Technical Report: Using Laplacian Methods, RKHS Smoothing Splines and Bayesian Estimation as a framework for Regression on Graph and Graph Related Domains

Eduardo Corona, Terran Lane, Curtis Storlie, Joshua Neil

June 11, 2008

Contents

1 Laplacian Methods: An Overview

We first present an overview of the Laplacian operator, its properties, and the methods that arise from analizing its eigenvalues and eigenfunctions. We mainly draw our material from [BLS00] and [Chu97], trying to combine the linear algebra approach with the more analytic one, which suggests the analogue with the continuous Laplacian Operator on manifolds.

1.1 Definition: The Laplacian operator of a Graph

Let $G = (V, E)$ be a simple (no loops or multiple edges), undirected graph, $|V| = n$, $|E| = m$. Let F_V the space of functions $f : V \to \mathbb{R}$, that is, the real functions whose domain is the nodes of G. We can identify through obvious isomorphisms V with $\{1, ..., n\}$ and F_V with $\mathbb{R}^n$. To allow for a generalization to infinite cases and a direct analogue with a continuous domain framework, we restrict ourselves to studying the subspace $l^2(V)$, that is $\{f \in F_V \mid \int |f|^2 \, d\nu = \sum_{i=1}^{n} |f(i)|^2 < \infty\}$, in the understanding that $n, m < \infty \implies F_V = l^2(V)$.

There are two main linear operators (matrices) that are typically used to work with this space of functions; each uniquely determines and represents G :

1. **The adjacency matrix** A: A nxn matrix, it assigns $a_{ij} = 1$ if v_i and v_j are linked by an edge, and 0 otherwise. We can think of A as an element of $L(l^2(V))$ (that is, the space of linear operators of the space $l^2(V)$). If we denote $j \to i$ as the set of indexes j such that v_j is linked with v_i, then the action of this operator is:

$$(Af)_i = \sum_{j \to i} f(j) \ \forall i \in \{1, ..., n\} \tag{1.1}$$

2. **The incidence matrix** M : A mxn matrix, it assigns $m_{ij} = 1$ if $v_j \in E_i$, 0 otherwise. We can think of M as an operator that provides us with a vector in $\mathbb{R}^m$, where:

$$(Mf)_i = \sum_{v_j \in E_i} f(j) \ \forall i \in \{1, ..., m\} \tag{1.2}$$

However, we are more interested in defining some sort of "discrete differential" linear operators on this space of functions as in **[BLS00]**. This allows us to inherit the power of regression,interpolation and approximation methods, as well as results of the theory of differential equations on continuous domains:

1. **Gradient Matrix** ∇: This is a mxn matrix, closely linked to the incidence matrix (in fact, in **[BLS00]**, they also call it that). Given an arbitrary orientation of the nodes of G (or a directed graph G), it assigns $a_{ij} = 1$ if $v_j \in E_i$ and v_j is the terminal vertex of E_i, -1 if $v_j \in E_i$ and v_j is the initial vertex of E_i and 0 otherwise. ∇f gives us a vector in $\mathbb{R}^m$ of differences of values at each edge. It is also known as the "Coboundary mapping" of G, since it maps information on $0-chains$ (nodes) to information in $1-chains$ (edges):

$$(\nabla f)_i = f(v_{TER_i}) - f(v_{INI_i}) \ \forall i \in \{1, ..., m\} \tag{1.3}$$

2. **Laplacian Matrix Δ and Second differences:** The conditions to properly define what a second difference on v_i would require us to have edges $E_b = (v_b, v_i)$ and $E_a = (v_i, v_a)$ (such that they both have the same direction, in a sense). In the case of a graph composed only of eulerian walks, we could define:

$$(\partial^2 f)_{i,a,b} = (\nabla f)_a - (\nabla f)_b = f(v_a) - 2f(v_i) + f(v_b) \tag{1.4}$$

We first note that this definition doesn't change if we "reverse" both edge orientations. We now define a "Laplacian" linear operator on $l^2(V)$, which in this case adds all the second differences on each vertex v_i (where $a \to i \to b$ means we have v_a, v_i and v_b connected in that particular order):

$$(\Delta f)_i = \sum_{a \to i \to b} (\partial^2 f)_{i,a,b} = \sum_{a \to i \to b} f(v_a) - 2f(v_i) + f(v_b) = \sum_{j \to i} [f(j) - f(i)] \tag{1.5}$$

We can extend this last expression to any simple graph G, and for convenience we define the Laplacian operator L as:

$$(Lf)_i = -(\Delta f)_i = \sum_{j \to i} [f(i) - f(j)] = \deg(v_i) f(i) - \sum_{j \to i} f(j) = (Df - Af)_i \tag{1.6}$$

Where $\deg(v_i)$ is the degree of v_i, and $D = diag(v_i)$. So, $L = D - A$, hence it is a symmetric operator. Another interesting construction of L (which also coincides with its continuous analog) is to define:

$$L = \nabla^T \nabla \tag{1.7}$$

In any case, the entries of matrix L are $l_{ij} = -1$ if v_i and v_j are linked by an edge, $l_{ii} = \deg(v_i)$, and 0 otherwise.

This second construction also immediately buys us a discrete version of Green's formula:

$$\langle Lf, g \rangle = \langle f, Lg \rangle = f^\top \nabla^\top \nabla g = \langle \nabla f, \nabla g \rangle \tag{1.8}$$

And L is a **symmetric positive semidefinite** operator (just like it's continuous analog), because:

$$\langle f, Lf \rangle = \langle \nabla f, \nabla f \rangle = \|\nabla f\|^2_{\mathbb{R}^m} = \sum_{(v_i, v_j) \in E} (f(i) - f(j))^2 \geq 0 \tag{1.9}$$

This will allow us to use L or L^2 as smoothing operators, since we will be able to use expressions of the form $\|\nabla f\|^2_{\mathbb{R}^m}$ and $\|Lf\|^2_{\mathbb{R}^n}$ based on this operator.

Finally, like in (Chung) we define a normalized version of this operator, which we call $\widetilde{L}$. We do this such that $\widetilde{L} = \widetilde{\nabla}^T \widetilde{\nabla} = D^{-1/2} \nabla^\top \nabla D^{-1/2}$ and:

$$(\widetilde{L} f)_i = \frac{1}{\sqrt{d_i}} \sum_{j \to i} \left[\frac{f(i)}{\sqrt{d_i}} - \frac{f(j)}{\sqrt{d_j}} \right] = (D^{-1/2}(D - A)D^{-1/2} f)_i \tag{1.10}$$

This provides useful properties and it forces the spectrum of $\widetilde{L}$ to be a subset of $[0, 2]$. The normalized Laplacian is therefore also a SPS operator, and it only differs from L in the way we define the first differences. Unless otherwise stated, we will use this as our standard Laplacian operator.

1.2 Properties of the Laplacian and its Spectrum

1.2.1 Spectrum of L and $\widetilde{L}$: Graph eigenvalues and eigenvectors:

Because these operators are symmetric, we know that they are diagonalizable. That means, we can find an orthonormal base of $l^2(V)$ of eigenvectors $X =$

$\{\phi_1, \phi_2, ..., \phi_n\}$ of L and $\widetilde{X}$ of $\widetilde{L}$. The specific construction of these operators as analogs of the continuous Laplacian operator identifies these eigenvectors as the Fourier basis, and will allow us to inherit the global smoothing and representation capabilities of their counterparts. Let the spectrum of L be $\{0 \leq \lambda_1 < \lambda_2 < ... < \lambda_k\}$. Then, we can define the according Rayleigh quotient, and we know that:

$$\lambda_1 = \inf_{f \in l^2(V)} \frac{\langle f, Lf \rangle}{\langle f, f \rangle} = \inf_{f \in l^2(V)} \frac{\|\nabla f\|^2}{\|f\|^2} = \inf_{f \in l^2(V)} \frac{\sum_{(v_i, v_j) \in E} (f(i) - f(j))^2}{\sum_{i=1}^{n} f(i)^2} \tag{1.11}$$

In this case, we can see that:

1. $\boldsymbol{\lambda_1 = 0}$: If f is constant, then the Rayleigh quotient is zero, and we know its always non-negative.
2. **The multiplicity of λ_1 is equal to the number of connected components of G** : If the Rayleigh quotient is zero, necessarily $f(i) = f(j)$ $\forall (i, j) \in E$. This forces the function to be constant on each connected component of the graph. The space of such functions therefore has a dimension equal to the number of components, and $\langle f, Lf \rangle = 0 \iff Lf = 0 \iff f$ is an eigenvector associated with λ_1.
3. **The corresponding eigenvectors are constant functions on each component.**

The positive eigenvalues and their corresponding eigenvectors correspond (both in intuition and if we can define the according difference equations) to the natural vibrational "frecuencies" of the graph, and share many properties with the continuous Fourier basis (more detail in [BLS00]) . One that is particularly interesting to us is the variational characterization of these eigenvectors as the functions that progressively minimize the norm of the gradient on the orthogonal complement of the span of the previous eigenvectors $\langle v_1, ..., v_{p-1} \rangle$:

$$\lambda_p = \inf_{f \in \langle v_1, ..., v_{p-1} \rangle^\top} \frac{\|\nabla f\|^2}{\|f\|^2} \tag{1.12}$$

The second eigenvalue λ_2 (smallest positive eigenvalue) is particularly valuable as a measure of how well connected the graph is, and its eigenpair (f_2, λ_2) has been used extensively in graph separation algorithms.

We now show that $\widetilde{L}$ inherits most of these properties from L, and that it also provides us with additional properties on the set of eigenvalues. Let $g \in l^2(V)$, $f = D^{-1/2} g$. The corresponding Rayleigh coefficient is:

$$\lambda_1 = \inf_{g \in l^2(V)} \frac{\left\langle g, \widetilde{L} g \right\rangle}{\langle g, g \rangle} = \inf_{f \in l^2(V)} \frac{\langle f, Lf \rangle}{\left\langle D^{1/2} f, D^{1/2} f \right\rangle} = \inf_{f \in l^2(V)} \frac{\|\nabla f\|^2}{\left\| D^{1/2} f \right\|^2} = \inf_{f \in l^2(V)} \frac{\sum_{(v_i, v_j) \in E} (f(i) - f(j))^2}{\sum_{i=1}^{n} f(i)^2 \deg(v_i)} \tag{1.13}$$

$$\lambda_1 = \inf_{g \in l^2(V)} \frac{\left\langle g, \widetilde{L} g \right\rangle}{\langle g, g \rangle} = \inf_{g \in l^2(V)} \frac{\left\| \widetilde{\nabla} g \right\|^2}{\|g\|^2} \tag{1.14}$$

It follows directly from the un-normalized case that:

1. $\lambda_1 = 0$
2. The multiplicity of λ_1 is equal to the number of connected components of G
3. The corresponding eigenvectors are functions of the form $D^{1/2}\chi$, with χ constant on each component.

We inherit the same kind of properties and variational characterization as with the unnormalized version, where we can either think of out eigenvectors as the functions that progressively minimize the gradient defined by $\widetilde{\nabla} g$ over the usual norm, or as $g = D^{1/2} f$, where we minimize the usual gradient using a weighted norm for f (it is a semi-norm if we have isolated vertices).

Either for the normalized and un-normalized Laplacian, this tells us two key things:

1. Aside from providing terms for smoothing and a basis, the Laplacian operators also allow us to represent "smooth" functions on the graph using a sparse representation. If the gradient of a function in $l^2(V)$ is small, we can assume it's well represented by a combination of the first p eigenvectors $\{\phi_1, ..., \phi_p\}$. **We want to know exactly how adequate this representation is.**
2. The Graph Laplacian is a topologically aware operator. This means it gives us a priori information on our domain, which allows us to speed up many computations. In particular, knowing the multiplicity of λ_1 allows us to write L as a block-diagonal matrix, and to work on each subgraph independently. We are also interested in using **other properties** of the spectrum, such as λ_2 (which tells us how well connected the graph is), and the existence and multiplicity of $\lambda = 1$ (which is particularly relevant for random walks on the graph and therefore diffission wavelets). Finally, as we will learn in the following sections, using these eigenvalues will be essential to assign "weights" to the basis functions according to their frequency content or how relevant they are in representing smooth functions.

1.2.2 Other interesting / useful properties of the normalized Laplacian (Chung):

1. $\sum_{i=1}^{n} \lambda_i \leq n$ with equality $\iff$ G has no isolated vertices.
2. For $n > 1$, $\lambda_2 \leq \frac{n}{n-1}$ with equality $\iff$ G is the complete graph. If we don't have isolated vertices, we also have $\lambda_n \geq \frac{n}{n-1}$.

