Stationary signal processing on graphs

Nathanaël Perraudin and Pierre Vandergheynst [*]

December 27, 2018

Abstract

Graphs are a central tool in machine learning and information processing as they allow to conveniently capture the structure of complex datasets. In this context, it is of high importance to develop flexible models of signals defined over graphs or networks. In this paper, we generalize the traditional concept of wide sense stationarity to signals defined over the vertices of arbitrary weighted undirected graphs. We show that stationarity is intimately linked to statistical invariance under a localization operator reminiscent of translation. We prove that stationary graph signals are characterized by a well-defined Power Spectral Density that can be efficiently estimated even for large graphs. We leverage this new concept to derive Wiener-type estimation procedures of noisy and partially observed signals and illustrate the performance of this new model for denoising and regression.

Index terms— Stationarity, graphs, spectral graph theory, power spectral density, Wiener filter, covariance, Gaussian random fields

1 Introduction

Stationarity is a traditional hypothesis in signal processing used to represent a special type of statistical relationship between samples of a temporal signal. The most commonly used is wide-sense stationarity, which assumes that the first two statistical moments are invariant under translation, or equivalently that the correlation between two samples depends only on their time difference. Stationarity is a corner stone of many signal analysis methods. The expected frequency content of stationary signals, called Power Spectral Density (PSD), provides an essential source of information used to build signal models, generate realistic surrogate data or perform predictions. In Figure 1, we present an example of a stationary process (blue curve) and two predictions (red and green curves). As the blue signal is a realization of a stationary process, the red curve is more probable than the green one because it respects the frequency content of the observed signal.

Classical stationarity is a statement of statistical regularity under arbitrary translations and thus requires a regular structure (often "time"). However many signals do not live on such a regular structure. For instance, imagine that instead of having one sensor returning a temporal signal, we have multiple sensors living in a two-dimensional space, each of which delivers only one value. In this case (see Figure 2 left), the signal support is no longer regular. Since there exists an underlying continuum in this example (2D space), one could assume the existence of a 2D stationary field and use Kriging [1] to interpolate observations to arbitrary locations, thus generalizing stationarity for a regular domain but irregularly spaced samples.

On the contrary, the goal of this contribution is to generalize stationarity for an irregular domain that is represented by a graph, *without resorting to any underlying regular continuum.* Graphs are convenient for this task as they are able to capture complicated relations between variables. In this work, a graph is composed of vertices connected by weighted undirected edges and signals are now scalar values observed at the vertices of the graph. Our approach is to use a weak notion of translation invariance, define on a graph, that captures the structure (if any) of the data. Whereas classical stationarity means correlations are computed by translating the auto-correlation function, here correlations are given by localizing a common graph kernel, which is a generalized notion of translation as detailed in Section 2.2.

Figure 2 (left) presents an example of random multivariate variable living in a 2-dimensional space. Seen as scattered samples of an underlying 2D stochastic function, one would (rightly) conclude it is not stationary. However, under closer inspection, the observed values look stationary *within* the spiral-like

[*]EPFL, Ecole Polytechnique Fédérale de Lausanne, LTS2 Laboratoire de traitement du signal, CH-1015 Lausanne, Switzerland

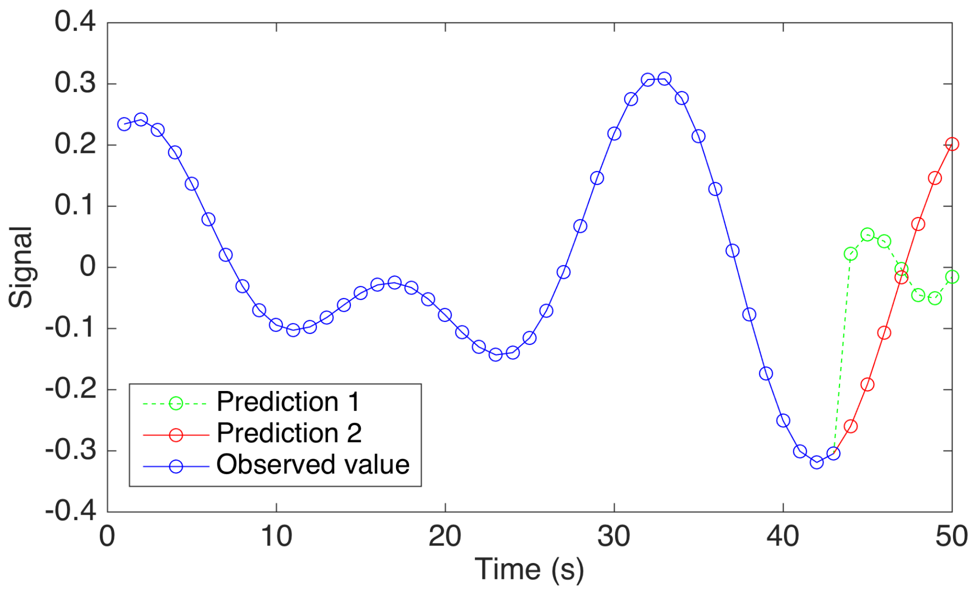


Figure 1: Signal prediction. The red curve is more likely to occur than the green curve because it respects the frequency statistics of the blue curve.

structure depicted by the graph in Figure 2 (right). The traditional Kriging interpolation technique would ignore this underlying structure and conclude that there are always rapid two dimensional variations in the underlying continuum space. This problem does not occur in the graph case, where the statistical relationships inside the data follow the graph edges resulting in this example in signals oscillating smoothly over the graph. A typical example of a stationary signal on a graph would be the result of a survey

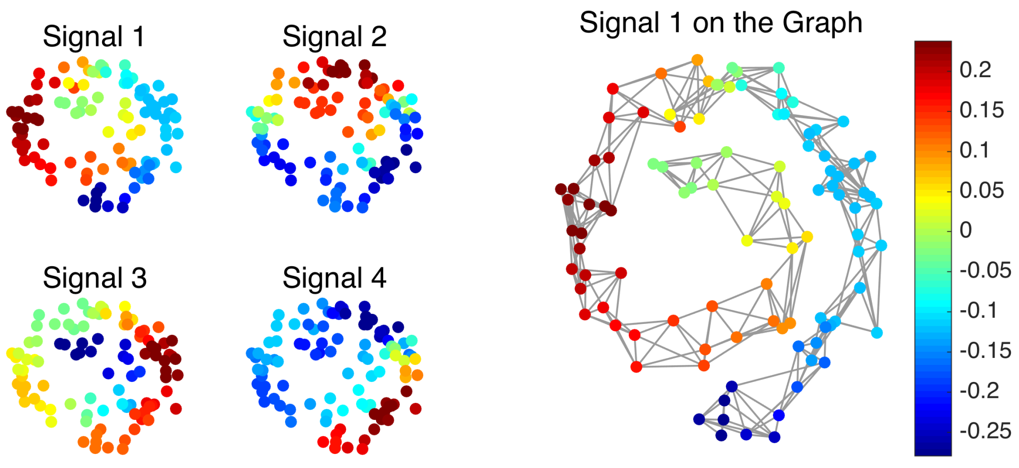


Figure 2: Example of stationary graph signals. The graph connections express relationships between the different elements of one signal. In this case, the signal varies smoothly along the snail shape of the graph.

performed by the users of a social network. If there is a relationship between a user's answers and those of his neighbours, this relationship is expected to be constant among all users. Using stationarity on the graph, we could predict the most probable answer for users that never took the survey.

1.1 Contributions

We use spectral graph theory to extend the notion of stationarity to a broader class of signals. Leveraging the graph localization operator, we establish the theoretical basis of this extension in Section 3. We show that the resulting notion of stationarity is equivalent to the proposition of Girault [2, Definition 16], although the latter is not defined in terms of a localisation operator. Localisation is a very desirable feature, since it naturally expresses the scale at which samples are strongly correlated.

Since our framework depends on the power spectral density (PSD), we generalize the Welch method [3, 4] in Section 4 and obtain a scalable and robust way to estimate the PSD. It improves largely the covariance estimation when the number of signals is limited.

Based on the generalization of Wiener filters, we propose a new regularization term for graph signal optimization instead of the traditional Dirichlet prior, that depends on the noise level and on the PSD of the signal. The new optimization scheme presented in Section 5 has three main advantages: 1) it allows to deal with an arbitrary regularization parameter, 2) it adapts to the data optimally as we prove that the optimization model is a Maximum A Posteriori (MAP) estimator, and 3) it is more scalable and robust than a traditional Gaussian estimator.

Finally, in Section 6, we show experimentally that common datasets such as USPS follow our stationarity assumption. In section 7, we exploit this fact to perform missing data imputation and we show how stationarity improves over classical graph models and Gaussian MAP estimator.

