VI presented two algorithms for MKL on graphs. On the one hand, the algorithm in Section VI-A finds an optimum sum of signals each one belonging to an RKHS in a prespecified dictionary. To obtain a *sparse* estimate, an algorithm was proposed and a solver was derived based on ADMM. On the other hand, the algorithm in Section VI-B first constructs an RKHS by optimally combining multiple kernels within a given dictionary and seeks an estimate within this space. The estimation criterion resulted in a non-convex program which is solved using IIA.

VII. Numerical Tests

This section compares the proposed methods with competing alternatives in synthetic- as well as real-data experiments. Monte Carlo simulation is used to average performance metrics across realizations of the signal $\boldsymbol{f}_0$, noise $\bar{\boldsymbol{e}}$ (only for synthetic-data experiments), and sampling set $\mathcal{S}$. The latter is drawn uniformly at random without replacement from $\{1, \ldots, N\}$.

A. Synthetic Signals

Three experiments were conducted on an Erdős-Rènyi random graph with probability of edge presence 0.25 [1]. Bandlimited signals were generated as in (21) with $\mathcal{B} = \{1, \ldots, B\}$ for a certain B. The coefficients $\{\tilde{f}_n\}_{n\in\mathcal{B}}$ are independent uniformly distributed over the interval $[0, 1]$. Gaussian noise was added to yield a target signal-to-noise ratio SNR $:= ||\boldsymbol{f}_0||^2/(N\sigma_e^2)$.

The first experiment is presented in Fig. 3 and briefly described in Section VI to illustrate the strong impact of the kernel choice on the NMSE $:= \mathbb{E}||\boldsymbol{f}_0 - \hat{\boldsymbol{f}}_0||_2^2/\mathbb{E}||\boldsymbol{f}_0||_2^2$.

The second experiment compares methods for estimating bandlimited signals. Fig. 4 depicts the NMSE in reconstructing a bandlimited signal with $B = 20$ across S. The first two curves correspond to the MKL approaches proposed in Section VI, which employ a dictionary with 5 bandlimited kernels, where the m-th kernel has $\beta = 10^4$ and bandwidth $5m + 5, m = 1, \ldots, 5$. The regularization parameter μ was set to 10^{-1} for RKHS superposition (RS), and to $5 \cdot 10^{-3}$ for kernel superposition (KS). The next three curves correspond to the LS estimator for bandlimited (BL) signals in (22b) [11]–[16]. In order to illustrate the effects of the uncertainty in B, each curve corresponds to a different

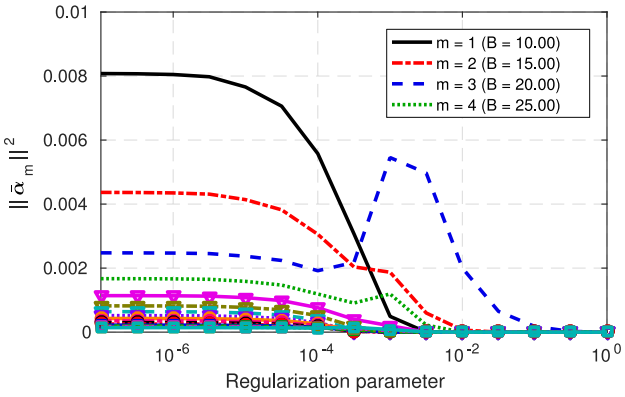


Fig. 5. Sparsity path of the estimate for a typical realization. The legend only displays the first four curves. The last curve to vanish indicates the bandwidth of the observed signal ($S = 80$, $N = 250$, SNR $= 20$ dB).

value of B used for estimation (all estimators observe the same synthetic signal of bandwidth $B = 20$). The last curve pertains to the estimator in [11], which is the LS estimator in (22b) with parameter B set to the cut-off frequency obtained from $\boldsymbol{L}$ and $\mathcal{S}$. To improve its performance, a *proxy* of order 5 [16] is employed instead of the order 1 proxy originally adopted in [11]. To the best of our knowledge, [11] is the only estimator for bandlimited signals to date that does not need bandwidth knowledge. Its performance degrades when $S \gg B$ since the cut-off bandwidth is in the order of S, thus resulting in a noisy estimate of the in-band Fourier coefficients.

Observe in Fig. 4 that although the proposed MKL estimators do not know the bandwidth, their performance is no worse than that of the BL estimator with perfect knowledge of the signal bandwidth. Remarkably, the MKL reconstruction schemes offer a reasonable performance for S small, thus overcoming the need of the LS estimator for $S \geq B$ samples.

The third experiment illustrates how the bandwidth of a graph signal can be estimated using the MKL scheme from Section VI-A. To this end, a dictionary of 17 bandlimited kernels was constructed with $\beta = 10^3$ and uniformly spaced bandwidth between 10 and 90, i.e., $\boldsymbol{K}_m$ is of bandwidth $B_m := 5m + 5$, $m = 1, \ldots, 17$. Fig. 5 depicts the sparsity path for a typical realization of a bandlimited signal with bandwidth $B = 20$. Each curve is obtained by executing Algorithm 1 for different values of μ and represents the squared modulus of the vectors $\{\bar{\boldsymbol{\alpha}}_m\}_{m=1}^M$ in (40) for a different m. As expected, the sparsity effected in the expansion (33) increases with μ, forcing Algorithm 1 to eventually rely on a single kernel. That kernel is expected to be the one leading to best data fit. Since the observed signal is bandlimited, such a kernel is in turn expected to be the one in the dictionary whose bandwidth is closest to B.