3. If G is not complete, $\lambda_2 \leq 1$

4. For all λ_i with $i < n$, $\lambda_i \leq 2$ and $\lambda_n = 2 \iff$ a connected component is bipartite and non-trivial

5. There are lots of sharper bounds on λ_2 in the literature.

1.2.3 Laplacians of Weighted Graphs and Generalized Laplacians as Elliptic Operators

1. A weighted undirected graph G is associated with a weight function $w : V \times V \to \mathbb{R}$ and the corresponding weight matrix W ($W_{ij} = w(v_i, v_j)$). This weight function is required to be nonnegative (where $w_{ij} > 0 \iff i$ and j are connected) and symmetric ($w_{ij} = w_{ji}$). We then redefine $\deg(v_i) = \sum_{j=1}^{n} w_{ij}$, and then define the un-normalized and normalized Laplacians:

$$L_w = D - W \tag{1.15}$$
$$\widetilde{L}_w = D^{-1/2}(D - W)D^{-1/2} \tag{1.16}$$

We immediately notice that this construction generalizes the unweighted case, and it preserves the properties of the Laplacian operators. These are symmetric operators (because we required W to be symmetric) and if we look at the resulting Rayleigh coefficients:

$$\frac{\langle f, L_w f\rangle}{\langle f, f\rangle} = \frac{\sum_{(v_i,v_j)\in E}(f(i) - f(j))^2 w_{ij}}{\sum_{i=1}^{n} f(i)^2} \tag{1.17}$$
$$\frac{\left\langle g, \widetilde{L_w} g\right\rangle}{\langle g, g\rangle} = \frac{\langle f, L_w f\rangle}{\left\langle D^{1/2}f, D^{1/2}f\right\rangle} = \frac{\sum_{(v_i,v_j)\in E}(f(i) - f(j))^2 w_{ij}}{\sum_{i=1}^{n} f(i)^2 \deg(v_i)} \tag{1.18}$$

We realize that they are also symmetric positive semidefinite, and that virtually all the results from the unweighted case can be extended or adapted (particularly, the results on the multiplicity of λ_1 are exactly the same). It is interesting to note that we can also define analogous "gradient" operators ∇_w and $\widetilde{\nabla}_w$ such that:

$$\begin{array}{cc} (\nabla_w f)_i = (f_{TER_i} - f_{INI_i})\sqrt{w_i} & \widetilde{\nabla}_w = D^{-1/2}\nabla_w \\ L_w = \nabla_w^{\top}\nabla_w & \widetilde{L}_w = \widetilde{\nabla}_w^{\top}\widetilde{\nabla}_w \end{array} \tag{1.19}$$

And therefore, our eigenvectors in this sense retain the desired smoothing properties. To gain intuition, we can think of a line-graph or a mesh on a copy of $\mathbb{R}^n$, and using $w_{ij} = 1/d(v_i, v_j)^2$. This would provide the usual finite difference formulas for the gradient and the Laplacian on the selected mesh.

2. Another approach that generalizes the unweighted Laplacian operators is the family of Generalized Laplacians or Discrete Schrodinger Operators on a graph G. To G we associate a symmetric matrix M such that $m_{ij} < 0 \iff i$ and j are linked by an edge, and $m_{ij} = 0$ if i and j are distinct and not linked. We call any of such matrices M a "generalized Laplacian", and we can decompose it using the form $M = P - W$, where P is the diagonal of M, and W can be thought of as an elliptic operator on non-homogeneous media, or the weight matrix of a weighted graph. The physical interpretation of these operators is that they are Hamiltonians for a certain system, where Pf is the potential energy and $-Wf$ the kinetic energy:

$$(Mf)_i = \sum_{j \to i} (f(i) - f(j)) w_{ij} + p_{ii} f(i) \tag{1.20}$$

If we use the base of eigenvectors to do function representation, approximation and global smoothing, we should choose the Laplacian which best fits our graph model and the way we interpret "smoothness" through the construction of the according first and second difference operators.

1.3 Other operators / Basis

One could legitimately ask if there are other operators or other basis / smoothing penalty construction methods, especially if we want to approximate functions which we know are not expected to be globally smooth, but still have local smoothness or some other expected properties. There is work in the literature on a wavelet basis ([MM07] and [BCMS04]) and its use to approximate locally smooth functions. We could also use analogues to polynomial basis, or finite element basis, especially if we also have data on the edges. Another question we have is if we can define higher order differential operators on a graph, or operators similar to these Laplacian operators when our domain is a hypergraph. A compilation of the work in this direction can be found in [ABB06].

1.4 Calculus on Graphs:

In their recent publications ([FT08] and [FJP02]) Friedman & Tillich have succesfully developed a framework to do calculus on Γ the geometric realization of a graph G. That is, the metric space resulting of mapping the nodes to points and edges $E = (v_i, v_j)$ to closed intervals of length l_{ij} between them. Under certain conditions, these metric spaces may be isometrically embedded in a copy of $\mathbb{R}^N$ for some N. However, we can also think of them as simplicial complexes with only 0 and 1 chains.They argue that the use of edgewise-linear functions on Γ corresponds to its previous use in graph theory, and that by introducing nonlinear functions it is easier to translate results from analysis on manifolds to graph theory.

Within this framework, it is natural to define edgewise-differentiable functions on Γ, and their gradient as a function as a vector field on the tangent

bundle of the graph, defined only in the interiors of its edges. Their first key result is that concepts such as Laplacians and their Rayleigh quotients on graphs require the use of two "volume" measures:

1. A counting measure ν on the vertex set, which assigns $\nu(v) > 0 \ \forall v \in V$
2. A "piecewise Lebesgue" measure E, which assigns $E(v) = 0 \ \forall v \in V$ and whose restriction to any edge interior e is the Lebesgue measure times a constant a_e.

The differences in the Rayleigh quotients are:

1. We think of the gradient as a number assigned to the whole edge, and define the gradient as a difference of the values in the edges. Therefore our Rayleigh quotient uses counting measures for both edges and vertices, and is:
$$R(f) = \frac{\int \nabla f^2 dE}{\int f^2 d\nu} = \frac{\sum_{(u,v)\in E}(f(u) - f(v))^2}{\sum_{v\in V} f^2(v)} \tag{1.21}$$
2. If we use the piecewise Lebesgue measure, and consider general functions over the graph realization, we get:
$$R(f) = \frac{\int |\nabla f|^2 \, dE}{\int f^2 d\nu} = \frac{\sum_{e\in E} \int_e |\nabla f|^2 \, d\lambda}{\sum_{v\in V} f^2(v)} \tag{1.22}$$
The most interesting part about this quotient is that, if we fix the values on the nodes, it is minimized exactly when f is edgewise-linear, and we recover the eigenvectors and eigenvalues from the graph Laplacian as the restriction of these functions to the nodes of Γ. The Rayleigh quotients are also the same, modulo the constants a_e/l_e.

It is interesting to consider this approach for further research for the following reasons:

1. We can think of our current regression problem as a smoothing splines problem with linear splines on Γ, especially in the case $m = 1$ where the penalties coincide.
2. We can define non-linear functions over nodes and edges of a graph, and apply the framework of higher order splines or approximate these functions using the Laplacian's Fourier basis.
3. This also grants the power to translate inequalities and results of Sobolev Spaces.

4. On [FJP02], an integrating factor and a mixed edge-vertex Laplacian and Rayleigh quotient are defined, which might have repercutions in interpolation / regression over vertices and edges.

5. We can think about the implications of this framework to extend it to hypergraph domains.

2 Our main goal: A Function Regression / Approximation Problem on a Graph

2.1 Formulation and Solution

Methods based on the Graph Laplacian operator, particularly those based on the properties of its eigenvalues, have been proposed to study graph domains analitically, and to solve problems such as graph separation, chromatic number, spectral clustering and graph embedding. However, until recently there has been considerably less focus on the uses of the corresponding basis of eigenvectors to perform function approximation, and therefore exploit the power of the Laplacian as an operator on the space of functions on the graph. Some previous work in this direction can be found in [MM07] as a comparison to approximation with a wavelet basis, and in [Mah05] as an application to policy representation of Markov Decision Processes. A general review of Laplacian eigenvector function approximation can be found in [BLS00].

We are interested in the following regression problem: Given a simple undirected graph $G = (V, E)$, and observations $\{g_i\}_{i \in I}$ of a real function f on the nodes of G, with $I \subseteq V$ a potentially arbitrary index set of nodes, we want to provide a smooth estimate of this function f, and to interpolate its values on unobserved nodes.

A specific approach that applies this problem to solve classification, link prediction and ranking problems can be found in [ZS04]. Our main contribution is to show that this smooth regression problem can be posed as a special case of the general Reproducing Kernel Hilbert Spaces (RKHS) Smoothing Splines problem [Wah90]. Using this alternative formulation, we can export thoroughly developed and validated techniques from the smoothing splines literature. Taking motivation from this example and from real-life problems, we also want to extend this framework to include functions with domain in G or $G \times H$, with H a Hilbert space with certain properties, that assume categorical, real or vector values. These observations may be taken to be either exact, or noisy. For example, we will often assume that our observations come from the well-known gaussian additive model, that is:

$$\{g_i = f(x_i) + \varepsilon_i\}_{i \in I} \ ; \ \varepsilon_i \sim N(0, \sigma^2)$$

It is also of extreme relevance to note that the use of the Laplacian operator in this framework allows us to explicitly incorporate the topology of the domain in the solution of these problems. This is the key property that substantially

separates these methods from the ones typically used in the literature and also what allows us to draw simple, yet powerful and well justified techniques.

In any case, we would like to have a general framework to provide an approximate or an estimate of the true underlying function f. If we have evidence that f is "smooth" in the sense we have defined, or if we can transform our problem into one of smooth approximation or estimation, we can apply this framework and obtain solutions to an immense variety of real-life problems, providing competitive algorithms. To our knowledge, this can be readily applied to a myriad of problems that arise in the study of genetic data, biological data, sensor networks, and probabilistic distributions on graph structures such as BNs or MRFs. It is also our belief that the use of Laplacian methods in graph related problems can potentially allow building techniques for problems in graph domains as powerful as some of their counterparts in continuous domains (Fourier Transform, Wavelets, Filtering, Diffusion and Wave equations, etc).

We present the simplest regression problem on a graph, given observations on certain nodes. As we have indicated, we use the base of eigenvectors of the Laplacian to model our function, and we also use it as a discrete differential operator to add a smoothing penalty. We provide the closed form of the solution of a least squares regression problem with this penalty incorporated. As we will see later on, this is in more than one way a natural analogue version of the smooth regression problem in a continuous setting. The first problem uses our entire basis to fit the model, and the second uses only the first q eigenvectors:

Problem 2.1 *Let $G = (V, E)$ be an undirected simple graph with $|V| = n$, $f : V \to \mathbb{R}$ a function defined over the graph's vertices. We have a sample of $\{g_{ij}\}_{j=1}^{k}$ observations of g, on the corresponding set $\{v_{i_j}\}_{j=1}^{k}$ of k distinct vertices on the graph. Let $L = L(G)$ the normalized graph Laplacian, $X = \{\phi_1|...|\phi_n\}$, ϕ_i eigenvectors of L, $\Lambda = diag(\lambda_i)$ and Z the submatrix which contains the rows $\{x_{i_j}\}_{j=1}^{k}$ of X, that is, the ones that correspond to nodes where we have observations. We want to obtain $\widehat{\beta}$, the solution of the following least squares problem (where c,$m \geq 0$ are regularization constants):*

$$\min_{\beta \in \mathbb{R}^n} \{\frac{1}{p} \sum_{i=1}^{p} \|Z\beta - g_i\|^2 + c(X\beta)^\top L^m X\beta\} = \min_{\beta \in \mathbb{R}^n} \{\frac{1}{n} \sum_{i=1}^{p} \|Z\beta - g_i\|^2 + c\beta^\top \Lambda^m \beta\} \tag{2.1}$$

The closed-form solution to this problem is

$$\widehat{\beta_{c,m}} = (Z^\top Z + c\Lambda^m)^\dagger Z^\top \overline{g} \tag{2.2}$$

Where $\overline{g} = \frac{1}{p}\sum_{i=1}^{p} g_i$, $\widehat{g_{c,m}} = Z(Z^\top Z + c\Lambda^m)^\dagger Z^\top \overline{g}$ is the linear estimate of f in the observed nodes given $\{g_{i_j}\}_{j=1}^{k}$ and $\widehat{f_{c,m}} = X(Z^\top Z + c\Lambda^m)^\dagger Z^\top \overline{g}$ is the corresponding estimate of f over our entire graph G.

Solution 2.2 *We first write the objective function of our minimization problem as a sum of interior products:*

$$\begin{aligned}\frac{1}{p}\sum_{i=1}^{p}\|Z\beta - g_i\|^2 + c(X\beta)^\top L^m X\beta &= \frac{1}{p}\sum_{i=1}^{p}\langle Z\beta - g_i, Z\beta - g_i\rangle + c\beta^\top X^\top X\Lambda^m X^\top X\beta \\ &= \frac{1}{p}\sum_{i=1}^{p}[\beta^\top Z^T Z\beta - 2Z\beta^\top g_i + g_i^\top g_i] + c\beta^T\Lambda^m\beta \\ &= \beta^\top(Z^\top Z + c\Lambda^m)\beta - 2(Z\beta)^\top\frac{1}{p}\sum_{i=1}^{p}g_i + \frac{1}{p}\sum_{i=1}^{p}g_i^\top g_i\end{aligned}$$

This yields a quadratic equation on β, and its gradient is:

$$(Z^\top Z + c\Lambda^m)\beta - Z^\top\overline{g}$$

Therefore, its critical points are solutions of the system of equations $(Z^\top Z + c\Lambda^m)\beta = Z^\top\overline{g}$. The least squares solution (or solution of minimum norm) is:

$$\widehat{\beta_{c,m}} = (Z^\top Z + c\Lambda^m)^\dagger Z^\top\overline{g}$$

And we know it's a global minimum because $(Z^\top Z + c\Lambda^m)$ is a sum of symmetric positive semidefinite matrices (and therefore, the quadratic is pseudoconvex). The estimates in the observed nodes and for the function in the entire graph follow from this result.