1.2 Related work

Graphs have been used for regularization in data applications for more than a decade [5, 6, 7, 8] and two of the most used models will be presented in Section A. The idea of graph filtering was hinted at by the machine learning community [9] but developed for the spectral graph wavelets proposed by Hammond et al. [10] and extended by Shuman et al. in [11]. While in most cases, graph filtering is based on the graph Laplacian, Moura et al. [12] have suggested to use the adjacency matrix instead.

We note that a probabilistic model using Gaussian random fields has been proposed in [13, 14]. In this model, signals are automatically graph stationary with an imposed covariance matrix. Our model differentiates itself from these contributions because it is based on a much less restrictive hypothesis and uses the point of view of stationarity. A detailed explanation is given at the end of Section 3.

Finally, stationarity on graphs has been recently proposed in [15, 2] by Girault et al. These contributions use a different translation operator, promoting energy preservation over localization. While seemingly different, we show that our approach and Girault's result in the same graph spectral characterisation of stationary signals. Girault et al [16] have also shown that using the Laplacian as a regularizer in a de-noising problem (Tikhonov) is equivalent to applying a Wiener filter adapted to a precise class of graph signals. In [2, pp 100], an expression of graph Wiener filter can be found.

2 Background theory

2.1 Graph signal processing

Graph nomenclature A graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ is defined by two sets: $\mathcal{V}, \mathcal{E}$ and a weight function $\mathcal{W}$. $\mathcal{V}$ is the set of vertices representing the nodes of the graph and $\mathcal{E}$ is the set of edges that connect two nodes if there is a particular relation between them. In this work all graphs are undirected. To obtain a finer structure, this relation can be quantified by a weight function $\mathcal{W} : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ that reflects to what extent two nodes are related to each other. Let us index the nodes from $1, \dots, N = |\mathcal{V}|$ and construct the weight matrix $W \in \mathbb{R}^{N \times N}$ by setting $W[i,j] = \mathcal{W}(v_i, v_j)$ as the weight associated to the edge connecting the node i and the node j. When no edge exists between i and j, the weight is set to 0. For a node $v_i \in \mathcal{V}$, the degree $d[i]$ is defined as $d[i] = \sum_{j=1}^{N} W[i,j]$. In this framework, a signal is defined as a function $f : \mathcal{V} \rightarrow \mathbb{R}$ (or $\mathbb{C}$) assigning a scalar value to each vertex. It is convenient to consider a signal f as a vector of size N with the n^{th} component representing the signal value at the n^{th} vertex.

The most fundamental operator in graph signal-processing is the (combinatorial) graph Laplacian, defined as: $L = D - W$, where D is the diagonal degree matrix ($D[i,i] = d[i]$).

Spectral theory Since the Laplacian L is always a symmetric positive semi-definite matrix, we know from the spectral theorem that it possesses a complete set of orthonormal eigenvectors. We denote them by $\{u_\ell\}_{\ell=0,1,\dots,N-1}$. For convenience, we order the set of real, non-negative eigenvalues as follows:

$0 = \lambda_0 < \lambda_1 \leq \cdots \leq \lambda_{N-1} = \lambda_{\max}$. When the graph is connected[1], there is only one zero eigenvalue. In fact, the multiplicity of the zero eigenvalue is equal to the number of connected components. For more details on spectral graph theory, we refer the reader to [17, 18]. The eigenvectors of the Laplacian are used to define a graph Fourier basis [5, 11] which we denote as U. The eigenvalues are considered as a generalization of squared frequencies. The Laplacian matrix can thus be decomposed as

$$L = U\Lambda U^*,$$

where U^* denotes the transposed conjugate of U. The graph Fourier transform is written $\hat{f} = U^* f$ and its inverse $f = U\hat{f}$. This Graph Fourier Transform possesses interesting properties further studied in [11]. Note that the graph Fourier transform is equivalent to the Discrete Fourier transform for cyclic graphs. The detailed computation for the "ring" can be found in [19, pp 136-137].

Graph convolutive filters The graph Fourier transform plays a central role in graph signal processing since it allows a natural extension of filtering operations. In the classical setting, applying a filter to a signal is equivalent with a convolution, which is simply a point-wise multiplication in the spectral domain. For a graph signal, where the domain is not regular, filtering is still well defined, as a point-wise multiplication in the spectral domain [11, Equation 17]. A graph convolutive filter $g(L)$ is defined from a continuous kernel $g : \mathbb{R}_+ \to \mathbb{R}$. In the spectral domain, filtering a signal s with a convolutive filter $g(L)$ is, as the classical case, a point-wise multiplication written as $\hat{s'}[\ell] = g(\lambda_\ell) \cdot \hat{s}[\ell]$, where $\hat{s'}, \hat{s}$ are the Fourier transform of the signals s', s. In the vertex domain, we have

$$s' := g(L)s = Ug(\Lambda)U^* s, \tag{1}$$

where $g(\Lambda)$ is a diagonal matrix with entries $g(\lambda_\ell)$. For convenience, we abusively call 'filter' the generative kernel g. We also drop the term convolutive as we are only going to use this type of filters. A comprehensive definition and study of these operations can be found in [11]. It is worth noting that these formulas make explicit use of the Laplacian eigenvectors and thus its diagonalization. The complexity of this operation is in general $\mathcal{O}(N^3)$. In order to avoid this cost, there exist fast filtering algorithms based on Chebyshev polynomials or the Lanczos method [10, 20]. These methods scale with the number of edges $|E|$ and reduce the complexity to $\mathcal{O}(|E|)$, which is advantageous in the case of sparse graphs.

2.2 Localization operator

As most graphs do not possess a regular structure, it is not possible to translate a signal around the vertex set with an intuitive shift. As stationarity is an invariance with respect to translation, we need to address this issue first. A solution is present in [11, Equation 26], where Shuman et. al. define the generalized translation for graphs as a convolution with a Kroneker delta. The convolution $*$ is defined as the element-wise multiplication in the spectral domain leading to the following generalized translation definition:

$$T_i s[n] := (s * \delta_i)[n] = \sum_{\ell=0}^{N-1} \hat{s}[\ell] u_\ell^*[i] u_\ell[n].$$

Naturally, the generalized translation operator does not perform what we would intuitively expect from it, i.e it does not shift a signal s from node n to node i as this graph may not be shift-invariant. Instead when $\hat{s}$ changes smoothly across the frequencies (more details later on), then $T_i s$ is localized around node i, while s is in general not localized at a particular node or set of nodes.

In order to avoid this issue, we define the localization operator as follow

Definition 1. *Let $\mathcal{C}$ be the set of functions $\mathbb{R}^+ \to \mathbb{R}$. For agraph kernel $g \in \mathcal{C}$ (defined in the spectral domain) and a node i, the localization operator $\mathcal{T}_i : \mathcal{C} \to \mathbb{R}^N$ reads:*

$$\mathcal{T}_i g[n] := \sum_{\ell=0}^{N-1} g(\lambda_\ell) u_\ell^*[i] u_\ell[n] = (g(L)\delta_i)[n] = g(L)[i, n]. \tag{2}$$

Here we use the calligraphic notation $\mathcal{T}_i$ to differentiate with the generalized translation operator T_i. We first observe from (2) that the i^{th} line of graph filter matrix $g(L)$ is the kernel g localized at node i. Intuitively, it signifies $[g(L)s](i) = \langle s, \mathcal{T}_i g\rangle$. We could replace $g(\lambda_\ell)$ by $\hat{s}[\ell]$ in Definition 1 and localize

[1]a path connects each pair of nodes in the graph

the discrete vector $\hat{s}$ instead. We prefer to work with a kernel for two reasons. 1) In practice when the graph is large, the Fourier basis cannot be computed making it impossible to localize a vector. On the other side, for a kernel g, there are techniques to approximate $\mathcal{T}_i g$. 2) The localization properties are theoretically easier to interpret when g is a filter. Let us suppose that g is a K order polynomial, then the support of $\mathcal{T}_i g$ is exactly contained in a ball of radius K centered at node i. Building on this idea, for a sufficiently regular function g, it has been proved in [11, Theorem 1 and Corollary 2] that the localization operator concentrates the kernel g around the vertex i.

Let us now clarify how generalized translation and localization are linked. The main difference between these two operators is the domain on which they are applied. Whereas, the translation operator acts on a discrete signal defined in the time or the vertex domain, the localization operator requires a continuous kernel or alternatively a discrete signal in the spectral domain. Both return a signal in the time or the vertex domain. To summarize, the localization operator can be seen as computing the inverse Fourier transform first and then translating the signal. It is an operator that takes a filter from the spectral domain and localizes it at a given node i while adapting it to the graph structure.