Constructing a rule that determines, without human intervention, which curve $||\bar{\boldsymbol{\alpha}}_m||^2$ is the last to vanish is not straightforward since it involves comparing $\{||\bar{\boldsymbol{\alpha}}_m||^2\}_{m=1}^M$ for a properly selected μ. Algorithms pursuing such an objective fall out of the scope of this paper. However, one can consider the naive approach that focuses on a prespecified value of μ and estimates the bandwidth as $\hat{B} = B_{m^*}$, where $m^* = \arg\max_{m\in\{1,\ldots,M\}} ||\bar{\boldsymbol{\alpha}}_m||^2$. Table II reports the performance of such estimator in terms of bias $\mathbb{E}|B - \hat{B}|$ and standard deviation $\sqrt{\mathbb{E}|B - \mathbb{E}\hat{B}|}$ for different values of B for a synthetically generated bandlimited signal.

TABLE II
BIAS AND STANDARD DEVIATION FOR THE NAIVE BANDWIDTH ESTIMATOR WITH $\mu = 10^{-2}$ ($S = 80$, $N = 250$, SNR $= 20$ dB)

	B = 10	B = 20	B = 30	B = 40	B = 50	B = 60
BIAS	0.0	0.6	0.5	0.4	0.4	3.6
STD	0.0	1.9	2.9	1.4	1.4	10.5

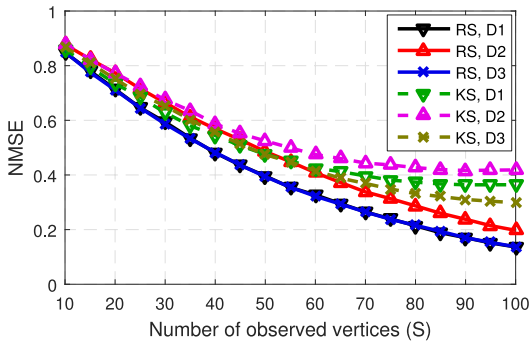


Fig. 6. Comparison of the MKL algorithms with different dictionaries. D1 and D2 are dictionaries respectively containing diffusion and regularized kernels, whereas D3 is the union of D1 and D2.

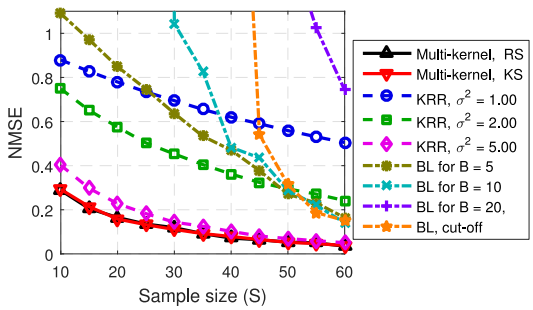


Fig. 7. Generalization NMSE for the data set in [46].

The last experiment of this section assesses the impact of dictionary selection when a non-bandlimited signal is to be reconstructed. Specifically, an exponentially decaying signal [16, eq. (43)] was generated with parameter $r = 20$. Three dictionaries were constructed: D1 contains 5 diffusion kernels with parameter σ uniformly spaced between 0.1 and 0.5; D2 contains 5 regularized Laplacian kernels (see Section IV-A1) with parameter σ uniformly spaced in the same interval; and D3 contains all the kernels in D1 and D2. Fig. 6 illustrates that for both RKHS superposition and kernel superposition, D1 results in a better performance than D2. As expected, the performance of the hybrid D3 is always as good as D1 or D2, which demonstrates that the proposed MKL methods are capable of selecting the best kernels even in presence of heterogeneous dictionaries.

B. Real Data

This section tests the performance of the proposed methods with two real-data sets. In both experiments, the data set is split into a training set used to learn the edge weights, and a test set from which the observations $\bar{\boldsymbol{y}}$ are drawn for performance

TABLE III
GENERALIZATION NMSE AND ROOT MEAN SQUARE ERROR FOR THE EXPERIMENT WITH THE AIRPORT DATA SET [45]

	KRR with cov. kernel	Multi-kernel, RS	Multi-kernel, KS	BL for $B = 2$	BL for $B = 3$	BL, cut-off
NMSE	0.34	0.44	0.43	1.55	32.64	3.97
RMSE [min]	3.95	4.51	4.45	8.45	38.72	13.50

evaluation. Different from the synthetic-data experiments in Section VII-A, where the generated noiseless function f_0 is available and therefore the reconstruction NMSE can be measured on observed and unobserved vertices, the experiments in this section measure generalization NMSE solely at unobserved vertices.