Problem 2.3 *Given the same conditions as in Problem 1, let Φ be the submatrix of X whose columns are the eigenvectors corresponding to the q smallest eigenvectors, Ψ the corresponding submatrix of Z. We want to estimate the function f with $\widehat{f}$ in the span of these first q eigenvectors, which leads to the following minimization problem:*

$$\min_{\beta\in\mathbb{R}^q}\{\frac{1}{p}\sum_{i=1}^{p}\|\Psi\widehat{\alpha} - g_i\|^2 + c(\Phi\widehat{\alpha})^\top L^m\Phi\widehat{\alpha}\} \tag{2.3}$$

The closed-form solution is then given by $\widehat{\alpha_{c,m}} = (\Psi^\top\Psi + c\Lambda^m)^\dagger\Psi^\top\overline{g}$, and the corresponding estimates are $\widehat{g_{c,m}} = \Psi\widehat{\alpha_{c,m}}$, $\widehat{f_{c,m}} = \Phi\widehat{\alpha_{c,m}}$. Its derivation is identical to the one for problem 1.

Some initial remarks we can make about the solution of these problems are that their closed form solutions are the same for all Generalized Laplacian Operators, and that the "smoothing" penalty penalizes each coefficient β_i according to a power of its frequency $c\lambda_i^m$. Here the parameter c is allowing for a balance between fit and smoothing on the least squares minimization, and m allows us to penalize different degrees of "roughness" or "smoothness" on the estimate $\widehat{f}$ of our model function. Letting $c \to 0$ will provide the solution of the least

squares fit problem, and letting $c \to \infty$ will yield a constant function on each connected component of G.

As we will see, this has a direct link with smoothing splines in continuous domains, where the penalty is a Sobolev semi-norm of order m (m not necessarily an integer) and the resulting spline estimate $\widehat{f} \in H^m(\Omega)$ and is piece-wise polinomial. Within the framework we have built, we can conclude the following:

$$\begin{array}{lcc} m=0 & f^\top f & \text{regularity (encourages small coefficients)} \\ m=1 & f^\top L f = \|\nabla f\|^2 & \text{encourages small first differences} \\ m=2 & f^\top L^2 f = \|Lf\|^2 & \text{encourages small second differences} \end{array} \tag{2.4}$$

We can explore in further analysis if $m > 2$ or $m \in \mathbb{R}^+$ has a specific meaning in terms of smoothness, but for now it is enough to say that we can think of it as penalizing each coefficient according to the *mth* power of its respective eigenvalue. As it is common practice in the continuous smoothing splines literature, we can use $m = 2$ or $m = 1$ as a standard choice for this parameter which corresponds to a "reasonable" balance between smoothness asumptions and fit.

Conjecture 2.4 *Let $m = 1$. Then $R(f) = \dfrac{f^\top L f}{\|f\|^2} = \dfrac{\beta^\top \Lambda \beta}{\beta^\top \beta} = \sum_{i=1}^n \left[\dfrac{\beta_i^2}{\|\beta\|^2}\right] \lambda_i$ is a convex combination of $\{\lambda_i\}_{i=1}^n$ (Penalty$(f) = c\|f\|^2 R(f)$). We could maybe calculate these normalized coefficients on our observation vectors to choose an adequate set of eigenvectors to estimate our solution.*

3 RKHS: Smoothing Splines Framework

The main strength of our approach is to merge the Laplacian-based methods as explained in the previous sections with the powerful and well established theory of smoothing splines on Reproducing Kernel Hilbert Spaces. The realization that the smooth regression problem on a graph is in fact a special case within this theory allows us to export many of the powerful tools that have been refined and validated in the smoothing splines and gaussian bayesian estimation literature, and thus to obtain a novel and versatile approach to this and related regression problems. We first present the introductory material for RKHS theory, as presented in [BTA04] and [Wah90].

Reproducing Kernel Hilbert Space methods (RKHS) have been widely used in statistics applications to approximate observational data with smooth splines or smooth regression methods. To restrict ourselves to this type of Hilbert Spaces is natural to tackle these problems, in the sense that they don't only provide the strength and power of working with a vector space with an inner product, but they are spaces in which the evaluation and other "observation" functionals turn out to be continuous. In other words, if two functions f and

g are close in the inner product norm, for any x our observed values (in this case $f(x)$ and $g(x)$) are also close. This is a strong assumption if we are working in an infinite dimentional space: it does not hold either in $L^2([a,b])$ or in $(C([a,b]), \|\cdot\|_\infty)$.

Even for finite dimentional domains, where these functionals are trivially continuous, it also provides a useful framework for the swift calculation of numerical solutions, and it provides natural links with bayesian estimation problems, which we will see in more detail in the following section. It also allows us to deal with product spaces in a very convenient way.

We first introduce the basic results of RKHS and smoothing splines, and immediately point to their application to our regression problem on G:

3.1 Symmetric Positive Functions and RKHS

3.1.1 Definitions:

Definition 3.1 *Let T be an arbitrary index set. In our regression problem, $T = V$ which is isomorphic to $\{1, 2, ..., n\}$. Then a symmetric, real-valued function $K(s,t)$ is positive definite if, for any $\{a_i\}_{i=1}^k \subset \mathbb{R}$ and any $\{t_i\}_{i=1}^k \subset T$,*

$$\sum_{i,j=1}^{k} a_i a_j K(t_i, t_j) \geq 0 \tag{3.1}$$

And it is strictly positive definite if the strict inequality holds. In the finite dimentional case, this is equivalent to requiring that the matrix $K_{ij} = (K(t_i, t_j))$ is SPS or SPD, respectively.

Given a symmetric positive definite K, there are two key identifications we can make:

1. By the Kolmogorov Consistency Theorem, we can always define a family $\{X(t)\}_{t\in T}$ of **zero-mean Gaussian random variables** such that their covariance function is K, that is:

   $$K(s,t) = E[X(s)X(t)] = Cov(X(s)X(t)) \ \forall s, t \in T$$

   In our regression problem, this is equivalent to generate a multivariate normal "vector" on the graph, with Variance-Covariance matrix K. This establishes a bijection between symmetric positive functions and gaussian processes, which is exploited in the smoothing splines and in the bayesian estimation problems.

2. A **real RKHS**: a Hilbert Space H_K of functions $f : T \to \mathbb{R}$ (in our case, it is a subspace of F_V) such that the evaluation linear functionals $L_t(f) = f(t)$ are bounded (continuous). That means:

   $$|L_t f| = |f(t)| \leq M \|f\|_{H_K} \ \ \forall f \in H_K, t \in T \tag{3.2}$$

By Riesz Representation Theorem, this implies that for any evaluation functional there exists an element $K_t \in H_K$ such that:

$$L_t f = f(t) = \langle K_t, f \rangle \ \ \forall f \in H_K \tag{3.3}$$

We can therefore define a unique symmetric positive function $K(s,t)$ in T with the "reproducing" property, that is:

$$K(s,t) = \langle K_s, K_t \rangle \tag{3.4}$$

We can see it is positive definite, since for any $\{a_i\}_{i=1}^k, \{t_i\}_{i=1}^k$:

$$\sum_{i,j} a_i a_j R(t_i, t_j) = \sum_{i,j} \left\langle a_i R_{t_i}, a_j R_{t_j} \right\rangle = \left\| \sum_j a_j R_{t_j} \right\|^2 \geq 0 \tag{3.5}$$

This tells us that for any RKHS H_K, there exists a unique positive definite function $K(s,t)$, which we call its Reproducing Kernel. The converse is also true: for any positive definite function, we can construct a unique RKHS such that K is its reproducing kernel, where H_K is the completion of the space spanned by linear combinations of $R_t = R(t, \cdot)$:

$$H_K = \overline{span(\{R_t\}_{t \in T})} \tag{3.6}$$

This theorem (Moore-Aronszajn) allows us to establish a bijection between positive functions and RKHS, and hence a isometric isomorphism between RKHS and the space of linear combinations of $\{X(t)\}_{t \in T}$. More importantly, in the context where $T = V$, it lets us work with any of our Laplacian operators L to obtain a reproducing kernel of the space $H = l^2(V)$, and will allow us to work with functions in $H \otimes H_K$, with H_K some $RKHS$ associated with functions defined over a continuous or discrete domain. This will enable us to propose a framework to work with more complex problems such as modelling time series on a graph or classification which considers attribute information on each node.

3.1.2 Eigenfunctions of a Kernel K:

Lets assume that $K(s,t)$ is continuous, and that

$$\int\int_{T \times T} K^2(s,t) ds dt < \infty \tag{3.7}$$

In the discrete version, this is trivial, since it only asks for $\|K\|_F^2 = tr(K^2) < \infty$. Then there exists an orthonormal sequence of continuous eigenfunctions $\{\phi_1, \phi_2, ...\} \subset L^2(T)$ and eigenvalues $\lambda_i \geq 0$ such that:

$$\int_T R(s,t)\phi_i(t) dt = \lambda_i \phi_i(s) \tag{3.8}$$

$$R(s,t) = \sum_{i \in I} \lambda_i \phi_i(s) \phi_i(t) \tag{3.9}$$

$$\int\int_{T \times T} R^2(s,t) ds dt = \sum_{i \in I} \lambda_i^2 < \infty \tag{3.10}$$

In the discrete version, these tell us that $K\phi_i = \lambda\phi_i$, $K = X\Lambda X^\top = \sum \lambda_i \phi_i \phi_i^\top$ and $tr(K^2) = \sum \lambda_i^2$.

Lemma 3.2 *If these conditions hold, and we let*

$$\beta_i = \langle f, \phi_i \rangle$$

then

$$f \in H_K \iff \sum_{i \in I} \frac{\beta_i^2}{\lambda_i} = \|f\|_K^2 < \infty \tag{3.11}$$

Which, in the discrete setting, means $f \in H_K \iff f = X\beta$, $\|f\|_K = \beta^\top \Lambda^\dagger \beta < \infty$.

3.1.3 Sum and Tensor Product of Reproducing Kernels

Theorem 3.3 *If we have two reproducing kernels* K_1, K_2 *of spaces* $(H_1, \|\cdot\|_{H_1}), (H_2, \|\cdot\|_{H_2})$ *of functions on* T*, then* $K = K_1 + K_2$ *is the reproducing kernel of the space* $H = H_1 + H_2$ *with:*

$$\|f\|_H^2 = \min_{f = f_1 + f_2} \{\|f_1\|_{H_1}^2 + \|f_2\|_{H_2}^2\} \tag{3.12}$$

If H *is a direct product,* $f = f_1 + f_2$ *is unique and* $\|f\|_H^2 = \|f_1\|_{H_1}^2 + \|f_2\|_{H_2}^2$.

Theorem 3.4 *Let* H_1 *and* H_2 *two reproducing kernel hilbert spaces of functions on* T_1 *and* T_2*, with kernels* K_1, K_2*. The tensor product* $H_1 \otimes H_2$ *is a vector space of functions on* $T_1 \times T_2$ *where:*

$$f_1 \otimes f_2(x_1, x_2) = f_1(x_1) f_2(x_2) \tag{3.13}$$

$$\langle f_1 \otimes f_2, g_1 \otimes g_2 \rangle = \langle f_1, g_1 \rangle_1 \langle f_2, g_2 \rangle_2 \tag{3.14}$$

And it admits a functional completion which is a RKHS with the reproducing Kernel $K = K_1 \otimes K_2$*. This particularly tells us that the product of a finite family of reproducing kernels is a reproducing kernel.*

3.2 Smoothing Splines and RKHS

We first present the special and the general smoothing spline problems as in [Wah90]. Then we proceed to apply the solution of the general problem to our regression problem. In each case, the formulation and solution of the problem depends almost exclusively on representing our space as the direct product of two RKHS.

1. **The Special Smoothing Problem:** We have n fixed points in $\{t_1, ..., t_n\} \subset T = [0, 1]$, and we wish to do interpolation with smooth splines in the Sobolev space $W_m([0,1]) = \{f \mid D^\alpha f \text{ is absolutely continuous } \forall \alpha < m, D^m f \in L^2([0,1])\}$. We have a data model $\{y_i = f(t_i) + \varepsilon_i\}_{i=1}^n$ where $\varepsilon \in N(0, \sigma^2 I)$. The estimate of f is found by solving the problem:

$$\widehat{f} = \arg\min_f \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - f(t_i))^2 + c \int_0^1 \left| f^{(m)}(u) \right|^2 du \right\} \tag{3.15}$$

2. **The General Smoothing Problem:** T is arbitrary, $f \in H_K$ a given RKHS and $\{L_i\}_{i=1}^n$ are bounded linear functionals on H_K (the simplest examples are evaluation functionals or their linear combinations). We have a data model $\{y_i = L_i f + \varepsilon_i\}_{i=1}^n$ where $\varepsilon \in N(0, \sigma^2 I)$ and a decomposition of $H_K = H_0 \oplus H_1$ where $\dim(H_0) = M \leq n$. The estimate of f is found by solving the problem:

$$\widehat{f} = \arg\min_f \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - L_i f)^2 + c \left\| P_1 f \right\|_K^2 \right\} \tag{3.16}$$

   Where P_1 is the orthogonal projection of f onto H_1 in H_K. The way to solve this problem is to write $L_i f = \langle \eta_i, f \rangle$ and obtain the representer using the kernel: $\eta_i(s) = \langle \eta_i, R_s \rangle = L_i R(s, \cdot)$. This ensures that given that $\dim(H_K) \geq n$, we can always construct the solution using as many representer functions as we have observations. In many cases, this allows for an almost straight-forward numerical solution once we have defined our kernels and our problem formulation.