In the classical periodic case ("ring" graph), the translation and localization operators coincide, i.e:

$$\begin{aligned} \mathcal{T}_i g[n] &= \frac{1}{N}\sum_{\ell=1}^{N} g(\lambda_\ell) e^{-j2\pi\frac{\ell i}{N}} e^{j2\pi\frac{\ell n}{N}} \\ &= \frac{1}{N}\sum_{\ell=1}^{N} g(\lambda_\ell) e^{j2\pi\frac{\ell(n-i)}{N}} = \mathcal{T}_0 g[n-i]. \end{aligned} \tag{3}$$

In this case, localizing a kernel g can be done by computing the inverse discrete Fourier transform of the vector $\hat{s}(\ell) = g(\lambda_\ell)$ and then translating at node i. However, for irregular graphs, localization differs from translation because the shape of the localized filter adapts to the graph and varies as a function of its topology. Figure 3 shows an example of localization using the Mexican hat wavelet filter. The shape of the localized filter depends highly on the graph topology. However, we observe that the general shape of the wavelet is preserved. It has large positive values around the node where it is localized. It then goes negative a few nodes further away and stabilizes at zero for nodes far away. To summarize, the localization operator preserves the global behavior of the filter while adapting to the the graph topology. Additional insights about the localization operator can be found in [11, 10, 21, 22].

2.3 Stationarity for temporal signals

Let $\mathbf{x}[t]$ be a time indexed stochastic process. Throughout this paper, all random variables are written in bold fonts. We use $m_\mathbf{x} = \mathbb{E}\{\mathbf{x}\}$ to denote the expected value of $\mathbf{x}$. In this section, we work with the periodic discrete case.

Definition 2 (Time Wide-Sense Stationarity). *A signal is Time Wide-Sense Stationary (WSS) if its first two statistical moments are invariant under translation, i.e:*

1. $m_\mathbf{x}[t] = \mathbb{E}\{\mathbf{x}[t]\} = c \in \mathbb{R}$,
2. $\mathbb{E}\{(\mathbf{x}[t] - m_\mathbf{x})(\mathbf{x}[s] - m_\mathbf{x})^*\} = \eta_\mathbf{x}[t-s]$,

where $\eta_\mathbf{x}$ *is called the autocorrelation function of* $\mathbf{x}$.

Note that using (3), the autocorrelation can be written in terms of the localization operator:

$$\eta_\mathbf{x}[t-s] = \mathcal{T}_s \gamma_\mathbf{x}[t]. \tag{4}$$

For a WSS signal, the autocorrelation function depends only on one parameter, $t-s$, and is linked to the Power Spectral Density (PSD) through the Wiener-Khintchine Theorem [23]. The latter states that the PSD of the stochastic process $\mathbf{x}$ denoted $\gamma_\mathbf{x}[\ell]$ is the Fourier transform of its auto-correlation :

$$\gamma_\mathbf{x}[\ell] = \frac{1}{\sqrt{N}}\sum_{i=1}^{N} \eta_\mathbf{x}[t] e^{-j2\pi\frac{\ell t}{N}}, \tag{5}$$

where $j = \sqrt{-1}$. As a consequence, when a signal is convolved with a filter $\check{h}$, its PSD is multiplied by the energy of the convolution kernel: for $\mathbf{y} = \check{h} * \mathbf{x}$, we have

$$\gamma_\mathbf{y}[\ell] = |h[\ell]|^2 \gamma_\mathbf{x}[\ell],$$

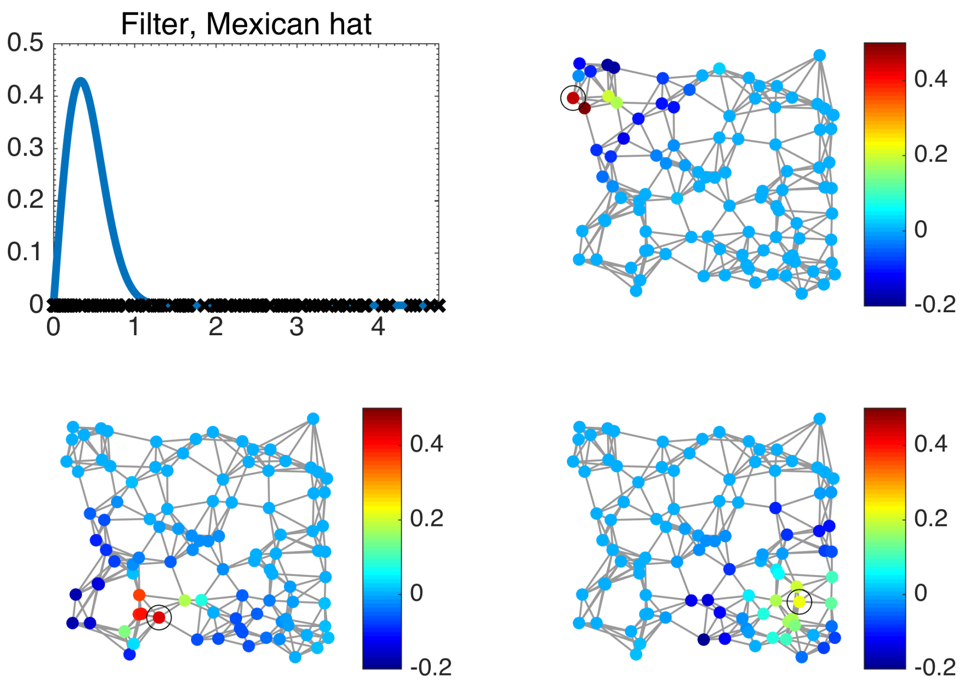


Figure 3: Top left: Mexican hat filter in the spectral domain $g(x) = \frac{5x}{\lambda_{\max}} \exp\left(-\frac{25x^2}{\lambda_{\max}^2}\right)$. The filter is localized around three different vertices (highlighted by a black circle).

where $h[\ell]$ is the Fourier transform of $\check{h}$. For more information about stationarity, we refer the reader to [24].

When generalizing these concepts to graphs, the underlying structure for stationarity will no longer be time, but graph vertices.

3 Stationarity of graph signals

We now generalize stationarity to graph signals. While we define stationarity through the localization operator, Girault [15] uses an isometric translation operator instead. That proposition is briefly described in Section 3.2, where we also show the equivallence of both definitions.

3.1 Stationarity under the localisation operator

Let $\mathbf{x} \in \mathbb{R}^N$ be a stochastic graph signal with a finite number of variables indexed by the vertices of a weighted undirected graph. The expected value of each variable is written $m_{\mathbf{x}}[i] = \mathbb{E}\{\mathbf{x}[i]\}$ and the covariance matrix of the stochastic signal is $\Sigma_{\mathbf{x}} = \mathbb{E}\{(\mathbf{x} - m_{\mathbf{x}})(\mathbf{x} - m_{\mathbf{x}})^*\}$. We additionally define $\tilde{\mathbf{x}} = \mathbf{x} - m_{\mathbf{x}}$. For discrete time WSS processes, the covariance matrix $\Sigma_{\mathbf{x}}$ is Toeplitz, or circulant for periodic boundary conditions, reflecting translation invariance. In that case, the covariance is diagonalized by the Fourier transform. We now generalize this property to take into account the intricate graph structure.

As explained in Section 2.2, the localization operator adapts a kernel to the graph structure. As a result, our idea is to use the localization operator to adapt the correlation between the samples to the graph structure. This results in a localised version of the correlation function, whose properties can then be studied via the associated kernel.

Definition 3. *A stochastic graph signal* $\mathbf{x}$ *defined on the vertices of a graph* $\mathcal{G}$ *is called Graph Wide-Sense (or second order) Stationary (GWSS), if and only if it satisfies the following properties:*

1. *its first moment is constant over the vertex set:* $m_{\mathbf{x}}[i] = \mathbb{E}\{\mathbf{x}[i]\} = c \in \mathbb{R}$ *and*

2. its covariance is invariant with respect to the localization operator:

$$\Sigma_{\mathbf{x}}[i,n] = \mathbb{E}\big\{(\mathbf{x}[i]-m_{\mathbf{x}})(\mathbf{x}[n]-m_{\mathbf{x}})\big\} = \mathcal{T}_i\gamma_{\mathbf{x}}[n].$$

The first part of the above definition is equivalent to the first property of time WSS signals. The requirement for the second moment is a natural generalization where we are imposing an invariance with respect to the localization operator instead of the translation. It is a generalization of Definiton 2 using (4). In simple words, the covariance is assumed to be driven by a global kernel (filter) $\gamma_{\mathbf{x}}$. The localization operator adapts this kernel to the local structure of the graph and provides the correlation between the vertices. Additionally, Definition 3 implies that the spectral components of $\mathbf{x}$ are uncorrelated.