The first data set comprises 24 signals corresponding to the average temperature per month in the intervals 1961–1990 and 1981–2010 measured by 89 stations in Switzerland [46]. The training set contains the first 12 signals, which correspond to the interval 1961–1990, whereas the test set contains the remaining 12. Each station is identified with a vertex and the graph is constructed by applying the algorithm in [47] with parameters $\alpha = 1$ and $\beta = 30$ to the training signals. Based on samples of a test signal on S vertices, the goal is to estimate the signal at the remaining $N - S$ vertices. NMSE is averaged across the 12 test signals for a randomly chosen set $\mathcal{S}$. Fig. 7 compares the performance of the MKL schemes from Section VI along with single-kernel ridge regression (KRR) and estimators for bandlimited signals. The MKL algorithms employ a dictionary comprising 10 diffusion kernels with parameter σ^2 uniformly spaced between 1 and 20. Single-kernel ridge regression uses diffusion kernels for different values of σ^2. Fig. 7 showcases the performance improvement arising from adopting the proposed multi-kernel approaches.

The second data set contains departure and arrival information for flights among U.S. airports [45], from which the $3 \cdot 10^6$ flights in the months of July, August, and September of 2014 and 2015 were selected. A graph was constructed with vertices representing the $N = 50$ airports with highest traffic. An edge was placed between a pair of vertices if the number of flights between the associated airports exceeds 100 within the observation window. A signal was constructed per day averaging the arrival delay of *all* inbound flights per selected airport. Thus, a total of 184 signals were considered, the first 154 were used for training (July, August, September 2014, and July, August 2015), and the remaining 30 for testing (September 2015).

Since it is reasonable to assume that the aforementioned graph approximately satisfies the Markovian property (cf. Section IV-B), a Markov random field was fit to the observations. To this end, the signals were assumed Gaussian so as to estimate the covariance matrix of the observations via maximum likelihood with constraints imposing the (n, n')-th entry of the inverse covariance matrix to be zero if $(v_n, v_{n'}) \notin \mathcal{E}$. Specifically, $\boldsymbol{S} := \boldsymbol{C}^{-1}$ was found by solving the following convex program:

$$\min_{\boldsymbol{S} \in \mathbb{R}^{N \times N}} \ \operatorname{Tr}(\boldsymbol{S}\hat{\boldsymbol{C}}^{-1}) - \log\det(\boldsymbol{S}) \qquad (43)$$

$$\text{s.to} \quad \boldsymbol{S} \succeq \boldsymbol{0}, (\boldsymbol{S})_{n,n'} = 0 \quad \forall (v_n, v_{n'}) \notin \mathcal{E}$$

where $\hat{\boldsymbol{C}}$ is the sample covariance matrix of the training signals after normalization to effect zero mean and unit variance per entry of $\boldsymbol{f}_0$. The inverse of $\boldsymbol{S}$ was used as a covariance kernel (see Section IV-B). Note that such a kernel will only be nearly optimal since the *true* data covariance is unknown.

Employing Laplacian kernels or applying estimators for bandlimited signals requires a Laplacian matrix. Although the edge set $\mathcal{E}$ has already been constructed, it is necessary to endow those edges with weights. Since our efforts to obtain a reasonable estimation performance over the graphs provided by the method in [47] turned out unsuccessful, a novel approach was developed. Specifically, the Laplacian matrix is sought as the minimizer of $||\boldsymbol{L} - \boldsymbol{S}||_F^2$, where $\boldsymbol{S}$ is the solution to (43) and $\boldsymbol{L}$ is a valid Laplacian with a zero at the (n, n')-th position if $(v_n, v_{n'}) \notin \mathcal{E}$. Due to space limitations, the rationale and details behind this approach are skipped.

Table III lists the NMSE and root mean-square error in minutes for the task of predicting the arrival delay at 40 airports when the delay at a randomly selected collection of 10 airports is observed. The second column corresponds to the ridge regression estimator that uses the nearly-optimal *estimated* covariance kernel. The next two columns correspond to the multi-kernel approaches in Section VI with a dictionary of 30 diffusion kernels with values of σ^2 uniformly spaced between 0.1 and 7. The rest of columns pertain to estimators for bandlimited signals. Table III demonstrates the good performance of covariance kernels as well as the proposed multi-kernel approaches relative to competing alternatives.

VIII. CONCLUSION

This paper introduced kernel-based learning as a unifying framework subsuming a number of existing graph signal estimators. SPoG notions such as bandlimitedness, graph filtering, and the graph Fourier transform were accommodated under this perspective. The notion of bandlimited kernel was considered to establish that LS estimators are limiting versions of the ridge regression estimator with Laplacian kernels. Optimality of vertex-covariance kernels was also revealed and a novel interpretation of kernel regression on graphs was presented in terms of Markov random fields. Graph filters were found tantamount to kernel-based smoothers, which suggested applying the former to implement the latter in a decentralized fashion. Finally, numerical experiments corroborated the validity of the theoretical findings.

Future research will pursue algorithms for learning graph Laplacian matrices tailored for regression, broadening regression to directed and dynamic graphs, and numerical experiments with further data sets.

APPENDIX A
PROOF OF THE REPRESENTER THEOREM

Theorem 1 can be proved upon decomposing $\boldsymbol{f}$ according to the following result.