3. **Graph Regression Problem:** $T = V$, $f \in H_K = l^2(V)$ and $\{L_i f = f(v_i)\}_{i=1}^n$. Our data model is exactly the same, and the decomposition (for $m > 0$) goes as follows: $H_K = E_0 \oplus \bigcup_{\lambda_i > 0} E_{\lambda_i} = Ker(L) \oplus Ran(L)$. We know that $\dim(E_0) = M = \#$ connected components of $G \leq n$, and we can define kernels on both spaces using the eigenvalues of the Laplacian operator L. In particular, we need:

$$\left\| P_1 f \right\|_{K_1}^2 = f^T L^m f = \beta^\top \Lambda^m \beta = \sum_{\lambda_i > 0} \lambda_i^m \langle f, \phi_i \rangle^2 \tag{3.17}$$

So λ_i^{-m} are the eigenvalues of K_1 and $X_1 = (\phi_{M+1} \mid .. \mid \phi_n)$ it's corresponding eigenvectors. This means $K_1 = X(\Lambda^\dagger)^m X^\top = X_1 \Lambda_1^{-m} X_1^\top$. If $X_0 = (\phi_1, .., \phi_M)$, we can choose $K_0 = X_0 X_0^\top = X P_{E_0} X^\top$, $\|P_0 f\|_{K_0}^2 = \sum_{\lambda_i=0} \langle f, \phi_i \rangle^2$. In our case, the evaluation functionals are coordinate evaluations, and so the representer η_j is simply the jth column of out kernel $K = K_0 + K_1$. This allows us to rewrite the regression problem as:

$$\widehat{f} = \arg\min_f \left\{ \frac{1}{n} \|y - Tb - K_1 e\|^2 + c e^\top K_1 e \right\} \tag{3.18}$$

We can see now that a solution scheme for the general problem will also provide us with an alternative way to calculate the solution to our graph regression problem.

We state the main theorem in [Wah90] that provides us with a system of linear equations to solve any problem we can write in this general framework:

Theorem 3.5 *Let $\{\phi_1, ..., \phi_M\}$ span the null space of P_1 and let the $n \times M$ matrix T defined by: $t_{ij} = L_i \phi_j$ of full column rank. Then the minimizer f_c of the general problem is:*

$$f_c = \sum_{i=1}^{M} d_i \phi_i + \sum_{i=1}^{n} e_i \xi_i \tag{3.19}$$

Where:

$$\begin{aligned} \xi_j &= P_1 \eta_j \\ d &= (T^\top M^{-1} T)^{-1} T^\top M^{-1} y \\ e &= M^{-1}(I - T(T^\top M^{-1} T)^{-1} T^\top M^{-1}) y \\ M &= \Sigma + ncI \\ \Sigma_{ij} &= \langle \xi_i, \xi_j \rangle \end{aligned} \tag{3.20}$$

An equivalent set of equations to calculate e and d is given by:

$$\begin{aligned} Me + Td &= y \\ T^\top e &= 0 \end{aligned}$$

Which is

$$\begin{pmatrix} M & T \\ T^\top & 0 \end{pmatrix} \begin{pmatrix} e \\ d \end{pmatrix} = \begin{pmatrix} y \\ 0 \end{pmatrix} \tag{3.21}$$

A system of $n + M$ linear equations of the form $Ax = b$, where A is symmetric and of full rank. This allows us to use powerful algorithms like GMRES in the general case, and PCG or Cholesky with minimum degree if A is positive definite. It is of extreme importance to note that the size of this system only depends on the dimention of H_0 and the number of observations.

In our problem if we have a complete basis and all the nodes are observed, $\xi_j = P_1\eta_j = \sum_{k=M+1}^{n} \lambda_k^{-m}\phi_k(j)\phi_k$, $\Sigma = K_1 = X\Lambda^{\dagger m}X^\top$ and $t_{ij} = \phi_j(i) \implies T = X_0$. Therefore:

$$X(\Lambda^{\dagger m} + ncI)X^\top e + X_0 d = y \tag{3.22}$$
$$X_0^\top e = 0 \tag{3.23}$$

In the general case, an "influence matrix" $A(c)$ is defined as a linear prediction of our observations and will be later used to define a mean square prediction error:

$$(L_1\widehat{f}, ..., L_n\widehat{f})^\top = A(c)y \tag{3.24}$$
$$A(c)y = Td + \Sigma e \tag{3.25}$$

In our case, since we are actually looking to estimate just the evaluations at the nodes of G :

$$\widehat{f} = A(c)y = X_0 d + K_1 e \tag{3.26}$$

Which after a long series of calculations, yields:

$$A(c) = X(I + \frac{1}{nc}\Lambda^{\dagger m})X^\top \tag{3.27}$$

This is exactly what we would get from our closed-form solution if we divided the first term of the regression problem by n. In the simplest case, this allows us to choose between calculating the closed-form solution and solving the aforementioned system of linear equations. However, we will see that choosing this approach allows for a swifter calculation. More importantly, in complex cases a closed-form solution of this type will not be available, but the RKHS approach will be readily applicable.

This formulation also allows us to account for unobserved nodes (we simply choose the functionals that correspond to evaluating at observed nodes) and for choosing only the first q eigenvectors. Assuming $q > M$, which justifies the existence of the smoothing penalty, we just define H_1 as the span of $\{\phi_{M+1}, ..., \phi_q\}$ and work with the according kernels.

4 Smoothing Splines on Tensor Product Spaces

We are now equipped to demonstrate the power of the RKHS framework for the graph regression problem. Many real-world graph data sets are quite complex: nodes in a social graph might be annotated with personal information such as gender or age, while nodes in a sensor network register variable data in time series. We want our graph regressor to take advantage of this information, in addition to the graph topology. Unlike the classical formulation presented in section 2, framing the regression problem in the RKHS framework of section 4.2 provides a natural extension to such cases. This has been applied succesfully to

multivariate smoothing splines in continuous domains [Wah90],[Wah03]. To our knowledge, we are the first to acknowledge that graph domains can be included using the RKHS generated by the Graph Laplacian. We are interested in three cases: $F : V \times [a, b] \to \mathbb{R}$, $H : V \times [a, b]^p \to \mathbb{R}$ and $J : V \times V_2 \to \mathbb{R}$. The first one can be used to learn time series on a graph, and we can apply all cases to functions with values on attributed nodes.

4.0.1 The motivating example: Product of Sobolev Spaces $W_2([0, 1])$

First, we present the classical example of biviariate tensor product cubic splines (which basically takes the product of two copies of a Sobolev Space):

Lets take the data model for the special smoothing spline problem, but in two dimensions. So now our space is $T = [0, 1]^2$, and we have data $\{y_{ij} = f(t_i)g(s_j)+ \varepsilon_{ij}\}$ with $f, g \in W_2(0, 1)$, $\varepsilon_{ij} \sim N(0, \sigma^2 I)$. That is, we wish to approximate this function smoothly using $f \otimes g$ in the tensor product $H = W_2([0, 1]) \otimes W_2([0, 1])$ and using the current decompositions and kernels on each space. For each marginal space, we have its norm and the corresponding kernel K:

$$\|u\|^2_{W_2} = \left[u^2(0) + u'^2(0)\right] + \left[\int_0^1 u''^2(t)dt\right] = \|P_0 u\|^2 + \|P_S u\|^2 \quad (4.1)$$

$$K(s, t) = [1 + st] + \left[\int_0^1 (s - z)_+(t - z)_+ dz\right] = K_0(s, t) + K_S(s, t) \quad (4.2)$$

A further decomposition is sometimes suggested, such that $W_2(0, 1) = H_0 \oplus H_P \oplus H_S$, where H_0 is generated by the constant term, H_P is the "parametric" space generated by the map t, and H_S is the "smooth" space. The tensor product space H can be written down as the orthogonal sum of the 9 subspaces $\{H_{0,0}, H_{0,P}, ..., H_{S,S}\}$. It is then logical to penalize only the subspaces that have a "smooth" space as a factor and are therefore infinite dimentional. The general form of that seminorm is:

$$J(u) = \theta_{S,0} \left\|P_{H_{S,0}} u\right\|^2 + \theta_{0,S} \left\|P_{H_{0,S}} u\right\|^2 + ... + \theta_{S,S} \left\|P_{H_{S,S}} u\right\|^2 \quad (4.3)$$

And then we solve the bivariate smoothing splines problem:

$$\min_{f \otimes g \in H} \left\{ \sum_{i,j} (y_{ij} - f(t_i)g(s_j))^2 + cJ(f \otimes g) \right\} \quad (4.4)$$

We can then assign different smoothing parameters to the five subspaces, and we can observe that the choice of norms for each space in the marginal space now changes the form of the penalty. It is also not necessary to include all the spaces with an H_S, and we can also take $H_0 \oplus H_P$ as one space. This makes model selection more relevant. In any case, we collect all the spaces in a "smooth" subspace H_1 and the rest in H_0. Finally, the reproducing kernel of H_1 can be calculated using the following result (as well as the kernel of tensor product spaces theorem):

Theorem 4.1 *If $H = \bigoplus_{i=1}^{p} H_i$ is an orthogonal sum of RKHS with kernels $\{K_i\}_{i=1}^{p}$, the reproducing kernel of H with the weighted norm:*

$$\|u\|^2 = \sum_{i=1}^{p} \theta_i \|P_{H_i} u\|^2 \ ; \ \theta_i > 0 \ \forall i \tag{4.5}$$

Is given by:

$$K(s,t) = \sum_{i=1}^{p} \frac{1}{\theta_i} K_i(s,t) \tag{4.6}$$

Then, we can apply the framework of general smoothing splines to solve this problem, given learned weights and model selection.

4.0.2 Product of $l^2(V)$ with $W_2(0,T)$:

Lets think of the following situation: we have observations on each node of a graph G, on several points in time $\{t_1, ..., t_m\} \subset [0,T]$. We call this observations $\{g_i^{t_j}\}_{i,j=1}^{n,m}$, and we choose the data model: $\{g_i^{t_j} = f(i)h(t_j) + \varepsilon_{ij}\}$, $f \in l^2(V)$ and $h \in W_2(0,T)$ with the corresponding norms and kernels, and $\varepsilon_{ij} \sim N(0, \sigma^2 I)$. A good example where this model could be relevant is the set of observations of a sensor network, which we gather in a certain time interval. It would also be useful to address the problem of collective classification of the nodes of G given one continuous attribute. We then want to approximate these observations and get information at unobserved nodes, taking into account the topology of the graph, and accounting for smoothness both in time and spatial domains. As in the previous example, we have decompositions $W_2(0,T) = W_0 \oplus W_P \oplus W_S$ and $l^2(V) = H_K = H_0 \oplus H_1$. The tensor product space $H = H_K \otimes W_2$ can be then decomposed in an analogous way into the orthogonal sum of six spaces $\{H_{0,0}, H_{0,P}, H_{0,S}, H_{1,0}, H_{1,P}, H_{1,S}\}$.

We again choose to penalize only the 4 subspaces which have one "smooth" component, either H_1 or W_S. The resulting seminorm / penalty is:

$$J(f \otimes h) = \theta_{0,S} \sum_{i=1}^{M} \langle f, \phi_i \rangle^2 \int_0^T h''^2(t)dt + [f^\top L^m f] \left[\theta_{1,0} h^2(0) + \theta_{1,P} h'^2(0) + \theta_{1,S} \int_0^T h''^2(t)dt \right] \tag{4.7}$$

We then wish to solve the corresponding problem:

$$\min_{f \otimes g \in H} \left\{ \sum_{i,j} (g_{ij} - f(i)h(t_j))^2 + cJ(f \otimes h) \right\} \tag{4.8}$$

Where $H = \Xi_0 \oplus \Xi_1$ with $\Xi_0 = H_{0,0} \oplus H_{0,P}, \Xi_1 = H_{0,S} \oplus H_{1,0} \oplus H_{1,P} \oplus H_{1,S}$

and the corresponding kernels are:

$$K_0((i,s),(j,t)) = (X_0 X_0^T)_{ij}[1+st] \tag{4.9}$$

$$K_1((i,s),(j,t)) = \theta_{0,S}^{-1}(X_0 X_0^T)_{ij}\int_0^T (s-z)_+(t-z)_+ dz$$

$$+(X_1\Lambda_1^{-m}X_1^\top)_{ij}\left[\theta_{1,0}^{-1}+\theta_{1,P}^{-1}st+\theta_{1,S}^{-1}\int_0^T (s-z)_+(t-z)_+dz\right] \tag{4.10}$$

Now we are faced with two issues: calculating the solution of this problem using the general smoothing spline framework, and more importantly, defining a scheme to learn the weights $\Theta = \{\theta_{H_i}\}$ for our norm/kernel, or simplify our model so that we only have to learn one or two parameters. We can learn all the parameters on the joint data using a Newton or QuasiNewton method, or learn each parameter while leaving the rest fixed (backfitting) in an iterative way. Both approaches are widely documented in the smoothing splines and statistical literature ([Gu02]).