Theorem 1. *If a signal is GWSS, its covariance matrix $\Sigma_{\mathbf{x}}[i,j]$is jointly diagonalizable with the Laplacian of $\mathcal{G}$, i.e $\Sigma_{\mathbf{x}} = U\Gamma_{\mathbf{x}}U^*$, where $\Gamma_{\mathbf{x}}$ is a diagonal matrix.*

Proof. By Definition 1, the covariance localization operator can be written as:

$$\mathcal{T}_i\gamma_{\mathbf{x}}[n] = \gamma_{\mathbf{x}}(L)[i,n] = U\gamma_{\mathbf{x}}(\Lambda)U^*[i,n] \tag{6}$$

where $\gamma_{\mathbf{x}}(\Lambda)$ is a diagonal matrix satisfying $\gamma_{\mathbf{x}}(\Lambda)[\ell,\ell] = \gamma_{\mathbf{x}}(\lambda_\ell)$. To complete the proof set $\Gamma_{\mathbf{x}} = \gamma_{\mathbf{x}}(\Lambda)$. □

The choice of the filter $\gamma_{\mathbf{x}}$ in this result is somewhat arbitrary, but we shall soon see that we are interested in localized kernels. In that case, $\gamma_{\mathbf{x}}$ will be typically be the lowest degree polynomial satisfying the constraints and can be constructed using Lagrange interpolation for instance.

Definition 3 provides a fundamental property of the covariance. The size of the correlation (distance over the graph) depends on the support of localized the kernel $\mathcal{T}_i\gamma_{\mathbf{x}}$. In [11, Theorem 1 and Corollary 2], it has been proved that the concentration of $\mathcal{T}_i\gamma_{\mathbf{x}}$ around i depends on the regularity[2] of $\gamma_{\mathbf{x}}$. For example, if $\gamma_{\mathbf{x}}$ is polynomial of degree K, it is exactly localized in a ball of radius K. Hence we will be mostly interested in such low degree polynomial kernels.

The graph spectral covariance matrix of a stochastic graph signal is given by $\Gamma_{\mathbf{x}} = U^*\Sigma_{\mathbf{x}}U$. For a GWSS signal this matrix is diagonal and the graph power spectral density (PSD) of $\mathbf{x}$ becomes:

$$\gamma_{\mathbf{x}}(\lambda_\ell) = (U^*\Sigma_{\mathbf{x}}U)_{\ell,\ell}\,. \tag{7}$$

Table 1 presents the differences and the similarities between the classical and the graph case. For a regular cyclic graph (ring), the localization operator is equivalent to the traditional translation and we recover the classical cyclic-stationarity results by setting $\eta_{\mathbf{x}} = \mathcal{T}_0\gamma_{\mathbf{x}}$. Our framework is thus a generalization of stationarity to irregular domains.

Example 1 (Gaussian i.i.d. noise)**.** *Normalized Gaussian i.i.d. noise is GWSS for any graph. Indeed, the first moment is $\mathbb{E}\big(\mathbf{x}[k]\big) = 0$. Moreover, the covariance matrix can be written as $I = \Sigma_{\mathbf{x}} = UIU^*$ with any orthonormal matrix U and thus is diagonalizable with any graph Laplacian. We also observe that the PSD is constant, which implies that similar to the classical case, white noise contains all "graph frequencies".*

When $\gamma_{\mathbf{x}}$ is a bijective function, the covariance matrix contains an important part of the graph structure: the Laplacian eigenvectors. On the contrary, if $\gamma_{\mathbf{x}}$ is not bijective, some of the graph structure is lost as it is not possible to recover all eigenvectors. This is for instance the case when the covariance matrix is low-rank. As another example, let us consider completely uncorrelated centered samples with variance 1. In this case, the covariance matrix becomes $\Sigma_{\mathbf{x}} = I$ and loses all graph information, even if by definition the stochastic signal remains stationary on the graph.

One of the crucial benefits of stationarity is that it is preserved by filtering, while the PSD is simply reshaped by the filter. The same property holds on graphs.

Theorem 2. *When a graph filter g is applied to a GWSS signal, the result remains GWSS, the mean becomes $m_{g(L)\mathbf{x}} = m_{\mathbf{x}}g(0)$ and the PSD satisfies:*

$$\gamma_{g(L)\mathbf{x}}(\lambda_\ell) = |g(\lambda_\ell)|^2 \cdot \gamma_{\mathbf{x}}(\lambda_\ell). \tag{8}$$

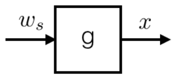
[2]A regular kernel can be well approximated by a smooth function, for instance a low order polynomial, over spectrum of the laplacian.

Proof. The output of a filter g can be written as $\mathbf{x}' = g(L)\tilde{\mathbf{x}} + g(L)m_{\mathbf{x}}$. If the input signal $\mathbf{x}$ is GWSS, we can check easily that the first moment of the filter's output is constant, $\mathbb{E}(g(L)\mathbf{x}[i]) = g(L)\mathbb{E}(m_{\mathbf{x}}) = g(0)m_{\mathbf{x}}$. The computation of the second moment gives:

$$\begin{aligned}\mathbb{E}\left(g(L)\tilde{\mathbf{x}}\big(g(L)\tilde{\mathbf{x}}\big)^*\right) &= g(L)\mathbb{E}\left(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^*\right)g(L)^* \\ &= g(L)\Sigma_{\mathbf{x}}g(L)^* \\ &= Ug^2(\Lambda)\gamma_{\mathbf{x}}(\Lambda)U^*,\end{aligned}$$

which is equivalent to our claim. □

Theorem 2 provides a simple way to artificially produce stationary signals with a prescribed PSD by simply filtering white noise :



The resulting signal will be stationary with PSD g^2. In the sequel, we assume for simplicity that the signal is centered at 0, i.e: $m_{\mathbf{x}} = 0$. Note that the input white noise could well be non-Gaussian.

	Classical	Graph
Stationary with respect to	Translation	The localization operator
First moment	$\mathbb{E}\big(\mathbf{x}[i]\big) = m_{\mathbf{x}} = c \in \mathbb{R}$	$\mathbb{E}\big(\mathbf{x}[i]\big) = m_{\mathbf{x}} = c \in \mathbb{R}$
Second moment	$\Sigma_{\mathbf{x}}[i,n] = \mathbb{E}\big(\tilde{\mathbf{x}}[i])\tilde{\mathbf{x}}^*[n]\big) = \eta_{\mathbf{x}}[t-s]$	$\Sigma_{\mathbf{x}}[i,n] = \mathbb{E}\big(\tilde{\mathbf{x}}[i])\tilde{\mathbf{x}}^*[n]\big) = \gamma_{\mathbf{x}}(L)_{i,n}$
(We use $\tilde{\mathbf{x}} = \mathbf{x} - m_{\mathbf{x}}$)	$\Sigma_{\mathbf{x}}$ Toeplitz	$\Sigma_{\mathbf{x}}$ diagonalizable with L
Wiener Khintchine	$\gamma_{\mathbf{x}}(\lambda_\ell) = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}\eta_{\mathbf{x}}[n]e^{-j2\pi\frac{n\ell}{N}}$	$\gamma_{\mathbf{x}}(\lambda_\ell) = (\Gamma_{\mathbf{x}})_{\ell,\ell} = (U^*\Sigma_{\mathbf{x}}U)_{\ell,\ell}$
Result of filtering	$\gamma_{\check{g}*\mathbf{x}}(\lambda_\ell) = \lvert g(\lambda_\ell)\rvert^2 \cdot \gamma_{\mathbf{x}}(\lambda_\ell)$	$\gamma_{g(L)\mathbf{x}}[\ell] = \lvert g(\lambda_\ell)\rvert^2 \cdot \gamma_{\mathbf{x}}(\lambda_\ell)$

Table 1: Comparison between classical and graph stationarity. In the classical case, we work with a N periodic discrete signal.

3.2 Comparison with the work of B. Girault

Stationarity for graph signals has been defined in the past [2, 15]. The proposed definition is based on an isometric graph translation operator defined for a graph signal s as:

$$T_B s := e^{i\frac{2\pi}{\lambda_{\max}}L}s = b(L)s,$$

where $b(x) = e^{i\frac{2\pi}{\lambda_{\max}}x}$. While this operator conserves the energy of the signal ($\|T_B s\|_2 = \|s\|_2$), it does not have localization properties. In a sense, one trades localisation for isometry. Using this operator, the stationarity definition of Girault [15] is a natural extension of classical case.

Definition 4. [2, Definition 16] *A stochastic signal* $\mathbf{x}$ *on the graph* $\mathcal{G}$ *is Wide-Sense Stationary (WSS) if and only if*

1. $\mathbb{E}\left(T_B\mathbf{x}\right) = \mathbb{E}\left(\mathbf{x}\right)$
2. $\mathbb{E}\left(T_B\mathbf{x}\left(T_B\mathbf{x}\right)^*\right) = \mathbb{E}\left(\mathbf{x}\mathbf{x}^*\right)$

While this definition is based on a different generalization of the translation operator, the resulting notion of stationarity is equivalent. Indeed, [2, Theorem 7] tells that if a signal is stationary with Definition 4, then its first moment is constant and the covariance matrix in the spectral domain $U^*\Sigma_{\mathbf{x}}U$ has to be diagonal. Using Theorem 1 we therefore recover Definition 3.