Lemma 1: If $\boldsymbol{\Phi}$ is as in Section III and f belongs to $\mathcal{H}$, then $\boldsymbol{f} := [f(v_1), \ldots, f(v_N)]^T$ can be expressed as

$$\boldsymbol{f} = \boldsymbol{K}\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}} + \boldsymbol{K}\boldsymbol{\beta} \tag{44}$$

for some $\bar{\boldsymbol{\alpha}} \in \mathbb{R}^S$ and $\boldsymbol{\beta} \in \mathbb{R}^N$ satisfying $\boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\beta} = \boldsymbol{0}$.

Proof: Since $f \in \mathcal{H}$, there exists $\boldsymbol{\alpha}$ such that $\boldsymbol{f} = \boldsymbol{K}\boldsymbol{\alpha}$. Thus, one needs to show that, for a given $\boldsymbol{\alpha}$, it is possible to choose $\bar{\boldsymbol{\alpha}}$ and $\boldsymbol{\beta}$ satisfying $\boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{K}\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}} + \boldsymbol{K}\boldsymbol{\beta}$ and $\boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\beta} = \boldsymbol{0}$. This is possible, for instance, if one fixes $\boldsymbol{\beta} = \boldsymbol{\alpha} - \boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}}$ and shows that there exists an $\bar{\boldsymbol{\alpha}}$ such that $\boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\beta} = \boldsymbol{\Phi}\boldsymbol{K}(\boldsymbol{\alpha} - \boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}}) = \boldsymbol{0}$. This, in turn, follows if one establishes that $\boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}}$ always admits a solution in $\bar{\boldsymbol{\alpha}}$, which holds since $\mathcal{R}\{\boldsymbol{\Phi}\boldsymbol{K}\} = \mathcal{R}\{\boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\Phi}^T\}$. To see this, consider the eigendecomposition $\boldsymbol{K} = \boldsymbol{U}_{\boldsymbol{K}}\boldsymbol{\Lambda}_{\boldsymbol{K}}\boldsymbol{U}_{\boldsymbol{K}}{}^T$ and note that

$$\begin{aligned}\mathcal{R}\{\boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\Phi}^T\} &= \mathcal{R}\{\boldsymbol{\Phi}\boldsymbol{U}_{\boldsymbol{K}}\boldsymbol{\Lambda}_{\boldsymbol{K}}\boldsymbol{U}_{\boldsymbol{K}}{}^T\boldsymbol{\Phi}^T\} = \mathcal{R}\{\boldsymbol{\Phi}\boldsymbol{U}_{\boldsymbol{K}}\boldsymbol{\Lambda}_{\boldsymbol{K}}{}^{1/2}\} \\ &= \mathcal{R}\{\boldsymbol{\Phi}\boldsymbol{U}_{\boldsymbol{K}}\boldsymbol{\Lambda}_{\boldsymbol{K}}\boldsymbol{U}_{\boldsymbol{K}}{}^T\} = \mathcal{R}\{\boldsymbol{\Phi}\boldsymbol{K}\}\end{aligned} \tag{45}$$

which concludes the proof. ■

Lemma 1 essentially states that, for arbitrary $\mathcal{S}$, any $f \in \mathcal{H}$ can be decomposed into two components as $f = f_{\mathcal{S}} + f_{\perp}$. The first can be expanded in terms of the vertices indexed by $\mathcal{S}$ as $f_{\mathcal{S}}(v) = \sum_{s=1}^{S}\bar{\alpha}_s\kappa(v, v_{n_s})$ whereas the second vanishes in the sampling set, i.e., $f_{\perp}(v_s) = 0\ \forall s \in \mathcal{S}$. Conversely, it is clear that any function that can be written as in (44) for arbitrary $\bar{\boldsymbol{\alpha}}$ and $\boldsymbol{\beta}$ belongs to $\mathcal{H}$. Hence, Lemma 1 offers an alternative parameterization of $\mathcal{H}$ in terms of $\bar{\boldsymbol{\alpha}}$ and $\boldsymbol{\beta}$. Thus, the minimizer $\hat{\boldsymbol{\alpha}}$ of (7) can be obtained as $\hat{\boldsymbol{\alpha}} = \boldsymbol{\Phi}^T\hat{\bar{\boldsymbol{\alpha}}} + \hat{\boldsymbol{\beta}}$, where

$$\begin{aligned}(\hat{\bar{\boldsymbol{\alpha}}}, \hat{\boldsymbol{\beta}}) := \underset{\bar{\boldsymbol{\alpha}}, \boldsymbol{\beta}}{\arg\min}\ & \mathcal{L}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{y}}, \boldsymbol{\Phi}\boldsymbol{K}(\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}} + \boldsymbol{\beta})) \\ & + \mu\Omega\left([(\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}} + \boldsymbol{\beta})^T\boldsymbol{K}(\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}} + \boldsymbol{\beta})]^{1/2}\right) \\ \text{s.to}\quad & \boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\beta} = \boldsymbol{0}.\end{aligned} \tag{46}$$

Since $\mathcal{L}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{y}}, \boldsymbol{\Phi}\boldsymbol{K}(\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}} + \boldsymbol{\beta})) = \mathcal{L}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{y}}, \boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}})$, the first term in the objective does not depend on $\boldsymbol{\beta}$. On the other hand, since Ω is increasing and

$$(\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}} + \boldsymbol{\beta})^T\boldsymbol{K}(\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}} + \boldsymbol{\beta}) = \bar{\boldsymbol{\alpha}}^T\boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}} + \boldsymbol{\beta}^T\boldsymbol{K}\boldsymbol{\beta} \tag{47}$$

it follows that the objective of (46) is minimized for $\boldsymbol{\beta} = \boldsymbol{0}$, which shows that $\hat{f}_0$ in (6) can be written as $\hat{\boldsymbol{f}}_0 = \boldsymbol{K}\hat{\boldsymbol{\alpha}} = \boldsymbol{K}\boldsymbol{\Phi}^T\hat{\bar{\boldsymbol{\alpha}}}$, thus completing the proof.