4.0.3 Alternative approach for data on $V \times [0,T]$: Functional Principal Component Analysis

In some instances, the product smoothing splines framework we have presented might work very well, especially since it takes into account interactions and smoothness both in spatial and temporal domains, and it uses the topology of the graph through the graph Laplacian L. However, we are motivated by the real life examples of volcanic and other sensor network data to present an alternative to the product space method that preprocesses and smoothly approximates temporal data in one step, and uses the Laplacian methods to smoothly interpolate the resulting compact representation over the whole graph in a second step. For the first step, we use the method of Functional Principal Component Analysis. For a more complete account, consult [RS05]. We first briefly present the FPCA framework theory, and then apply it to our case. Then, we show how to insert the resulting components to the corresponding Laplacian method.

The functional version of the PCA is defined when we have samples $\{x_i(s)\}_{i=1}^p$ of data defined over a continuous domain, which without loss of generality we will consider as the interval $[0,T]$. As in the multivariate case, we want to summarize the information provided by our data and reduce the dimentionality by considering only a few linear combinations of the original variables, which are also easier to interpret. We want this components to explain most of the variability we observe on our data.

On the first step we want to find the unitary weight function $\xi_1(s)$ (with norm 1) such that the average sum of squares of the component scores is maxi-

mized. That is:

$$\xi_1(s) = \arg \max_{\xi_1^\top \xi_1 = 1} \{\frac{1}{p} \sum_{i=1}^{p} \left[\int_0^T \xi_1 x_i \right]^2 \} = \arg\max \{\frac{1}{p} \sum_{i=1}^{p} f_{i1}^2\} \tag{4.11}$$

The next weight functions $\xi_i(s)$ are computed in a strictly analogous way to the discrete case: we find the weight function that maximizes the average sum of square scores restricted to the space of functions of norm 1 which are orthogonal to the previous weight functions. This produces what is called an optimal empirical orthonormal basis, and these weight functions can be calculated by solving an infinite dimentional eigenvalue problem as follows:

We want to find a complete set of orthonormal functions $\{\xi_i\}_{i=1}^{\infty}$ such that the expansion of each curve in terms of the first K is the best in a K dimentional subspace. This means that, if we have the expansion

$$\widehat{x}_i(s) = \sum_{j=1}^{K} \langle \xi_j, x_i \rangle \, \xi_j(s) = \sum_{j=1}^{K} f_{ij} \xi_j(s) \tag{4.12}$$

The basis that minimizes the average square error

$$\sum_{i=1}^{p} \int [x_i(s) - \widehat{x}_i(s)]^2 ds \tag{4.13}$$

is exactly the same as the basis functions that progressively maximize variance components as we have defined.

We know the covariance function $v(s,t)$ of our data:

$$v(s,t) = \frac{1}{p} \sum_{i=1}^{p} x_i(s) x_i(t) \tag{4.14}$$

As in the multivariate case, minimizing this variace is equivalent to solving the infinite dimentional eigenvalue problem:

$$(V\xi)(s) = \int v(s,t)\xi(t)dt = \lambda \xi(s) \tag{4.15}$$

One way to implement this numerically is to choose a finite basis $\{\phi_i\}_{i=1}^{K}$ and calculate the basis expansion for each x_i :

$$x_i(t) = \sum_{i=1}^{K} c_{ik} \phi_k(t) \tag{4.16}$$

The whole vector $x = (x_1, ..., x_p)$ can then be written like:

$$x = C\Phi \tag{4.17}$$

Then the covariance function can be written as:

$$v(s,t) = \frac{1}{p}\phi(s)^\top C^\top C\phi(t) \tag{4.18}$$

If we also expand $\xi(s) = \Phi b$, the eigenvalue problem is now:

$$\begin{aligned} \phi(s)^\top \frac{1}{p} C^\top CWb &= \lambda\phi(s)^\top b \\ \frac{1}{p} C^\top CWb &= \lambda b \end{aligned} \tag{4.19}$$

where $w_{ij} = \int \phi_i(t)\phi_j(t)dt$. If we pick an orthonormal basis, such as the Fourier basis or the basis of orthonormal polinomials, then $W = I$ and the eigenvalue problem is the same as in the multivariate case.

Other numerical implementations imply a discretization of the integral equation, or applying an integration quadrature scheme such as the trapezoid or simpson rules.

Smooth FPCA: As in the smoothing spline literature, it is desireable to penalize the "roughness" of our components with a penalty of the form $\|D^m\xi\|^2$ weighted by a smoothing parameter $c > 0$ that we can learn by cross validation. The value for m most commonly used in the literature (also in [RS05]) is again $m = 2$. In this case, we maximize a modified sample variance:

$$\frac{\frac{1}{p}\sum_{i=1}^{p}\left[\int_0^T \xi_1 x_i\right]^2}{\|\xi\|^2 + c\|D^2\xi\|^2} \tag{4.20}$$

From our experience with smoothing splines, we know this is equivalent to pick the first K orthonormal functions that maximize this variance from the space $W_2([0,T])$, with norm $\|\xi\|^2 + c\left\|D^2\xi\right\|^2$. If we wish to model periodic functions in $[0,T]$, we can then use the classic Fourier basis on $[0,T]$ and our problem is reduced to an eigenvalue problem. The main advantage of this approach is that these functions are automatically eigenfunctions for the operators D^2 and $I + D^2$. We can also use a different basis in the non periodic case, and convert the problem again to an eigenvalue problem (both of these are explained in detail in [RS05]). In each case, algorithms such as FFT (periodic case), SVD and Cholesky with minimum degree can be used to reduce the computational complexity of this approach.

An alternative to smooth FPCA would be to use the standard smoothing splines framework on each curve, and then perform standard FPCA on the resulting data. In [RS05], they conclude that although these two approaches produce almost the same results, the latter usually produces rougher results because of the difference in tuning the smoothing parameter. In either case, smoothing is beneficial to produce an estimation that is robust to noise in our data.

Application to the $V \times [0,T]$ product space problem: In our sensor network or general product space problem, we have observations on a proper subset $S \subset V$ of the nodes of the graph G, and on each we have data on a set $\{t_j\}_{j=1}^k \subset T$. In the volcanic data example from Mt. Erebus, this amounts to having up to 40 samples per second on each node. If we call $x_i(s)$ the continuous curve that is sufficiently sampled by the observations $\{x_i(t_j)\}_{j=1}^k$ on each node $v_i \in S$, it makes sense to perform $FPCA$ on our data, choosing one of the two alternatives for smoothing. We now have a compact representation of the information on the sensor nodes through a few components $\xi_l(s)$ and the corresponding scores $f_{il} = \langle \xi_l, x_i \rangle$.

For each $\xi_l(s)$, we can think of the set of scores $\{f_{il}\}_{i=1}^m$ as values of a function over the entire set of nodes V of G. Because we expect the data curves to change "smoothly" through the graph, this can be naturally translated to a smooth change of the scores through the graph. We then apply the simple regression over G to the scores of each component, and construct the missing data curves by writting each as the corresponding linear combination of the components we have chosen. In practice, only a few components are chosen to gain a simple, compact interpretation of the data. However, we might want to try adding a few more components, especially if the number of sensor nodes is a relatively small fraction of the number of nodes in our graph.

A comparison between this method and the product space smoothing splines approach would be useful to determine when to use each of these Laplacian-based methods. We hypothesize that the combination of SFPCA with regression over G will yield a more compact and easier to interpret representation, and the product space approach will yield a more accurate representation and a better fit. However, these two methods would have to be tested in different situations to better asess their performance.

We also plan to test their performance against the method of tensor product smoothing splines on $\{1,..,K\} \times [0,T]$ ([Gu02]) to determine the relevance of taking into account the graph structure of G. Adding this knowledge about the topology of G allows us to build statistical models over domains that are closer to the true underlying domain that we are studying. Therefore, it is our hypothesis that this addition will yield results improved both in accuracy and interpretability.

4.0.4 Product of $l^2(V)$ with $\{f : \{0,1\}^d \to \mathbb{R}\}$ or some other $l^2(W)$

In the machine learning literature, we often encounter the collective classification problem. That is, given a graph $G = (V, E)$, we want to classify each node (which can be seen as function representation over G, especially when we know some of the classes) according to its neighbors classes and using attribute information, often in the form of a bit vector $\{0,1\}^d$ assigned to each node. Interpreting this problem from our smooth regression standpoint: we have an underlying function $f : V \times \{0,1\}^d \to \mathbb{R}$ which is "smooth" in the sense that nodes that are close spatially or have closer attributes should have similar values

(which would account for some case of homophily). We hypothesize that, by changing the smoothing parameter for the nodes, we could actually also account for heterophily.

The way we deal with the vectors in $\{0,1\}^d$ is to think of them as vertices of the unit hypercube in $\mathbb{R}^d$ (a Hamming cube). In this same sense, we can think of them as nodes in the corresponding graph H^d and work with the corresponding Laplacian operator L_H and it's eigenvalues and eigenvectors $\{\psi_i\}_{i=1}^{2^d}$ (which we can precompute for a certain d. Actually, $\Lambda_{H^d} = diag(\{\frac{2k}{d}\}_{k=1}^{d})$ with multiplicity $\binom{n}{k}$).This graph has been studied extensively from a combinatorics perspective, and is amongst other things, $d - regular$ (which in our context means normalized and un-normalized Laplacian eigenvectors coincide) and we can obtain the formulae for it's adjacency and Laplacian matrices recursively. So now, we have a data model: $\{g_i\}_{i=1}^{n}$ where $g_i = f(v_i)h(A(v_i)) + \varepsilon_i$ with $f \in H_G$, $h \in H_{H^d}$ and $A : V \to \{0,1\}^d$ assigns attribute vectors to each node, $\varepsilon \sim N(0, \sigma^2 I)$.

Because each space decomposes into two subspaces $H_G = H_0 \oplus H_1$, $H_{H^d} = B_0 \oplus B_1$ we write $H = H_G \otimes H_{H^d}$ as the sum of the resulting 4 orthogonal subspaces, and penalize only three:

$$J(f \otimes h) = \theta_{0,1} \sum_{i=1}^{M} \langle f, \phi_i \rangle^2 \, h^\top L_H^{m_h} h + [f^\top L^m f] \left[\theta_{1,0} \langle h, \psi_1 \rangle^2 + \theta_{1,1} h^\top L_H^{m_h} h \right] \tag{4.21}$$

To solve the problem:

$$\min_{f \otimes g \in H} \left\{ \sum_{i=1}^{n} (g_i - f(i)h(A(v_i)))^2 + cJ(f \otimes h) \right\} \tag{4.22}$$

We calculate the kernel of this product with the weighted products of the according kernels, and perform model selection / parameter learning following the recomendations on the previous example. This same framework with minor modifications can also enable us to use products of spaces of functions on graphs.

An alternative to this approach is to consider the graph G, and assign weights to each edge based on the Hamming distance of the assigned attributes of its nodes. Assuming similar attributes define a notion of closeness, if two adjacent nodes have the same attributes, we can assign $\epsilon \in (0,1)$, and $d_{H^d}(A(v_i), A(v_j))$ otherwise. We can also take the inverse of these distances to account for the opposite notion: that similar attributes will account for radically different classes. Once we have defined our notion of distance, we can apply our original regression problem using the corresponding weighted Laplacian.

4.0.5 Product of $l^2(V)$ with $W_2[a,b]^d$

We can use our first example ($d = 1$) as a model and take the product of $l^2(V)$ with d copies of $W_2([a,b])$. However, this requires to do careful model selection, since following that scheme blindly would probably increase the number of parameters in our model exponentially. One alternative is to take $W_2([a,b])^d$ as a whole, using some multivariate splines and penalty / kernel model, and build the product with $l^2(V)$ as we have done with the simplest case (like thin plate or partial splines models). Another is to use these product space spline, but to disregard interactions between three or more smooth subspaces as irrelevant or too complicated (and proceed in a similar manner to ANOVA models such as the ones presented in [Wah90], [SBRZ08], [Wah03] and [Gu02]). Finally, we can learn our parameters using optimization techniques such as Newton or Quasinewton methods or backfitting.

5 RKHS: Bayesian Estimation Approach

5.1 Duality between RKHS and Stochastic Processes

(From [Wah90]) Let H_K be a $RKHS$ on T with kernel K. Then, as we have mentioned before, we can identify this space with a family $\{X_t\}_{t\in T}$ of zero-mean Gaussian random variables such that $E[X(s)X(t)] = K(s,t)$. We can then establish a bijection between H_K and the Hilbert space X spanned by $\{X_t\}_{t\in T}$, which we will use to relate our smoothing splines problems with bayesian estimation problems.