To summarize, though they are based on different "translation" operators, both definitions are equivalent. Interestingly this shows that, for stationary signals, the non-isometric nature of the localisation operator does not really matter. Since localisation is a very desirable feature, allowing for instance to study the scale of correlations over the vertex set, we believe that assuming invariance with respect to the localisation operator is a more natural way to define graph stationarity.

3.3 Gaussian random field interpretation

The framework of stationary signals on graphs can be interpreted using Gaussian Marcov Random Field (GMRF). Let us assume that the signal $\mathbf{x}$ is drawn from a distribution

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z_p} e^{-(\mathbf{x}-m_\mathbf{x})^* p(L)(\mathbf{x}-m_\mathbf{x})}, \tag{9}$$

where $Z_p = \int_{\mathbb{R}^N} e^{-(\mathbf{x}-m_\mathbf{x})^* p(L)(\mathbf{x}-m_\mathbf{x})} \mathrm{d}\mathbf{x}$. If we assume that $p(L)$ is invertible, drawing from this distribution will generate a stationary $\mathbf{x}$ with covariance matrix given by:

$$\Sigma_\mathbf{x} = (p(L))^{-1} = p^{-1}(L).$$

In other words, assuming a GRF probabilistic model with inverse covariance matrix $p(L)$ leads to a stationary graph signal with a PSD $= p^{-1}$. However a stationary graph signal is not necessarily a GRF. Indeed, stationarity assumes statistical properties on the signal that are not necessarily based on Gaussian distribution.

In Section 3 of [13], Gadde and Ortega have presented a GMRF model for graph signals. But they restrict themselves to the case where $p(L) = L + \delta I$. Following a similar approach Zhang et al. [14] link the inverse covariance matrix of a GMRF with the Laplacian. Our approach is much broader than these two contributions since we do not make any assumption on the function $p(L)$. Finally, we exploit properties of stationary signals, such as the characterization of the PSD, to explicitly solve signal processing problems in Section 5.

4 Estimation of the signal PSD

As the PSD is central in our method, we need a reliable and scalable way to compute it. Equation (7) suggests a direct estimation method using the Fourier transform of the covariance matrix. We could thus estimate the covariance $\Sigma_\mathbf{x}$ empirically from N_s realizations $\{x_n\}_{n=1\dots,N_s}$ of the stochastic graph signal $\mathbf{x}$, as

$$\bar{\Sigma}_\mathbf{x}[i,j] = \frac{1}{N_s - 1} \sum_{n=1}^{N_s} (x_n[i] - \bar{m}_\mathbf{x}[i]))(x_n[j] - \bar{m}_\mathbf{x}[j])^*,$$

where $\bar{m}_\mathbf{x}[i] = \sum_{n=1}^{N_s} x_n[i]$. Then our estimate of the PSD would read

$$\bar{\gamma}_\mathbf{x}(\lambda_\ell) = U^* \bar{\Sigma}_\mathbf{x} U[\ell, \ell].$$

Unfortunately, when the number of nodes is considerable, this method requires the diagonalization of the Laplacian, an operation whose complexity in the general case scales as $O(N^3)$ for the number of operations and $O(N^2)$ for memory requirements. Additionally, when the number of available realizations N_s is small, it is not possible to obtain a good estimate of the covariance matrix. To overcome these issues, inspired by Bartlett [4] and Welch [3], we propose to use a graph generalization of the Short Time Fourier transform [11] to construct a scalable estimation method.

Bartlett's method can be summarized as follows. After removing the mean, the signal is first cut into equally sized segments without overlap. Then, the Fourier transform of each segment is computed. Finally, the PSD is obtained by averaging over segments the squared amplitude of the Fourier coefficients. Welch's method is a generalization that works with overlapping segments.

On the other hand, we can see the PSD estimation of both methods as the averaging over time of the squared coefficients of a Short Time Fourier Transform (STFT). Let x be a realization of the stochastic graph signal $\mathbf{x}$ and $\tilde{x} = x - \bar{m}_{\tilde{x}}$ with $\bar{m}_\mathbf{x}[n] = \frac{1}{N}\sum_{n=1}^{N} x[n] = c$, the classical PSD estimator can thus be written as

$$\bar{\gamma}_\mathbf{x}[\ell] = \frac{\sum_{n=1}^{N} (\mathrm{STFT}\{\tilde{x}\}[\ell, n])^2}{N\|g\|_2^2},$$

where g is the window used for the STFT. This is shown in Figure 4.

Method Our method is based on this idea, using the windowed graph Fourier transform [11]. Instead of a translated rectangular window in time, we use a kernel g shifted by multiples of a step τ in the spectral domain, i.e.

$$g_m(\lambda_\ell) = g(\lambda_\ell - m\tau), \qquad m = 1 \dots M, \quad \tau = \frac{\lambda_{\max}}{M}.$$

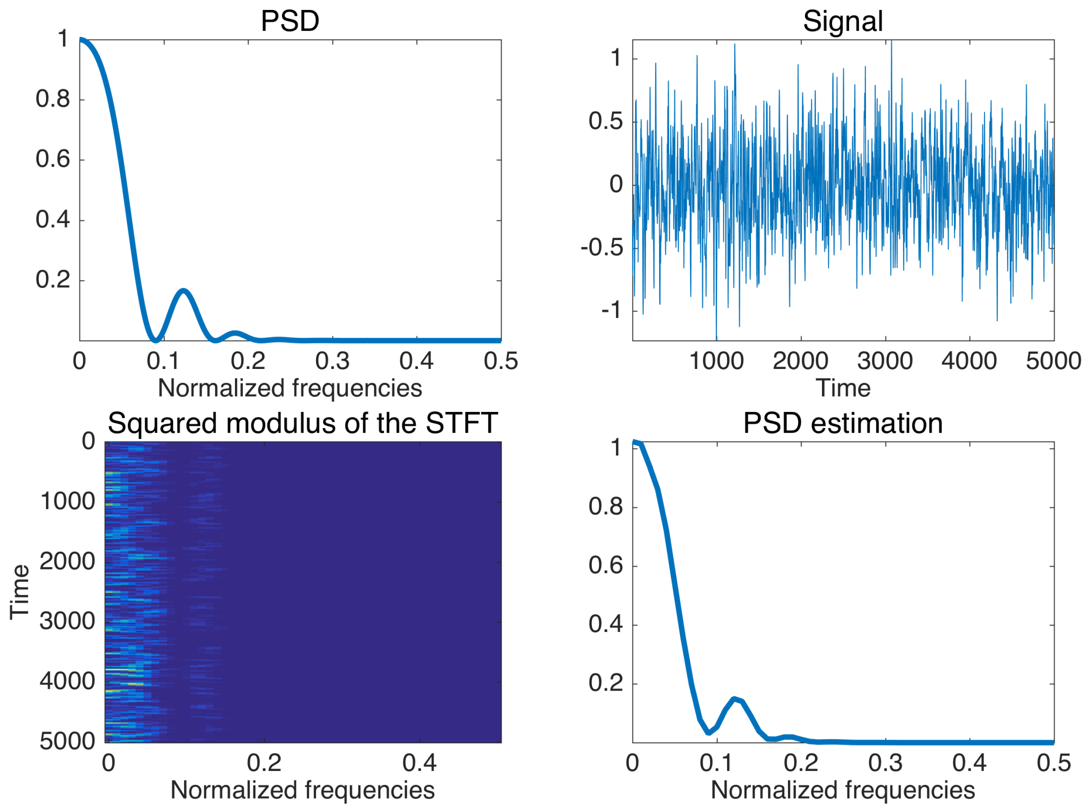


Figure 4: Illustration of the PSD estimation process for a temporal signal. Top right: original PSD. Top left: a stationary signal. Bottom right: The squared modulus of the STFT of the signal. Bottom left: Sum of the STFT squared coefficients over time. We observe that averaging the squared STFT coefficients approximate well the PSD. This method is a version of to the Welch method that is generalizable to graphs.

We then localize each spectral translation at each individual node of the graph. The coefficients of the graph windowed Fourier transform can be seen as a matrix with elements

$$C[i,m] = \langle x, \mathcal{T}_i g_m \rangle = [g_m(L)x]_i \,.$$

Our algorithm consists in averaging the squared coefficients of this transform over the vertex set. Because graphs have an irregular spectrum, we additionally need a normalization factor which is given by the norm of the window g_m: $\|g_m\|_2^2 = \sum_\ell g(\lambda_\ell - m\tau)^2$. Note that this norm will vary for the different m. Our final estimator reads :

$$\bar{\gamma}_{\mathbf{x}}(m\tau) = \frac{\|g_m(L)x\|_2^2}{\|g_m\|_2^2} = \frac{\sum_{i=1}^N C[i,m]^2}{\|g_m\|_2^2}, \tag{10}$$

where x is a single realization of the stationary stochastic graph signal $\mathbf{x}$. This estimator provides a discrete approximation of the PSD. Interpolation is used to obtain a continuous estimator. This approach avoids the computation of the eigenvectors and the eigenvalues of the Laplacian.