APPENDIX B
BIG DATA SCENARIOS

Evaluating the $N \times N$ Laplacian kernel matrix in (14) incurs complexity $\mathcal{O}(N^3)$, which does not scale well with N. This appendix explores two means of reducing this complexity. Both rely on solving (8) rather than (11) since the former employs $\boldsymbol{K}^{\dagger} = \boldsymbol{U}r(\boldsymbol{\Lambda})\boldsymbol{U}^T$, whereas the latter needs $\boldsymbol{K}$.

Recall from Section IV-A1 that Laplacian kernels control the smoothness of an estimate by regularizing its Fourier coefficients $|\tilde{f}_n|$ via r. Computational savings can be effected if one is willing to finely tune the regularization only for large n, while allowing a coarse control for small n. Specifically, the key idea here is to adopt a function of the form

$$r(\lambda_n) = \begin{cases} d\lambda_n & \text{if } 1 < n \leq B \\ d_n & \text{if } n > B \text{ or } n = 1 \end{cases} \tag{48}$$

where d and d_n are constants freely selected over the ranges $d, d_1 > 0$ and $d_n > -\lambda_n$ for $n > B$. Note that (48) can be employed, in particular, to promote bandlimited estimates of bandwidth B by setting $\{d_n\}_{n=B+1}^{N}$ sufficiently large. Defining $\boldsymbol{U}_B^c$ as the matrix whose columns are the $N - B$ principal eigenvectors of $\boldsymbol{L}$, one obtains

$$\boldsymbol{K}^{-1} = d\boldsymbol{L} + \boldsymbol{U}_B^c(\boldsymbol{\Delta} - d\boldsymbol{\Lambda}_B^c)\boldsymbol{U}_B^{c\,T} + d_1\boldsymbol{1}\boldsymbol{1}^T + \epsilon\boldsymbol{I}_N \tag{49}$$

where $\boldsymbol{\Delta} := \text{diag}\{d_{B+1}, \ldots, d_N\}$ and $\epsilon\boldsymbol{I}_N$ with $\epsilon > 0$ is added to ensure that $\boldsymbol{K}$ is invertible in case that the multiplicity of the zero eigenvalue of $\boldsymbol{L}$ is greater than one, which occurs when the graph has multiple connected components.

Alternative functions that do not require eigenvector computation are low-order polynomials of the form

$$r(\lambda) = \sum_{p=0}^{P} a_p\lambda^p. \tag{50}$$

In this case, the resulting $\boldsymbol{K}^{-1}$ reads as

$$\boldsymbol{K}^{-1} = a_0\boldsymbol{I}_N + \sum_{p=1}^{P} a_p\boldsymbol{L}^p. \tag{51}$$

The cost of obtaining this matrix is reduced since powers of $\boldsymbol{L}$ can be efficiently computed when $\boldsymbol{L}$ is sparse, as is typically the case. In the extreme case where $P = 1$, $a_1 > 0$, and $a_0 \to 0$, the regularizer becomes $\boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f}$, which corresponds to Laplacian regularization (cf. Section IV-A1).

APPENDIX C
PROOF OF PROPOSITION 1

Without loss of generality, let $\mathcal{B} = \{1, \ldots, B\}$; otherwise, simply permute the order of the eigenvalues. Define also the $N \times B$ matrix $\boldsymbol{\Psi} := [\boldsymbol{I}_B, \boldsymbol{0}]^T$ and the $N \times (N - B)$ matrix $\boldsymbol{\Psi}_c = [\boldsymbol{0}, \boldsymbol{I}_{N-B}]^T$, whose concatenation clearly satisfies $[\boldsymbol{\Psi}, \boldsymbol{\Psi}_c] = \boldsymbol{I}_N$. Since in this case $\boldsymbol{U}_{\mathcal{B}} = \boldsymbol{U}\boldsymbol{\Psi}$, (22b) becomes

$$\hat{\boldsymbol{f}}_{\text{LS}} = \boldsymbol{U}\boldsymbol{\Psi}[\boldsymbol{\Psi}^T\boldsymbol{U}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{\Psi}]^{-1}\boldsymbol{\Psi}^T\boldsymbol{U}^T\boldsymbol{\Phi}^T\bar{\boldsymbol{y}}. \tag{52}$$