The standard way to produce X is to start with the space of random variables of the form

$$Z = \sum_{j=1}^{m} a_j X(t_j) \ ; \ \{t_j\}_{j=1}^{m} \subset T \tag{5.1}$$

That is, the preHilbert space of finite linear combinations of variables in our family, with the inner product $\langle Z_1, Z_2\rangle = E[Z_1 Z_2]$. We complete this space with all of its quadratic mean limits, and obtain X. This space is isometrically isomorphic to H_K, and the variables $X(t_j)$ correspond naturally to the columns $K_{t_j} = K(t_j, \cdot)$ of the Kernel. More importantly, if we have a certain bounded linear functional L with representer η, we can think of it as $\eta = \lim_{m\to\infty} \sum_{j=1}^{m} a_j K_{t_j}$ and identify it via isomorphism with the random variable $LX = \lim_{m\to\infty} \sum_{j=1}^{m} a_j X(t_j)$. It is important to note that the use of L in this last term is an abuse of notation and is only employed to denote this identification.

We now follow Wahba to present the duality of bayesian estimation problems and the special and general smoothing splines problems. Because we have formulated all our examples within the framework of the general smoothing splines problem, this will automatically render a bayes estimation problem for each one.

5.2 Bayesian Estimation

5.2.1 Special and General Smoothing Splines: The corresponding Bayesian Estimation problems

We now consider the special smoothing splines problem. We have $W_m([0,1]) = H_0 \oplus H_1$, and $K = K_0 + K_1$ as in Section 3. We now want to generate the family of zero-mean Gaussian variables corresponding to H_1. Using results of stochastic calculus and the properties of the Wiener process, it can be derived that:

$$X(t) = \int_0^1 \frac{(t-u)_+^{m-1}}{(m-1)!} dW(u) \tag{5.2}$$

The main result which allows us to conclude this is the family we are looking for is the following: Because of the independent increments property of $W(t)$, for $g_1, g_2 \in L^2([0,1])$:

$$E\left[\int_0^1 g_1(u)dW(u) \int_0^1 g_2(u)dW(u)\right] = \int_0^1 g_1(u)g_2(u)du \tag{5.3}$$

We note that this result can be applied generally, since we are only using the fact that $g(u) = \frac{(t-u)_+^{m-1}}{(m-1)!}$ is the Green function for the corresponding differential operator D^m. Applying this to our process (which in this case is the $m-1$ fold integrated Wiener process, which is the formal solution to $D^m X = dW$) yields the isometric isomorphism between X and H_1.

In [Wah90], Wahba considers two types of Bayes estimates, which lead to the same smoothing spline solution. These estimates can be seen as the solution to Best Linear Predictor or Stochastic Filtering problems as presented in [BTA04]:

Problem 5.1 ***Special Fixed Effects Model:*** *We have a data model of the form*

$$F(t) = \sum_{i=1}^{M} \theta_i \phi_i(t) + b^{1/2} X(t), \quad t \in [0,1] \tag{5.4}$$

$$\{Y_i = F(t_i) + \epsilon_i\}_{i=1}^n \quad ; \quad \epsilon \sim N(0, \sigma^2 I) \tag{5.5}$$

Where θ is a fixed, unknown vector, $b > 0$, and $X(t)$ is our zero-mean Gaussian process with covariance K_1, and uncorrelated with ϵ. We want to estimate $F(t)$ given data $Y_i = y_i$. As in our first example, we want the BLUE estimate of $F(t)$. Again, the condition of being unbiased forces $\widehat{F}(t_i) = \left\langle \widehat{F}(t_i), Y_i \right\rangle = y_i$, and minimizing the MSE is then equivalent to the special problem of smoothing splines, with a penalizing parameter of $\lambda = \frac{\sigma^2}{nb}$.

Problem 5.2 ***Special Random Effects with an Improper Prior:*** *We have the same data model as in the Fixed Effects Model, except that now* $\theta \sim N(0, aI)$ *with* $a > 0$*. Theorem 1.5.3 (in Wahba 1990) guarantees that, if we make* $a \to \infty$ *(impose an improper / minimum informative prior on* θ*), the corresponding estimate* $\widehat{F}_a(t) = E[F(t) \mid \{Y_i = y_i\}]$ *converges pointwise to* $f_\lambda(t)$*.*

The General Fixed and Random Effects Problems have a similar structure, but again provide us with a general framework to work with any smoothing splines problem. We present the Fixed Effects Model for the general case, and then reflect on its application to solve the graph and product space spline problems presented on last section:

Problem 5.3 ***General Fixed Effects Model:*** *Let* $H = H_0 \oplus H_1$ *be a RKHS with kernel* $K = K_0 + K_1$*,* $H_0 = span(\{\phi_i\}_{i=1}^M)$*, and* $\{L_i\}_{i=1}^n$ *a set of bounded linear functionals on* H*. If we have a data model of the form:*

$$F(t) = \sum_{i=1}^{M} \theta_i \phi_i(t) + b^{1/2} X(t), \quad t \in [0,1] \tag{5.6}$$

$$\{Y_i = L_i F(t) + \epsilon_i\}_{i=1}^n \;\; ; \;\; \epsilon \sim N(0, \sigma^2 I) \tag{5.7}$$

Where θ *is a fixed, unknown vector,* $b > 0$*, and* $X(t)$ *is our zero-mean Gaussian process with covariance* K_1*, and uncorrelated with* ϵ*. Given another bounded linear functional* L_0*, we want to estimate* $L_0 F(t)$ *given data* $Y_i = y_i$*. Theorem 1.5.2 in [Wah90] tells us that the BLUE of* $L_0 F$ *given the data is* $L_0 f_\lambda$*, where* f_λ *is the solution to the general smoothing splines problem with* $\lambda = \frac{\sigma^2}{nb}$*.*

The Random Effects Model proceeds in the same way as in the special case. We first note that this general model and its solution allow us to conclude that our smoothing splines solution is not only the best linear estimate given our data model, but we can also safely use it to estimate bounded "observation" linear functionals applied to the true underlying function in our model by evaluating them on our estimate. In the continuous framework, this means we can estimate linear combinations of evaluations, derivatives and certain integral operators. We can then draw confidence intervals and parameter estimation from either of these bayesian models (from minimizing mean square prediction error) and use all the power of generalized linear models.

The application of this model to our proposed smoothing splines problems is now straightforward. We only need to plug in the corresponding spaces, kernels and evaluation functionals. On all the models we have proposed, L_i are just evaluations on a set $\{t_i\}_{i=1}^n$, and therefore our data model is $\{Y_i = F(t_i) + \epsilon_i\}_{i=1}^n$; $\epsilon \sim N(0, \sigma^2 I)$. The only thing that changes in the formulation is the definition of the space H, the corresponding kernels for H_0 and H_1, and therefore $X(t)$. We make some notes on each case:

1. **Regression on a Graph:** Since $T = V \simeq \{1, .., n\}$, $\{X(t)\}_{t \in T}$ can be thought of and handled as the components of a multivariate normal $X \sim N(0, K_1)$. One interesting thing to note is that this allows us to interpret our estimate of either the random effects model (or the norm of f_λ in the $RKHS$) as follows: we set a minimum informative prior on the coefficients of the zero eigenvectors (therefore allowing them to change freely, since they don't change the variance of our estimate) and an independent normal prior on each of the other coefficients, with variance proportional to the corresponding power of $\frac{1}{\lambda_i}$. As in other statistical methods (such as PCA), this renders the lower frequencies the most relevant to our estimate, which is equivalent to the action of the smoothing penalty. It also allows us to discard the highest frequencies based on the low variance of the assigned prior. The use of the Laplacian operator in the smoothing penalty is thus justified in the bayesian estimation framework, since it allows us to use the topological information available to us to construct intelligent priors in an almost automatic and powerful way.

2. **Product of Finite Dimentional Spaces:** Here, T is the product of finite dimentional spaces, like in the graph product example. Because of this, $\{X(t)\}_{t \in T}$ can still be handled as a multivariate normal, although the kernel resulting from penalizing both one way and two way interactions might yield a more complicated interpretation than the one presented in the simple regression on a graph case. It would also be interesting to compare it with a model that parted from the cartesian graph product as a novel graph and used the graph Laplacian on this product.

3. **Product with Infinite Spaces:** In this case, T is the product of V with an infinite dimentional space. We know the existence of $\{X(t)\}_{t \in T}$ is guaranteed, but analizing the properties of this gaussian stochastic process is beyond the scope of this report. However, we know some of its properties and its covariance function K indirectly from the smoothing splines and $RKHS$ framework. In any case, knowing that we are doing linear estimation of the mean of a continuous stochastic process on our product space might yield both intuition and power to build adequate confidence intervals.

6 Estimating Criteria for Smoothing Parameters:

In the Smoothing Splines and Smooth Regression literature, there has been a considerable amount of effort to device criteria to estimate smoothing parameters, and to prove that these estimates provide a good balance between smoothing and fit as well as other desireable properties. This is also true for the graph regression problem: using a complete basis, we have the potential to go

from an exact fit of the data ($c = 0$) to a constant function on each connected component ($c \to \infty$).

All the novel problems we have presented so far are written within the same general smoothing splines framework. We can therefore export the proposed estimates from the groundbreaking work of Wahba, Gu, Nychka and others (See [Wah90],[Iri04],[Gu02],[Nyc88],[GW93] and [GHW79]) for the smoothing parameters for both the simple and the product space cases with minor modifications. These estimates have been shown to have good properties both in theory and in practice. This gives us good motivation to expect a similar performance for our purposes, which we will test in further work.

6.1 Ordinary Cross Validation (OCV)

The idea of Ordinary Cross Validation (OCV from now on) was first presented by Allen (1974) and Wahba & Wold (1975) and is quite simple: we calculate the estimate for our smoothing spline estimate leaving each observation out at a time:

$$f_c^{(k)} = \arg \min_{f \in H_K} \{ \frac{1}{n} \sum_{i \neq k} (y_i - L_i f)^2 + c \left\| P_1 f \right\|^2 \} \tag{6.1}$$

We then wish to pick the smoothing parameter c that minimizes the OCV function $V_0(c)$:

$$c = \arg\min_{c \geq 0} \{V_0(c)\} = \arg\min_{c \geq 0} \{ \frac{1}{n} \sum_{k=1}^{n} (y_k - L_k f_c^{(k)})^2 \} \tag{6.2}$$

That is, the value of $c \geq 0$ such that the mean effect of leaving one out in the fit is minimized. We now present the 'leaving one out lemma', which allows for a direct calculation of $V_0(c)$:

Lemma 6.1 *Let $f_c^{(k)}$ be the solution of (ref). Fix k and let $h_c(k, z)$ be the solution to:*

$$\min_{f \in H_K} \left\{ \frac{1}{n} \left[(z - L_k f)^2 + \sum_{i \neq k} (y_i - L_i f)^2 \right] + c \left\| P_1 f \right\|^2 \right\} \tag{6.3}$$

Then $h_c(k, L_k f_c^{(k)}) = f_c^{(k)}$.

The proof of this lemma is straighforward, because it is only stating $f_c^{(k)}$ is the best $f \in H_K$ that solves the smoothing splines problem without observation k and fits its own value at k. This lemma basically allows us to link the influence matrix $A(c)$ to the formulation of the OCV function. Considering the following identity:

$$y_k - L_k f_c^{(k)} = \frac{y_k - L_k f_c}{1 - a_{kk}^*(c)} \tag{6.4}$$

Where

$$a^*_{kk}(c) = \frac{L_k f_c - L_k f_c^{(k)}}{y_k - L_k f_c^{(k)}} \tag{6.5}$$

If we define $L_k f_c^{(k)} = \widetilde{y}_k$, the leaving one out lemma tells us that $f_c^{(k)}$ is the solution of the general smoothing splines problem, replacing observation y_k with $\widetilde{y}_k$. Therefore, we know that $L_k f_c = (A(c)y)_k$ and $L_k f_c^{(k)} = (A(c)\widetilde{y})_k$, where $\widetilde{y}$ is the modified data vector. Substitution yields:

$$a^*_{kk}(c) = \frac{(A(c)(y_k - \widetilde{y}_k))_k}{y_k - \widetilde{y}_k} = \frac{a_{kk}(c)(y_k - \widetilde{y}_k)}{(y_k - \widetilde{y}_k)} = a_{kk}(c) \tag{6.6}$$

Because our estimates are linear on our data, we can simply identify the diagonal element $a_{kk}(c)$ of the influence matrix to be $\frac{\partial L_k f_c}{y_k}$, that is, the partial derivative of our estimate in the coordinate y_k. Substituting in the expression of the OCV function yields:

$$V_0(c) = \frac{1}{n}\sum_{i=1}^{n} \frac{(y_k - L_k f_c)^2}{(1 - a_{kk}(c))^2} \tag{6.7}$$

Which can be readily calculated and minimized solving the corresponding smoothing splines problems for each value of c. Because the definition of the influence matrix is general, we can adapt this scheme to any of our regression problems.

6.2 Generalized Cross Validation (GCV)

6.2.1 Formulation

The formula for the GCV function can be seen as a generalization or modification of the OCV $V_0(c)$. It results from substituting $a_{kk}(c)$ with the average trace of A, that is, $\frac{1}{n}\sum_{k=1}^{n} a_{kk}(c)$:

$$V(c) = \frac{1}{n}\sum_{i=1}^{n} \frac{(y_k - L_k f_c)^2}{(1 - \frac{1}{n}\sum_{k=1}^{n} a_{kk}(c))^2} = \frac{\|(I - A(c))y\|^2}{\frac{1}{n}Tr(I - A(c))^2} \tag{6.8}$$

The original motivation for this generalization can be found on more detail in Wahba (1990), but it is basically to ensure the estimated c is invariant under similarity transformations. Using the GCV estimate, we can directly calculate the OCV estimate that results from transforming the regression problem to a similar one such that the rows of the design matrix are "maximally coupled", and the resulting influence matrix is circulant (all its diagonal elements are the same, which is the case for periodic smoothing splines problems).