Our complete estimation procedure is as follows. First, we design a filterbank by choosing a mother function g (for example a Gaussian $g(\lambda) = e^{-\lambda^2/\sigma^2}$). A frame is then created by shifting uniformly M times g in the spectral domain: $g_m(\lambda) = g(\lambda - m\tau) = e^{-(\lambda - m\tau)^2/\sigma^2}$. Second, we compute the estimator $\bar{\gamma}_x(m\tau)$ from the stationary signal $\mathbf{x}$. Note that if we have access to K_1 realizations $\{x_k\}_{k=1,\dots,K_1}$ of the stationary signal, we can of course average them to further reduce the variance using $\mathbb{E}\left(\|g_m(L)\tilde{\mathbf{x}}\|_2^2\right) \approx 1/K_1 \sum_k \|g_m(L)\tilde{x}_k\|_2^2$. Third we use the following trick to quickly approximate $\|g_m\|_2^2$. Using K_2 randomly-generated Gaussian normalized zero centered white signals, we estimate

$$\mathbb{E}\left(\|g_m(L)\mathbf{w}\|_2^2\right) = \|g_m\|_2^2.$$

Finally, the last step consists in computing the ratio between the two quantities and interpolating the discrete points $(m\tau, \left(g * \gamma_{\mathbf{x}}\right)(m\tau))$.

Variance of the estimator Studying the bias of (10) reveals its interest :

$$\frac{\mathbb{E}\left(\|g_m(L)\tilde{\mathbf{x}}\|_2^2\right)}{\|g_m\|_2^2} = \frac{\sum_{\ell=0}^{N-1}\left(g(\lambda_\ell - m\tau)\right)^2 \gamma_{\mathbf{x}}(\lambda_\ell)}{\sum_{\ell=0}^{N-1}\left(g(\lambda_\ell - m\tau)\right)^2}, \tag{11}$$

where $\mathbf{x}$ is the stationary stochastic graph signal. For a filter g well concentrated at the origin, (11) gives a smoothed estimate of $\gamma_{\mathbf{x}}(m\tau)$. This smoothing corresponds to the windowing operation in the vertex domain: the less localized the kernel g in the spectral domain, the more pronounced the smoothing effect in (11) and the more concentrated the window in the vertex domain. It is very interesting to note we recover the traditional trade-off between bias and variance in non-parametric spectral estimation. Indeed, if g is very sharply localized on the spectrum, ultimately a Dirac delta, the estimator (10) is unbiased. Let us now study the variance. Intuitively, if the signal is correlated only over small regions of the vertex set, we could isolate them with localized windows of a small size and averaging those uncorrelated estimates together would reduce the variance. These small size windows on the vertex set correspond to large band-pass kernel g_m and therefore large bias. However, if those correlated regions are large, and this happens when the PSD is localized in low-frequencies, we cannot hope to benefit from vertex-domain averaging since the graph is finite. Indeed the corresponding windows g_m on the vertex set are so large that a single window spans the whole graph and there is no averaging effect: the variance increases precisely when we try to suppress the bias.

Experimental assessment of the method Figure 5 shows the results of our PSD-estimation algorithm on a 10-nearest neighbors graph of $20'000$ nodes (random geometric graph, weighted with an exponential kernel) and only $K = 1$ realization of the stationary graph signal. We compare the estimation using frames of $M = 10, 30, 100$ Gaussian filters. The parameters σ and τ are adapted to the number of filters such that the shifted windows have an overlap of approximately 2 ($\tau = \sigma^2 = \frac{(M+1)\lambda_{\max}}{M^2}$). For this experiment K_2 is set to 4 and the Chebysheff polynomial order is 30 The estimated curves are smoothed versions of the PSD.

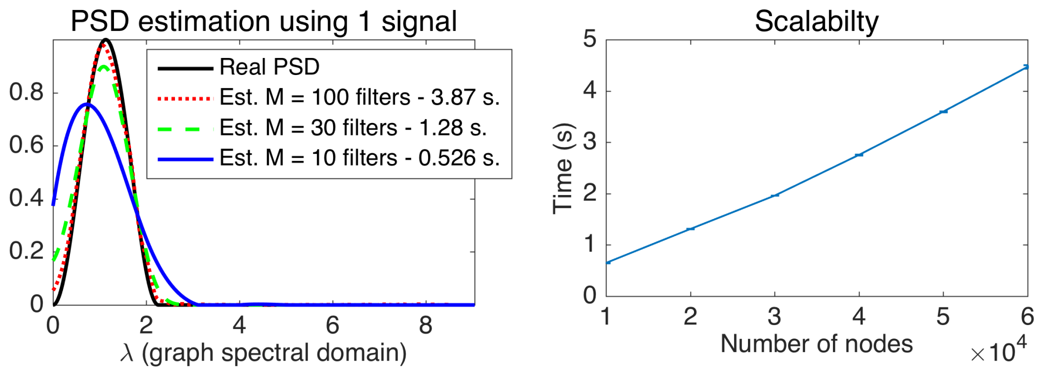


Figure 5: Left: PSD estimation on a graph of $20'000$ nodes with $K = 1$ measurements. Our algorithm is able to successively estimate the PSD of a signal. Right: Computation time versus size of the graph. We use $m = 30$ filters. The algorithm scales linearly with the number of edges.

Complexity analysis The approximation scales with the number of edges of the graph $\mathcal{O}(|\mathcal{E}|)$, (which is proportional to N in many graphs). Precisely, our PSD estimation method necessitates $(K + K_2)M$ filtering operations (with M the number of shifts of g). A filtering operation costs approximatively $O_c|E|$, with O_c the order of the Chebysheff polynomial [20]. The final computational cost of the method is thus $\mathcal{O}\left(O_c(K + K_2)M|\mathcal{E}|\right)$.

Error analysis The difference between the approximation and the exact PSD is caused by three different factors.

1. The inherent bias of the estimator, which is now directly controlled by the parameter σ.
2. We estimate the expected value using K_1 realization of a the signal (often $K_1 = 1$). For large graphs $N \gg K_1$ and a few filters $M \ll N$, this error is usually low because the variance of $\|g_m(L)\tilde{\mathbf{x}}\|_2^2$ is inversely proportional to bias. The estimation error improves as $\frac{1}{K_1}$.

3. We use a fast-filtering method based on a polynomial approximation of the filter. For a rough approximation, $\sigma \gg \frac{\lambda_{\max}}{N}$, this error is usually negligible. However, in the other cases, this error may become large.

5 Graph Wiener filters and optimization framework

Using stationary signals, we can naturally extend the framework of Wiener filters [25] largely used in signal processing for Mean Square Error (MSE) optimal linear prediction. Wiener filters for graphs have already been succinctly proposed in [2, pp 100]. Since the construction of Wiener filters is very similar for non-graph and graph signals, we present only the latter here. The main difference is that the traditional frequencies are replaced by the graph Laplacian eigenvalues[3] λ_ℓ. igure 6 presents the Wiener estimation scheme.

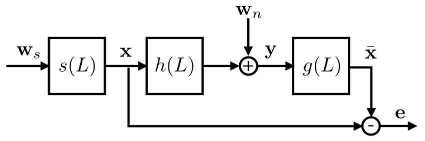


Figure 6: Wiener estimation scheme. $\mathbf{w}_s$ is a centered random variable with covariance I. $\mathbf{w}_s$ generates the stationary stochastic signal $\mathbf{x}$ thanks to the filter $s(L)$. The random variable $\mathbf{y}$ is then generated by filtering $\mathbf{x}$ through $h(L)$ and adding uncorrelated noise $\mathbf{w}_n$ with PSD $n(\lambda)$. The estimator of $\mathbf{x}$ given $\mathbf{y}$: $\bar{\mathbf{x}}|\mathbf{y}$ is obtained with the Wiener filter $g(L)$. The estimation error is denoted $\mathbf{e}$. For clarity, we assume that $m_{\mathbf{x}} = 0$

Graph Wiener filtering The Wiener filter can be used to produce a mean-square error optimal estimate of a stationary signal under a linear but noisy observation model. Let us consider the GWSS stochastic signal $\mathbf{x}$ with PSD of $s^2(\lambda_\ell)$. For simplicity, we assume $m_{\mathbf{x}} = 0$. The measurements $\mathbf{y}$ are given by:

$$\mathbf{y} = h(L)\mathbf{x} + \mathbf{w}_n, \tag{12}$$

where $h(L)$ is a graph filter and $\mathbf{w}_n$ additive uncorrelated noise of PSD $n(\lambda_\ell)$.