On the other hand, the ridge regression version of (8) is

$$\hat{\boldsymbol{f}}_0 := \underset{\boldsymbol{f}}{\arg\min}\ \frac{1}{S}||\bar{\boldsymbol{y}} - \boldsymbol{\Phi}\boldsymbol{f}||^2 + \mu\boldsymbol{f}^T\boldsymbol{K}^{-1}\boldsymbol{f} \tag{53}$$

where the constraint has been omitted since $r_{\beta}(\lambda_n) > 0\ \forall n$. The minimizer of (53) is

$$\hat{\boldsymbol{f}}_0 = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \mu S\boldsymbol{K}^{-1})^{-1}\boldsymbol{\Phi}^T\bar{\boldsymbol{y}} \tag{54a}$$

$$= \boldsymbol{U}(\boldsymbol{U}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{U} + \mu S r_{\beta}(\boldsymbol{\Lambda}))^{-1}\boldsymbol{U}^T\boldsymbol{\Phi}^T\bar{\boldsymbol{y}}. \tag{54b}$$

Establishing that $\hat{\boldsymbol{f}}_0 \to \hat{\boldsymbol{f}}_{\text{LS}}$ therefore amounts to showing that the right-hand side of (52) converges to that of (54b). For this, it suffices to prove that

$$(\boldsymbol{G} + \mu S r_\beta(\boldsymbol{\Lambda}))^{-1} \to \boldsymbol{\Psi}[\boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi}]^{-1} \boldsymbol{\Psi}^T \tag{55}$$

where $\boldsymbol{G} := \boldsymbol{U}^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{U}$. Note that $\boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi} = {\boldsymbol{U}_{\mathcal{B}}}^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{U}_{\mathcal{B}}$ is invertible by hypothesis. With $\boldsymbol{\Lambda}_B := (1/\beta)\boldsymbol{I}_B$ and $\boldsymbol{\Lambda}_B^c := \beta \boldsymbol{I}_{N-B}$ representing the in-band and out-of-band parts of $r_\beta(\boldsymbol{\Lambda})$, the latter can be written as $r_\beta(\boldsymbol{\Lambda}) = \text{diag}\{\boldsymbol{\Lambda}_B, \boldsymbol{\Lambda}_B^c\}$. With this notation, (55) becomes

$$\left(\begin{bmatrix} \boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi} & \boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi}_c \\ {\boldsymbol{\Psi}_c}^T \boldsymbol{G} \boldsymbol{\Psi} & {\boldsymbol{\Psi}_c}^T \boldsymbol{G} \boldsymbol{\Psi}_c \end{bmatrix} + \mu S \begin{bmatrix} \boldsymbol{\Lambda}_B & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Lambda}_B^c \end{bmatrix} \right)^{-1} \to \begin{bmatrix} (\boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi})^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}. \tag{56}$$

Using block matrix inversion formulae, it readily follows that the left-hand side equals the following matrix product

$$\begin{bmatrix} \boldsymbol{I}_B & -(\boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi} + \mu S \boldsymbol{\Lambda}_B)^{-1} \boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi}_c \\ \boldsymbol{0} & \boldsymbol{I}_{N-B} \end{bmatrix} \begin{bmatrix} (\boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi} + \mu S \boldsymbol{\Lambda}_B)^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_B & \boldsymbol{0} \\ -{\boldsymbol{\Psi}_c}^T \boldsymbol{G} \boldsymbol{\Psi} (\boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi} + \mu S \boldsymbol{\Lambda}_B)^{-1} & \boldsymbol{I}_{N-B} \end{bmatrix} \tag{57}$$

where

$$\boldsymbol{M} := {\boldsymbol{\Psi}_c}^T \boldsymbol{G} \boldsymbol{\Psi}_c + \mu S \boldsymbol{\Lambda}_B^c - {\boldsymbol{\Psi}_c}^T \boldsymbol{G} \boldsymbol{\Psi} (\boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi} + \mu S \boldsymbol{\Lambda}_B)^{-1} \boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi}_c. \tag{58}$$

Recalling that $\boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{\Psi}$ is invertible and letting $\beta \to \infty$, it follows that $\boldsymbol{M}^{-1} \to \boldsymbol{0}$ and $\boldsymbol{\Lambda}_B \to \boldsymbol{0}$ as $\beta \to \infty$, which implies that (57) converges to the right-hand side of (56) and concludes the proof.

Appendix D
Proof of Proposition 3

The first-order optimality condition for (12a) is given by

$$\sigma_e^2 \boldsymbol{C}^{-1} \boldsymbol{f} = \boldsymbol{\Phi}^T (\bar{\boldsymbol{y}} - \boldsymbol{\Phi} \boldsymbol{f}). \tag{59}$$