Amongst reasonable estimates for the smoothing parameter, the GCV estimate has shown to have very favourable properties both in theory and in practice. The main result that justifies the use of this estimate in theory is that it is a consistent estimate of the value of c that minimizes predictive mean-square error. Therefore, if the number of observations is sufficiently large, GCV

will probably be one of the best criterions in performance. In [Wah90], Wahba records many of the practical advantages and the limits of GCV She argues that a good and robust performance can be expected of this estimate, except when errors are highly correlated or when we have exact data.

We will now briefly present the theoretical properties of this estimate, and some of the convergence results using it that apply to our graph regression problem. On the next section, we will see that the GCV criterion linked with the bayesian estimation problem also allows the proposition of Bayesian Maximum Posterior Probability intervals, which are highly competitive against other alternatives and also have a global frequentistic interpretation.

6.2.2 GCV as a Predictive Mean-Square Error Criteria

Given a value of c and the corresponding estimate f_c, the predictive mean-square error or PMSE is defined as:

$$T(c) = \frac{1}{n}\sum_{i=1}^{n}(L_i f_c - L_i f)^2 \tag{6.9}$$

Because this measure depends on the unknowns ε and f, it is of course imposible to calculate in real life problems. However, it would be desireable to have an estimate of the c that minimizes PMSE. If we denote $ET(c) = E[T(c)]$ and $g = (L_1 f, .., L_n f)$, from our data model we have that $(L_1 f_c, .., L_n f_c) = A(c)y = A(c)(g + \varepsilon)$ and:

$$\begin{aligned} ET(c) &= \frac{1}{n}E\left\|A(c)(g+\varepsilon) - g\right\|^2 \\ &= \frac{1}{n}\left\|(I - A(c))g\right\|^2 + \frac{1}{n}E[\varepsilon^\top A^2(c)\varepsilon] \\ &= \frac{1}{n}\left\|(I - A(c))g\right\|^2 + \frac{\sigma^2}{n}Tr(A^2(c)) \\ &= b^2(c) + \sigma^2\mu_2(c) \end{aligned} \tag{6.10}$$

Where the terms $b(c)$ and $\mu_2(c)$ are the bias and variance terms, respectively.

We remember from the solution to the general smoothing splines problem that $A(c)y = Td + \Sigma e$. Wahba uses a QR decomposition of $T = (Q_1 : Q_2)(R; 0)$ to rewrite $I - A(c)$ as:

$$I - A(c) = ncQ_2(Q_2^T(\Sigma + ncI)Q_2)^{-1}Q_2^T \tag{6.11}$$

If we let UDU^T be the diagonalization of $Q_2^T\Sigma Q_2$, $\Gamma = Q_2 U$, $h = \Gamma^T g$:

$$I - A(c) = nc\Gamma(D + ncI)^{-1}\Gamma^T \tag{6.12}$$

And substituting in the bias and variance terms yields:

$$b^2(c) = \frac{1}{n}\sum_{i=1}^{n-M}\left(\frac{nch_i}{d_{ii}+nc}\right)^2 \tag{6.13}$$

$$\mu_2(c) = \frac{1}{n}\left[\sum_{i=1}^{n-M}\left(\frac{d_{ii}}{d_{ii}+nc}\right)^2 + M\right] \tag{6.14}$$

This formulation allows us to study the behaviour of $ET(c)$, and to determine that it has at least one minimizer c^* :

1. If $g = T\theta$ ($f \in H_0$, belongs to the span of the zero eigenvectors), then $h = \Gamma g = U^T Q_2^T T\theta = 0$. $ET(c) = \sigma^2\mu_2(c)$, where c is monotone decreasing. This confirms the obvious choice of $c = \infty$ in this case, which corresponds to the least squares regression on H_0, $ET(c) \to \sigma^2\frac{M}{n}$.

2. If $h > 0$, we have that:

$$\frac{db^2}{dc}(c) = 2nc\sum_{i=1}^{n-M}\left(\frac{h_i^2 d_{ii}}{(d_{ii}+nc)^3}\right) \tag{6.15}$$

   Which means b^2 is monotone increasing for $c > 0$, with $\frac{db^2}{dc}(0) = 0$, and $\mu_2(c)$ is monotone decreasing for $c \geq 0$.This, along with the behaviour of these derivatives when $c \to \infty$ allows us to conclude that there is at least one minimizer $c^* > 0$.

The "weak GCV theorem" basically states that there is a sequence $\widehat{c}_n$ of minimizers of $E[V(c)]$ such that the expectation inefficiency:

$$I^* = \frac{ET(\widehat{c})}{ET(c^*)} \tag{6.16}$$

under certain conditions tends to 1 when $n \to \infty$.

6.2.3 Efficient Calculation of GCV and a proposed algorithm for RKHS smoothing splines

After selecting a criterion to select a smoothing parameter, we can proceed to implement an algorithm and solve any problem that can be written in the RKHS general smoothing spline framework. At first glance, minimizing any of these criteria with respect to c requires re-solving the RKHS problem for e and d to get f_c (and thus $A(c)$ and terms related to its trace) at each iteration of the minimization. However, it is possible to precompute the most expensive parts of criteria evaluation, making evaluation within the inner loop of the optimization quite cheap [Wah90, BLWY86].

As we have mentioned earlier, we pick the GCV criteria because of its favorable theoretical properties and its performance in practice. However, we note that the choice of an estimate for the smoothing parameter is often problem-dependant, and we realize a source of further work would be to compare the existing criteria on different graph domain problems. We present a version of the algorithm in [BLWY86] to calculate GCV efficiently. The main idea is to transform the problem into a Ridge Regression [GHW79], and then use the SVD to obtain a formula for $V(c)$. The resulting algorithm can be adapted to all the problems we have posed.

Algorithm 6.2 (RKHS Graph Regression) ***1: Inputs:*** *Graph $G = (V, E)$ w/ adjacency and degree matrices A and D; $n = |V|$; observation functionals $\{L_i\}_{i=1}^n$, observed node indices O; observed values y*
2: Outputs: *Estimated function $f_{\widehat{c}} : V \to \mathbb{R}$; model parameters $\widehat{c}$, $d_{\widehat{c}}$ and $e_{\widehat{c}}$; $V(\widehat{c})$, $\widehat{\sigma^2}$ and Bayesian or Bootstrap Intervals.*
3: $\widetilde{L} = I - D^{-1/2}AD^{-1/2}$ *// Graph Laplacian for G*
4: $[X, \Lambda] = eig(\widetilde{L})$ *// Eigendecomposition of $\widetilde{L}$*
5: $X_0 = \{\phi_i : \lambda_i = 0\}$; $X_1 = \{\phi_i : \lambda_i > 0\}$ $\Lambda_1 = diag\{\lambda_i : \lambda_i > 0\}$
6: $T = LX_0(O, :)$ *//Observ. matrix on null Hilbert space, restr. to Obs. nodes*
7: $K_1 = X_1\Lambda_1^{-m}X_1^\top$; $\Sigma = K_1(O; O)$ *//Kernel / basis for non null Hilbert space restr to Obs. nodes*
8: $[(Q_1\ Q_2), (R_1\ 0)^\top] = qr(T)$ *// Step 1*
9: $B = chol(Q_2^\top \Sigma Q_2)$ *// Step 2*
10: $[U, D, V] = svd(B^\top)$ *// Step 3*
11: $z = U^T Q_2^T y$
12: Define: $V(c) = \dfrac{n\sum_i \left(\frac{nc}{d_{ii}^2+nc}\right)^2 z_i^2}{\left(\sum_i \frac{nc}{d_{ii}^2+nc}\right)^2}$ *// Optimization Objective*
13: $\widehat{c} = Optimize(V, c; n, D, z)$ *// Optimize $V(c)$ given fixed n, D, z*
14: $e_{\widehat{c}} = Q_2U(D^2 + n\widehat{c}I)^{-1}U^TQ_2^Ty$
15: $d_{\widehat{c}} = R_1^{-1}(Q_1^\top \Sigma e_{\widehat{c}})$
16: Return: $f_{\widehat{c}} = X_0 d_{\widehat{c}} + K_1 e_{\widehat{c}}$ *// Function Estimate over the whole graph*

1. **Initial Formulation:** We part from the alternative formulation of the regression problem using the RKHS framework. That is:

$$\widehat{f} = \arg\min_f \left\{ \frac{1}{n} \|y - Tb - \Sigma e\|^2 + ce^\top \Sigma e \right\} \tag{6.17}$$

Where T is of full column rank, and $T^\top e = 0$. In our graph problem, $T = Z_0$ corresponds to the observed coordinates of the zero eigenvectors, and $\Sigma = K_1$ the kernel of H_1.

2. **QR decomposition of T:** We saw on the former section that calculating this decomposition yields a simpler formula for $(I - A(c))$. We compute:

$$T = QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \tag{6.18}$$

Because we want e such that $T^\top e = 0$, we write $e = Q_2\zeta$, $w_1 = Q_1^\top y$ and $w_2 = Q_2^\top y$. We then rewrite the objective function of our minimization problem as:

$$\begin{aligned} & \frac{1}{n}\|y - Tb - \Sigma e\|^2 + ce^\top \Sigma e \\ = \; & \frac{1}{n}\left\|\begin{pmatrix} Q_1 & Q_2 \end{pmatrix}^\top (y - Tb - \Sigma e)\right\|^2 + c\zeta^\top Q_2^\top \Sigma Q_2 \zeta \\ = \; & \frac{1}{n}\left\|w_1 - R_1 b - Q_1^\top \Sigma Q_2 \zeta\right\|^2 + \frac{1}{n}\left\|w_2 - Q_2^\top \Sigma Q_2 \zeta\right\|^2 + c\zeta^\top Q_2^\top \Sigma Q_2 \zeta \end{aligned} \tag{6.19}$$

R_1 is non-singular, since T is of full column rank. Therefore, we can make the last term in this last expression vanish by making:

$$b_c = R_1^{-1}(Q_1^\top \Sigma Q_2 \zeta_c) \tag{6.20}$$

3. **Cholesky decomposition of $Q_2^\top \Sigma Q_2$:** Because the matrix $Q_2^\top \Sigma Q_2$ is SPD, we can form the Cholesky decomposition $Q_2^\top \Sigma Q_2 = L^T L$, where L is upper triangular. We then write $\gamma = L\zeta$, and our objective function becomes:

$$\begin{aligned} & \frac{1}{n}\left\|w_2 - Q_2^\top \Sigma Q_2 \zeta\right\|^2 + c\zeta^\top Q_2^\top \Sigma Q_2 \zeta \\ = \; & \frac{1}{n}\left\|w_2 - L^T L\zeta\right\|^2 + c\zeta^\top L^\top L\zeta \\ = \; & \frac{1}{n}\left\|w_2 - L^T\gamma\right\|^2 + c\gamma^\top\gamma \end{aligned} \tag{6.21}$$

Which is exactly a ridge regression problem on γ. We know the closed-form solution for this problem is:

$$\gamma_c = (L^T L + ncI)^{-1} L w_2 \tag{6.22}$$

4. **SVD of the factor $\mathbf{L}^T$:** Parting from this expression, as in [GHW79], we compute the SVD decomposition of the factor L^T to come up with an efficient way to evaluate $V(c)$. By substitution on our former expressions, we conclude that:

$$\gamma_c = V(D^2 + ncI)^{-1} D U^T Q_2^T y \tag{6.23}$$

Therefore,

$$\begin{aligned} A(c) &= Q_1 Q_1^T + Q_2 U D^2 (D^2 + ncI)^{-1} U^T Q_2^T \\ &= Q \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & D^2(D^2 + ncI)^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & U^T \end{pmatrix} Q^T \end{aligned} \tag{6.24}$$

Finally,

$$I - A(c) = Q \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & I - D^2(D^2 + ncI)^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & U^T \end{pmatrix} Q^T \tag{6.25}$$

And so, if we write $z = U^T Q_2^T y$,

$$V(c) = \frac{n \sum_i \left(\frac{nc}{d_{ii}^2 + nc} \right)^2 z_i^2}{\left(\sum_i \frac{nc}{d_{ii}^2 + nc} \right)^2} \tag{6.26}$$

5. **Solution of the regression problem:** Once we have found the optimum c, this also gives us an efficient way to calculate the solution parameters $(e_{\widehat{c}}, b_{\widehat{c}})$:

$$\begin{aligned} e_{\widehat{c}} &= Q_2 U (D^2 + n\widehat{c}I)^{-1} U^T Q_2^T y \\ b_{\widehat{c}} &= R_1^{-1} (Q_1^\top \Sigma d_{\widehat{c}}) \end{aligned} \tag{6.27}$$

6.3 Unbiased Predictive Risk Estimate

When σ^2 is known, or we have a fairly good estimate of it, we can formulate an unbiased risk estimate for c. This was first suggested by Mallows (1973) and applied to regression by Craven & Wahba ([CW79]). It basically issues an estimate for the expectation of the mean-square prediction error $E[T(c)] = \frac{1}{n} \|(I - A(c))g\|^2 + \frac{\sigma^2}{n} Tr(A^2(c))$ and picks $\widehat{c}$ as its minimizer:

$$\widehat{T}(c) = \frac{1}{n} \|(I - A(c))y\|^2 - \frac{\sigma^2}{n} Tr(I - A(c))^2 + \frac{\sigma^2}{n} Tr(A^2(c)) \tag{6.28}$$

Craven and Wahba show that this estimate behaves exactly like GCV when the same σ^2 is used to generate the data.