To recover $\mathbf{x}$, Wiener filters can be extended to the graph case:

$$g(\lambda_\ell) = \frac{h(\lambda_\ell)s^2(\lambda_\ell)}{h^2(\lambda_\ell)s^2(\lambda_\ell) + n(\lambda_\ell)}. \tag{13}$$

The expression above can be derived by exactly mimicking the classical case and minimises the expected quadratic error, which can be written as:

$$e[\ell] = \mathbb{E}\left(\left(\hat{\mathbf{x}}[\ell] - \hat{\bar{\mathbf{x}}}[\ell]\right)^2\right) = \mathbb{E}\left(\hat{\mathbf{x}}[\ell] - g(\lambda_\ell)\hat{\mathbf{y}}[\ell]\right)^2,$$

where $\bar{\mathbf{x}} = g(L)\mathbf{y}$ is the estimator of $\mathbf{x}$ given $\mathbf{y}$. Theorem 5 proves the optimality of this filter for the graph case.

Wiener optimization In this contribution, we would like to address a more general problem. Let us suppose that our measurements are generated as:

$$\mathbf{y} = H\mathbf{x} + \mathbf{w}_n, \tag{14}$$

where the GWSS stochastic graph signal $\mathbf{x}$ has a PSD denoted $s^2(\lambda_\ell)$ and the noise $\mathbf{w}_n$ a PSD of $n(\lambda_\ell)$. We assume $\mathbf{x}$ and $\mathbf{w}_n$ to be uncorrelated. H is a general linear operator not assumed to be diagonalizable with L. As a result, we cannot build a Wiener filter that constructs a direct estimation of the signal x. If $\mathbf{x}$ varies smoothly on the graph, i.e is low frequency based, a classic optimization scheme would be the following:

$$\bar{\mathbf{x}}|\mathbf{y} = \arg\min_{\mathbf{x}} \|H\mathbf{x} - \mathbf{y}\|_2^2 + \beta\mathbf{x}^*L\mathbf{x}. \tag{15}$$

[3]The graph eigenvalues are equivalent to classical squared frequencies.

This optimization scheme presents two main disadvantages. Firstly, the parameter β must be tuned in order to remove the best amount of noise. Secondly, it does not take into account the data structure characterized by the PSD $s^2(\lambda_\ell)$.

Our solution to overcome these issues is to solve the following optimization problem that we suggestively call *Wiener optimization*

$$\bar{\mathbf{x}}|\mathbf{y} = \arg\min_{\mathbf{x}} \|H\mathbf{x} - \mathbf{y}\|_2^2 + \|w(L)(\mathbf{x} - m_{\mathbf{x}})\|_2^2, \tag{16}$$

where $w(\lambda_\ell)$ is the Fourier penalization weights. These weights are defined as

$$w(\lambda_\ell) = \left| \frac{\sqrt{n(\lambda_\ell)}}{s(\lambda_\ell)} \right| = \frac{1}{\sqrt{SNR(\lambda_\ell)}}.$$

Notice that compared to (15), the parameter β is exchanged with the PSD of the noise. As a result, if the noise parameters are unknown, Wiener optimization does not solve completely the issue of finding the regularization parameter. In the noise-less case, one can alternatively solve the following problem

$$\bar{\mathbf{x}} = \arg\min_{x} \|s^{-1}(L)(\mathbf{x} - m_{\mathbf{x}})\|_2^2, \qquad \text{s. t. } H\mathbf{x} = \mathbf{y}. \tag{17}$$

Problem (16) generalizes Problem (15) which assumes implicitly a PSD of $\frac{1}{\lambda_\ell}$ and a constant noise level of γ across all frequencies. Note that this framework generalizes two main assumptions done on the data in practice:

1. The signal is smooth on the graph, i.e: the edge derivative has a small ℓ_2-norm. As seen before this is done by setting the PSD as $\frac{1}{\lambda_\ell}$. This case is studied in [16].

2. The signal is band-limited, i.e it is a linear combination of the k lowest graph Laplacian eigenvectors. This class of signal simply have a null PSD for $\lambda_\ell > \lambda_k$.

Theoretical motivations for the optimization framework The first motivation is intuitive. The weight $w(\lambda_\ell)$ heavily penalizes frequencies associated to low SNR and vice versa.

The second and main motivation is theoretical. If we have a Gaussian Random multivariate signal with i.i.d Gaussian noise, then Problem (16) is a MAP estimator.

Theorem 3. *If* $\mathbf{x} \sim \mathcal{N}\left(0, s^2(L)\right)$ *and* $\mathbf{w}_n \sim \mathcal{N}\left(0, \sigma^2\right)$, *i.e: u is GWSS and Gaussian, then problem* (16) *is a MAP estimator for* $\mathbf{x}|\mathbf{y}$

The proof is given in Appendix B.

Theorem 4. *If* $\mathbf{x}$ *is GWSS with PSD* $s^2(L)$ *and* $\mathbf{w}_n$ *is i.i.d white noise, i.e:* $n(\ell) = \sigma^2$, *then problem* (16) *leads to the linear minimum mean square estimator:*

$$\mathbf{x}|\mathbf{y} \quad = \quad \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1}\mathbf{y} + \left(I - \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1}H\right) m_{\mathbf{x}} \tag{18}$$

with $\Sigma_{\mathbf{xy}} = s^2(L)H^*$ *and* $\Sigma_{\mathbf{y}} = Hs^2(L)H^*$

The proof is given in Appendix D.

Additionally, when H is jointly diagonalizable with L, Problem (16) can be solved by a single filtering operation.

Theorem 5. *If the operator H is diagonalizable with L, (i.e: $H = h(L) = Ua(\Lambda)U^*$), then problem* (16) *is optimal with respect of the weighting w in the sense that its solution minimizes the mean square error:*

$$\mathbb{E}\left(\|\mathbf{e}\|_2^2\right) = \mathbb{E}\left(\|\bar{\mathbf{x}} - \mathbf{x}\|_2^2\right) = \mathbb{E}\left(\sum_{i=1}^{N} (\bar{\mathbf{x}}[i] - \mathbf{x}[i])^2\right).$$

Additionally, the solution can be computed by the application of the corresponding Wiener filter.

The proof is given in Appendix C.

The last motivation is algorithmic and requires the knowledge of proximal splitting methods [26, 27]. Problem (16) can be solved by a splitting scheme that minimizes iteratively each of the terms. The minimization of the regularizer, i.e the proximal operator of $\|w(L)\tilde{x}\|_2^2$, becomes a Wiener de-noising operation:

$$\begin{aligned}\operatorname{prox}_{\frac{1}{2}\|w(L))\tilde{x}\|_2^2}(y) &= m_{\mathbf{x}} + \arg\min_x \|w\mathcal{F}\tilde{x}\|_2^2 + \|\tilde{x}-\tilde{y}\|_2^2 \\ &= m_{\mathbf{x}} + g(L)\tilde{y} = m_{\mathbf{x}} + g(L)(y - m_{\mathbf{x}})\end{aligned}$$

with

$$g(\lambda_\ell) = \frac{1}{1+w^2(\lambda_\ell)} = \frac{s^2(\lambda_\ell)}{s^2(\lambda_\ell)+n(\lambda_\ell)}.$$

Advantage of the Wiener optimization framework over a Gaussian MAP estimator Theorem 3 shows that the optimization framework is equivalent to a Gaussian MAP estimator. In practice, when the data is only close to stationary, the true MAP estimator will perform better than Wiener optimization. So one could ask why we bother defining stationarity on graphs. Firstly, assuming stationarity allows us for a more robust estimate of the covariance matrix. This is shown is in Figure 5, where only one signal is used to estimate the PSD (and thus the covariance matrix). Another example is the USPS experiment presented in the next section. We estimate the PSD by using only 20 digits. The final result is much better than a Gaussian MAP based on the empirical covariance. Secondly, we have a scalable solution for Problem (16) (See Algorithm 1 below). On the contrary the classical Gaussian MAP estimator requires the explicit computation of a large part of the covariance matrix and it's inverse, which are both not scalable operations.

Solving Problem (16) Note that Problem (16) can be solved with a simple gradient descent. However, for a large number of nodes N, the matrix $w(L)$ requires $\mathcal{O}(N^3)$ operations to be computed and $\mathcal{O}(N^2)$ bits to be stored. This difficulty can be overcome by applying its corresponding filter operator at each iteration. As already mentioned, the cost of the approximation scale with the number of edges $\mathcal{O}(O_c|E|)$ [20].