Without loss of generality, one can focus on the relation implied by the first row of (59). To this end, partition $\boldsymbol{C}$ as

$$\boldsymbol{C} = \begin{bmatrix} c_{1,1} & \boldsymbol{c}_{2:N,1}^T \\ \boldsymbol{c}_{2:N,1} & \boldsymbol{C}_{2:N,2:N} \end{bmatrix} \tag{60}$$

and apply block matrix inversion formulae to obtain

$$\boldsymbol{C}^{-1} = \frac{1}{c_{1|2:N}} \begin{bmatrix} 1 & -\boldsymbol{c}_{2:N,1}^T \boldsymbol{C}_{2:N,2:N}^{-1} \\ -\boldsymbol{C}_{2:N,2:N}^{-1} \boldsymbol{c}_{2:N,1} & \tilde{\boldsymbol{C}}^{-1} \end{bmatrix} \tag{61}$$

where $c_{1|2:N} := c_{1,1} - \boldsymbol{c}_{2:N,1}^T \boldsymbol{C}_{2:N,2:N}^{-1} \boldsymbol{c}_{2:N,1}$ and

$$\tilde{\boldsymbol{C}}^{-1} := \boldsymbol{C}_{2:N,2:N}^{-1} \boldsymbol{c}_{2:N,1} \boldsymbol{c}_{2:N,1}^T \boldsymbol{C}_{2:N,2:N}^{-1} + c_{1|2:N} \boldsymbol{C}_{2:N,2:N}^{-1}. \tag{62}$$

Note that $\sigma_{n|\mathcal{N}_n}^2 := c_{1|2:N}$ is in fact the variance of the LMMSE predictor for $f_0(v_1)$ given $f_0(v_2), \ldots, f_0(v_N)$.

Two cases can be considered for the first row of (59). First, if $1 \notin \mathcal{S}$, then the first row of $\boldsymbol{\Phi}$ is zero, and the first row of (59) becomes

$$f(v_1) = \boldsymbol{c}_{2:N,1}^T \boldsymbol{C}_{2:N,2:N}^{-1} \boldsymbol{f}_{2:N} \tag{63a}$$

$$= \sum_{n: v_n \in \mathcal{N}_1} (-c_{1|2:N} \gamma_{1,n}) f(v_n) \tag{63b}$$

where $\boldsymbol{f}_{2:N} := [f(v_2), \ldots, f(v_N)]^T$ and $\gamma_{n,n'} := (\boldsymbol{C}^{-1})_{n,n'}$. The sum in (63b) involves only the neighbors of v_1 since the graph is a Markov random field, for which if there is no edge between v_n and $v_{n'}$, then $f_0(v_n)$ and $f_0(v_n)$ are conditionally independent given the rest of vertices, which in turn implies that $\gamma_{n,n'} = 0$. Note that the right-hand side of (63a) is the LMMSE predictor of $f(v_1)$ given the estimated function value at its neighbors. Since this argument applies to all vertices $v_n,\ n \notin \mathcal{S}$, it follows that the optimality condition (59) seeks values of $f(v_n)$ so that the function value at unobserved vertices agrees with its LMMSE estimate given the estimated value at its neighbors.

On the other hand, if $1 \in \mathcal{S}$, the first row of $\boldsymbol{\Phi}$ has a 1 at the $(1,1)$ position, which implies that the first row of (59) is

$$y_1 = f(v_1) + \frac{\sigma_e^2}{c_{1|2:N}} (f(v_1) - \boldsymbol{c}_{2:N,1}^T \boldsymbol{C}_{2:N,2:N}^{-1} \boldsymbol{f}_{2:N}). \tag{64}$$

The second term on the right can be thought of as an estimate of the noise e_1 present in y_1. Therefore, the optimality condition imposes that each observation $y_{s(n)}$ agrees with the estimated noisy version of the function given the neighbors of v_n.

References

[1] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. Berlin, Germany: Springer-Verlag, 2009.

[2] 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.

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

[4] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

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

[6] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.

[7] O. Chapelle, V. Vapnik, and J. Weston, "Transductive inference for estimating values of functions," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, vol. 12, 1999, pp. 421–427.

[8] C. Cortes and M. Mohri, "On transductive regression," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, 2007, pp. 305–312.

[9] 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.

[10] J. Lafferty and L. Wasserman, "Statistical analysis of semi-supervised regression," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, 2007, pp. 801–808.

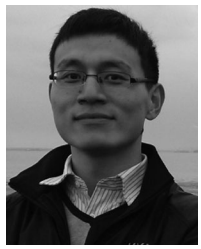
[11] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, Canada, 2013, pp. 5445–5449.

[12] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Austin, TX, USA, 2013, pp. 491–494.

[13] A. Gadde and A. Ortega, "A probabilistic interpretation of sampling theory of graph signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Brisbane, Australia, Apr. 2015, pp. 3257–3261.