This then requires us to estimate σ^2 : As it is common practice in Generalized Linear Models, we take $Tr(A(c))$ as the degrees of freedom for the signal, and given $\widehat{c}$ the $UPRE$ minimizer:

$$\widehat{\sigma^2} = \frac{\|(I - A(\widehat{c}))y\|^2}{Tr(I - A(\widehat{c}))} \tag{6.29}$$

6.4 Generalized Maximum Likelihood (GML) estimate

Using the random effects bayesian estimation model, a generalized maximum likelihood estimate for c can be obtained. This estimate was first obtained by Wahba in 1985. It turns out to be the minimizer of:

$$M(c) = \frac{y^\top (I - A(c))y}{\det^\dagger (I - A(c))^{1/(n-M)}} \tag{6.30}$$

where $\det^\dagger$ indicates the product of the non-zero eigenvalues. There are strong arguments to conclude that this estimate is asymtotically smaller than the optimum estimate, and that it is unadvisable to use if there are considerable deviations from the Gaussian stochastic model that generates it.

6.5 Tensor Product Solution and Parameter Estimation

The advantage of working with a general RKHS formulation is that the modifications we have to make in order to extend this to the product space cases are minor. The main difference lies in the fact that the matrix K_1 now depends on Θ, and we must now optimize $V(c, \Theta)$. The modified algorithm is:

1. Perform steps 1 through 4 given the according initial formulation. We choose initial values for $\Theta = \{\theta_{H_i}\} : \theta_{H_i 0} = Tr(K_{H_i})$.
2. Run a few steps of an optimization or search algorithm that evaluates $V(c \,|\Theta_0)$ using n, c, D and z which returns an initial value c_0.

While stopping criterion for optimization methods for c and Θ are not met:

3. Run one or a few BFGS steps for $V(P|c)$, where $P = \{\log(\theta_{H_i}^{-1})\}$. Save the corresponding approximation of the Hessian B.
4. Recalculate steps 3 and 4 for the new values of Θ
5. Run one Newton step for $\log(c)$

Using $(\widehat{c}, \widehat{\Theta})$, we calculate the solution of the smoothing splines problem.

7 Experiments and Applications

We evaluate the performance of our algorithms for regression over graphs and over $V \times [0, T]$ with the help of synthetic data. First, we present our results on more "familiar" graphs, such as lines or grids. We take advantage of what we know of these domains to test our algorithms, by adding noise to our observations and using a small fraction of our eigenvectors. Then, we present results on randomly generated small world graphs. Finally, we present two examples of the product space algorithm: one over a toy graph and the second over a larger randomly generated small world graph. For each example, results of 10 experiments are averaged.

Familiar Graphs: We choose a **line graph** that corresponds to a uniform discretization of an interval with 500 nodes. We evaluate the results approximating two functions: $f_1(x_i) = \sin(20\pi x_i)$ in $[0, 1]$, and $f_2(x_i) = \frac{\sin(x_i)}{x_i}$ in $[-20, 20]$. We choose $m = 2$ as a good balance between smoothness and fit, and remove observations uniformly (100 nodes are not observed). Finally, we add noise $\varepsilon \sim N(0, \sigma^2)$ on each node. We then choose a **grid graph**, a uniform grid of 30×30 nodes in $[-5, 5]^2$, and evaluate the function $f_3(x, y) = \frac{\sin(x^2+y^2)}{x^2+y^2}$.

We choose the same value of m, and we take out 180 observations uniformly. We again add gaussian noise on each node. We show the results, varying q, the number of eigenvectors:

Table 1. Mean Square Error, $\widehat{c}$ and GCV score for the line graphs and the grid graph

f_1	$\sigma^2 =$	0.1		f_2	$\sigma^2 =$	0.01		f_3	$\sigma^2 =$	0.01	
q	MSE	$\widehat{c}$	$V(\widehat{c})$	MSE	$\widehat{c}$	$V(\widehat{c})$		q	MSE	$\widehat{c}$	$V(\widehat{c})$
500	0.0918	1.1839	0.1172	0.0095	16.1899	0.0109		900	0.0081	0.0042	0.0155
100	0.0928	1.2114	0.1173	0.0095	16.2423	0.0109		300	0.0106	0.0061	0.0157
30	0.1019	0.1265	0.1184	0.0097	16.8878	0.0109		100	0.0146	0.0103	0.0172

Our method reconstructs the signal when we add a reasonable amount of noise, and using less eigenvectors for our function representation increases MSE and the GCV scores only slightly, until we reach a number below which this representation is no longer possible. This confirms our hypothesis: our method provides a sparse representation of the function, even in the harder cases (2 and 3). In Figure 1., we can observe the behaviour of these quantities for $q = 20$ to $q = 100$:

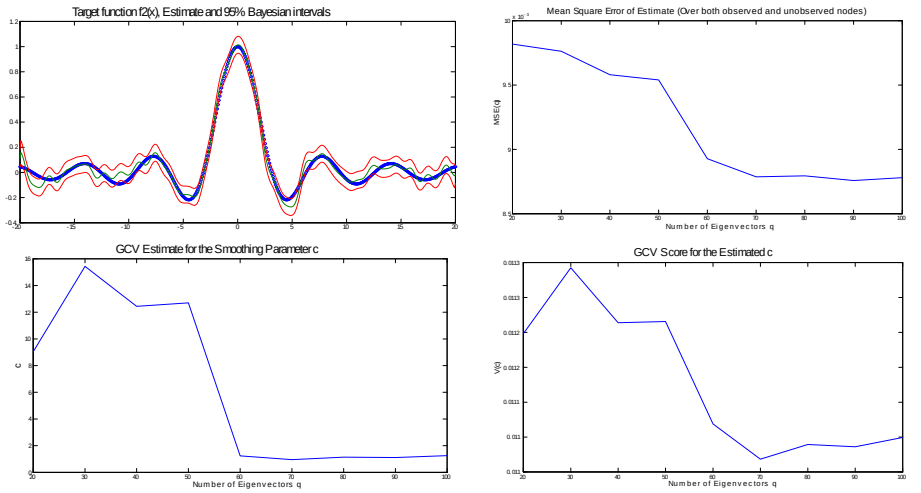

Figure 1. a) Estimate for $f_2(x)$ b),c) and d) MSE,c and V(c) for q=20 to q=100

Small World Graph (Strogratz-Watts): We have tested on several functions over a random small world graph generated by the Watts-Strogratz generator of the Networkx Python 2.4 package. It is constructed by joining each node in a circle with its closest $2b$ neighbors, and rewiring each edge with probability p. We generated a graph of 1000 nodes with $b = 4$ and $p = 0.2$, and constructed three functions based on linear combinations of $4, 100$ and 1000 eigenvectors. We now remove observations by sorting them according to their degree and then taking a uniform sample. Although we expected these graphs to be 'harder' on our methods because of their high local connectivity, we were glad to find that in the first two cases, the method behaves in a similar fashion to the familiar graph cases. As expected, the case with 1000 eigenvectors fails, since this function is equivalent to noise.

Regression on $V \times [0,1]$: We take two graphs, sampling the interval on each case on a uniform mesh of 20 points. We model a function of the same form: $f(t_i, v_j) = \sin(2\pi x_i)\log(1+\phi_1(v_j))$, where ϕ_1 is the first eigenvector of the laplacian of each graph. On the first example, a toy graph of 4 nodes is used (we observe 3). On the second, a small world graph with 100 nodes, $b = 2$, $p = 0.2$ is used, removing 10 observations as in the small world example. We show the results in the toy graph in Figure 2. and results from low and high degree nodes from our second example:

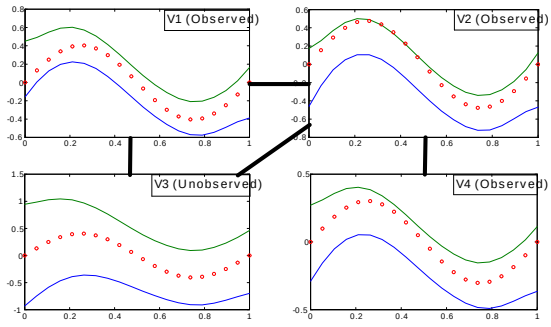

Figure 2. Estimates in our toy graph

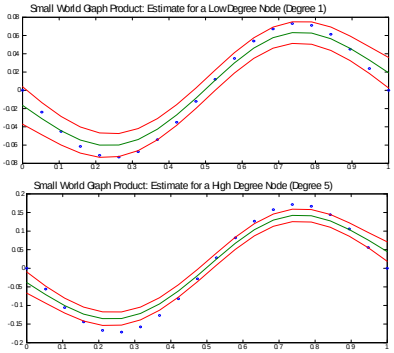

Figure 3. Estimates in Small World Graph

We can conclude that our method provides reasonable estimates for these graphs, even for nodes where our function is not observed at all. By studying different kinds of nodes in the small world graph, we found that the only significant difference is that bayesian intervals tend to be wider on unobserved nodes, especially those of low degree. This follows our intuition: low degree nodes have less 'information' from other nodes to reconstruct the signal.

Acknowledgements

This work was supported by NSF award IIS-0705681. We thank Kiri Wagstaff and the members of the UNM Machine Learning Lab for valuable discussions.

References

[ABB06] S. Agarwal, K. Branson, and S. Belongie. Higher order learning in graphs. *Proceedings of the 23rd International Conference on Machine Learning ICML* pages 17–24 (2006).

[BCMS04] James C. Bremer, Ronald Coifman, Mauro Maggioni, and Arthur Szlam. Diffusion wavelet packets. (2004).

[BLS00] Türker Biyikoglu, Josef Leydold, and Peter Stadler. "Laplacian Eigenvectors of Graphs: Perron-Frobenius and Faber-Krahn Type Theorems". Lecture Notes in Mathematics. Springer Verlag (2000).

[BLWY86] Bates, Lindstrom, Wahba, and Yandell. GCVPACK - routines for generalized cross validation. Technical Report 775, Univeristy of Wisconsin-Madison (1986).

[BTA04] Alain Berlinet and Christine Thomas-Agnan. "Reproducing Kernel Hilbert Spaces in Probability and Statistics". Kluwer Academic Publishers (2004).

[Chu97] Fan R. K. Chung. "Spectral Graph Theory". Conference Board of the Mathematical Sciences. AMS (1997).

[CW79] Peter Craven and Grace Wahba. Smoothing noisy data with spline functions. *Springer Verlag Numerische Mathematik* **31**, 377–403 (1979).

[FJP02] Joel Friedman and Tillich Jean-Pierre. Wave equations for graphs and the edge-based laplacian. (2002).

[FT08] Joel Friedman and Jean-Pierre Tillich. Calculus on graphs. (2008).

[GHW79] Golub, Heath, and G. Wahba. Generalized cross validation as a method for choosing a good ridge parameter. *Technometrics* **21**(2) (May 1979).

[Gu02] Chong Gu. "Smoothing Splines ANOVA Models". Springer Verlag (2002).

[GW93] Chong Gu and Grace Wahba. Smoothing spline ANOVA with component-wise bayesian "confidence intervals". *Journal of Computational and Graphical Statistics* (1993).

[Iri04] Rafael Irizarry. Choosing smoothing parameters for smoothing splines by minimizing an estimate of risk. Technical Report Working Paper 30, Johns Hopkins University, Dept. of Biostatistics (2004).

[Mah05] Sridhar Mahadevan. Representation policy iteration. *UAI* (2005).

[MM07] Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *J. Machine Learning Research* **8**, 2169–2231 (2007).

[Nyc88] Douglas Nychka. Bayesian confidence intervals for smoothing splines. *Journal of the American Statistical Association* **83**(404), 1134 – 1143 (Dec 1988).

[RS05] James Ramsay and B.W. Silverman. "Functional Data Analysis". Springer Series in Statistics. Springer Verlag (2005).

[SBRZ08] Curtis Storlie, H. D. Bondell, B. J. Reich, and H. H. Zhang. The adaptive COSSO for nonparametric surface estimation and model selection. *Annals of Statistics* (2008). in review.

[Wah90] Grace Wahba. "Spline Models for Observational Data". CBMS-NSF Regional Conference Series in Applied Mathematics. SCIAM (1990).

[Wah03] Grace Wahba. An introduction to (smoothing spline) ANOVA models in RKHS, with examples in geographical data, medicine, atmospheric science and machine learning. *Proceedings of the 13th IFAC Symposium on System Identification, Rotterdam* pages 549–559 (2003).

[ZS04] Dengyong Zhou and Berhnard Schölkopf. A regularization framework for learning from graph data. *Workshop on Statistical Relational Learning at ICML, Banff, Canada* (2004).

.