[14] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4845–4860, Sep. 2016.
[15] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.
[16] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, Jul. 2016.
[17] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2432–2444, May 2015.
[18] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1832–1843, Apr. 2016.
[19] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.
[20] G. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *J. Math. Anal. Appl.*, vol. 33, no. 1, pp. 82–95, 1971.
[21] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *Proc. Annu. Conf. Learn. Theory*, Banff, Canada, Jul. 2004, vol. 3120, pp. 624–638.
[22] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, Sep. 2015.
[23] G. Wahba, *Spline Models for Observational Data (CBMS-NSF Regional Conference Series in Applied Mathematics)*, vol. 59. Philadelphia, PA, USA: SIAM, 1990.
[24] X. Zhu, J. Kandola, Z. Ghahramani, and J. D. Lafferty, "Nonparametric transforms of graph kernels for semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, 2004, pp. 1641–1648.
[25] 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.
[26] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S. Kandola, "On kernel-target alignment," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, 2001, pp. 367–373. [Online]. Available: http://papers.nips.cc/paper/1946-on-kernel-target-alignment.pdf
[27] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Computational Learning Theory*. Berlin, Germany: Springer-Verlag, 2001, pp. 416–426.
[28] C. Carmeli, E. De Vito, A. Toigo, and V. Umanita, "Vector valued reproducing kernel Hilbert spaces and universality," *Anal. Appl.*, vol. 8, no. 1, pp. 19–61, 2010.
[29] V. N. Vapnik, *Statistical Learning Theory*, vol. 1. New York, NY, USA: Wiley, 1998.
[30] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *Proc. ICML Workshop Statist. Relational Learn. Connections Other Fields*, Banff, Canada, Jul. 2004, vol. 15, pp. 67–68.
[31] P. A. Forero, K. Rajawat, and G. B. Giannakis, "Prediction of partially observed dynamical processes over networks via dictionary learning," *IEEE Trans. Signal Process.*, vol. 62, no. 13, pp. 3305–3320, Jul. 2014.
[32] R. M. Gray, *Toeplitz and Circulant Matrices: A Review*. Delft, The Netherlands: Now, 2006.
[33] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics Series)*. Berlin, Germany: Springer-Verlag, 2006. [Online]. Available: https://books.google.com/books?id=kTNoQgAACAAJ
[34] J.-A. Bazerque and G. B. Giannakis, "Nonparametric basis pursuit via kernel-based learning," *IEEE Signal Process. Mag.*, vol. 28, no. 30, pp. 112–125, Jul. 2013.
[35] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: From transductive to semi-supervised learning," in *Proc. Int. Conf. Mach. Learn.*, 2005, pp. 824–831.
[36] V. Sindhwani, W. Chu, and S. S. Keerthi, "Semi-supervised Gaussian process classifiers," in *Proc Int. Joint Conf. Artif. Intell.*, 2007, pp. 1059–1064.
[37] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol. I: Estimation Theory*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.
[38] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *J. Mach. Learn. Res.*, vol. 9, pp. 485–516, 2008.
[39] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
[40] J.-A. Bazerque, G. Mateos, and G. B. Giannakis, "Group-lasso on splines for spectrum cartography," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4648–4663, Oct. 2011.
[41] G. B. Giannakis, Q. Ling, G. Mateos, I. D. Schizas, and H. Zhu, "Decentralized learning for wireless communications and networking," 2016, arXiv:1503.08855.
[42] C. A. Micchelli and M. Pontil, "Learning the kernel function via regularization," in *Proc. J. Mach. Learn. Res.*, 2005, pp. 1099–1125.
[43] A. Argyriou, M. Herbster, and M. Pontil, "Combining graph Laplacians for semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 2005, pp. 67–74.
[44] C. Cortes, M. Mohri, and A. Rostamizadeh, "l_2 regularization for learning kernels," in *Proc. Conf. Uncertainty Artif. Intell.*, Montreal, Canada, Jun. 2009, pp. 109–116.
[45] Bureau of Transportation, Washington, DC, USA, 2016. [Online]. Available: http://www.transtats.bts.gov/
[46] *Federal Office of Meteorology and Climatology MeteoSwiss*, 2014. [Online]. Available: http://www.meteoswiss.admin.ch/home/climate/past/climate-normals/climate-diagrams-and-normal-values-per-station.html
[47] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," 2014, arXiv:1406.7842.

Daniel Romero (M'16) received the M.Sc. and Ph.D. degrees from the University of Vigo, Vigo, Spain, in 2011 and 2015, respectively. In July 2015, he joined the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. His research interests include the areas of statistical signal processing, communications, and machine learning.



Meng Ma received the B.Eng. degree in communication engineering from Tianjin University, Tianjin, China, in 2012, the M.Eng. degree in control engineering from Shanghai Jiao Tong University, Shanghai, China, in 2015, and the M.Sc. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2015. Since August 2015, he has been working toward the Ph.D. degree with SPiNCOM in the Department of ECE, University of Minnesota, Minneapolis, MN, USA. His research interests include the areas of signal processing and network science.

Georgios B. Giannakis (F'97) received the Diploma in electrical engineering from the National Technical University of Athens, Zografou, Greece, in 1981, and the M.Sc. degree in electrical engineering, the M.Sc. degree in mathematics, and the Ph.D. degree in electrical engineering, all from the University of Southern California, Los Angeles, CA, USA, in 1983, 1986, and 1986, respectively. From 1987 to 1998, he was with the University of Virginia, and since 1999, he has been a Professor in the University of Minnesota, Minneapolis, MN, USA, where he holds an Endowed Chair in wireless telecommunications, a University of Minnesota McKnight Presidential Chair in ECE, and serves as the Director of the Digital Technology Center.

His general interests span the areas of communications, networking, and statistical signal processing—subjects on which he has published more than 390 journal papers, 670 conference papers, 25 book chapters, 2 edited books, and 2 research monographs (h-index 119). His current research focuses on learning from big data, wireless cognitive radios, and network science with applications to social, brain, and power networks with renewables. He is the (co-)inventor of 28 patents issued. He received eight best paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award (2015). He is a Fellow of EURASIP, and has served the IEEE in a number of posts, including a Distinguished Lecturer for the IEEE-SP Society.