When $s(\lambda_\ell) \approx 0$ for some λ_ℓ the operator $w(L)$ becomes badly conditioned. To overcome this issue, Problem (16) can be solved efficiently using a forward-backward splitting scheme [28, 26, 27]. The proximal operator of $\|w(L)\tilde{x}\|_2^2$ has been given above and we use the term $\|Hx-y\|_2^2$ as the differentiable function. Algorithm 1 uses an accelerated forward backward scheme [29] to solve Problem (16) where β is the step size (we select $\beta = \frac{1}{2\lambda_{\max}(H)^2}$), ϵ the stopping tolerance, J the maximum number of iterations and δ is a very small number to avoid a possible division by 0.

Algorithm 1 Fast Wiener optimization to solve (16)

```
INPUT: $z_1 = x$, $u_0 = x$, $t_1 = 1$, $\epsilon > 0$, $\beta \leq \frac{1}{2\lambda_{\max}(H)^2}$
SET: $g(\lambda) = \frac{s^2(\lambda)}{s^2(\lambda)+\beta n(\lambda)}$        ▷ Wiener filter
for $j = 1, \dots J$ do
    $v = z_j - \beta H^*(Hz_j - y)$        ▷ Gradient step
    $u_{j+1} = g(L)v$        ▷ Proximal step
    $t_{j+1} = \frac{1+\sqrt{1+4t_j^2}}{2}$        ▷ FISTA scheme
    $z_{j+1} = z_j + \frac{t_j - 1}{t_{j+1}}(u_j - u_{j-1})$        ▷ Update step
    if $\frac{\|z_{j+1}-z_j\|_F^2}{\|z_j\|_F^2+\delta} < \epsilon$ then        ▷ Stopping criterion
        BREAK
    end if
end for
SOLUTION: $z_J$
```

6 Evidence of graph stationarity: illustration with USPS

Stationarity may not be an obvious hypothesis for a general dataset, since our intuition does not allow us to easily capture the kind of shift invariance that is really implied. In this section we give additional insights on stationarity from a more experimental point of view. To do so, we will show that the well-known USPS dataset is close to stationary on a nearest neighbor graph. We show similar results with a dataset of faces.

Data \Graph	2-dimensional grid	20 nearest neighbors graph
Shifted all digits	0.86	1
All digits	0.66	0.79
Digit 3	0.64	0.83
Digit 7	0.52	0.79
Digit 9	0.52	0.81

Table 2: $s_r(\Gamma) = \frac{\|\text{diag}(\Gamma)\|_2}{\|\Gamma\|_F}$: stationarity measures for different graphs and different datasets. The nearest neighbors graph adapts to the data. The individual digits are stationary with the nearest neighbor graph.

Images can be considered as signals on the 2-dimensional euclidean plane and, naturally, when the signal is sampled, a grid graph is used as a discretization of this manifold. The corresponding eigenbasis is the 2 dimensional DCT[4]. Following Julesz's initial assumption [30] many papers have exploited the fact that natural texture images are stationary 2-dimensional signals, i.e stationary signals on the grid graph. In [31], the authors go one step further and ask the following question: suppose that pixels of images have been permuted, can we recover their relative two-dimensional location? Amazingly, they answer positively adding that only a few thousand images are enough to approximatively recover the relative location of the pixels. The grid graph seems naturally encoded within images.

The observation of [31] motivates the following experiment involving stationarity on graphs. Let us select the USPS data set which contains 9298 digit images of 16×16 pixels. We create 5 classes of data: (a) the circularly shifted digits[5], (b) the original digits and (c), (d) and (e) the classes of digit 3, 7 and 9. As a pre-processing step, we remove the mean of each pixel, thus forcing the first moment to be 0, and focus on the second moment. For those 5 cases, we compute the covariance matrix Σ and its graph covariance matrix,

$$\Gamma = U^*\Sigma U, \tag{19}$$

for 2 different graphs: (a) the grid and (b) the 20 nearest neighbors graph. In this latter case, each node is a pixel and is associated to a feature vector containing the corresponding pixel value of all images. We use the squared euclidean distance between feature vectors to define edge weights. We then compute the stationarity level of each class of data with both graphs using the following measure:

$$s_r(\Gamma) = \left(\frac{\sum_\ell \Gamma_{\ell,\ell}^2}{\sum_{\ell_1}\sum_{\ell_2} \Gamma_{\ell_1,\ell_2}^2} \right)^{\frac{1}{2}} = \frac{\|\text{diag}(\Gamma)\|_2}{\|\Gamma\|_F}. \tag{20}$$

The closer $s_r(\Gamma)$ is to 1, the more diagonal the matrix Γ is and the more stationary the signal. Table 2 shows the obtained stationarity measures. The less universal the data, the less stationary it is on the grid. Clearly, specificity inside the data requires a finer structure than a grid. This is confirmed by the behavior of the nearest neighbors graph. When only one digit class is selected the nearest neighbors graph still yields very stationary signals.

Let us focus on the digit 3. For this experiment, we build a 20 nearest neighbors graph with only 50 samples. Figure 7 shows the eigenvectors of the Laplacian and of the covariance matrix. Because of stationarity, they are very similar. Moreover, they have a 3-like shape. Since the data is almost stationary, we can use the associated graph and the PSD to generate samples by filtering i.i.d Gaussian noise with the following PSD based kernel: $g(\lambda_\ell) = \sqrt{\Gamma_{\ell,\ell}}$. The resulting digits have a 3-like shape confirming the that the class is stationary on the nearest neighbors graph.

To further illustrate this phenomenon on a different dataset, we use the CMUPIE set of cropped faces. With a nearest neighbor graph we obtained a stationarity level of $s_r = 0.92$. This has already been observed in [32] where the concept of Laplacianfaces is introduced. Finally in [33] the authors succesfully use the graph between features to improve the quality of a low-rank recovery problem. The reason seems to be that the principal components of the data are the lowest eigenvectors of the graph, which is again a stationarity assumption.

To intuitively motivate the effectiveness of nearest neighbors at producing stationary signals, let us define the centering operator $J = I - \mathbf{1}\mathbf{1}^\top/N$. Given K signal x_k, the matrix of average squared distances

[4]This is a natural extension of [19]

[5]We performed all possible shifts in both directions. Because of this, the covariance matrix becomes Toeplitz

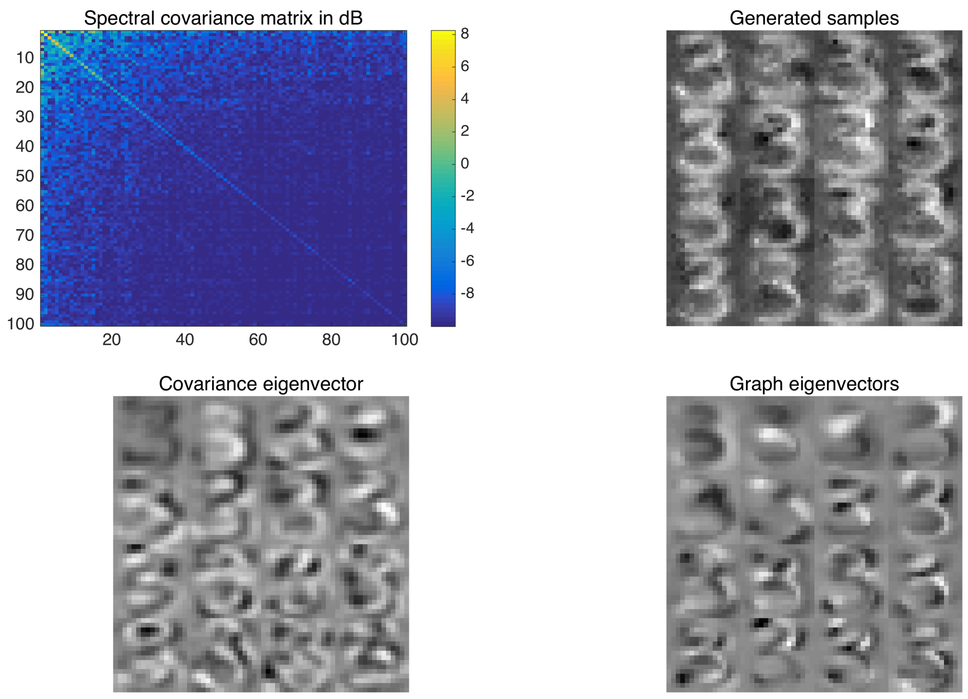


Figure 7: Studying the number 3 of USPS using a 20-neighbors graph. Top left: Spectral covariance matrix of the data (Note the diagonal shape of the matrix). We only display the upper left part for better visibility. Top right: generated samples by filtering Gaussian random noise on the graph. Bottom left: Covariance eigenvectors associated with the 16 highest eigenvalues. Bottom right: Laplacian eigenvectors associated to the 16 smallest non-zero eigenvalues. Because of stationarity, Laplacian eigenvectors are similar to the covariance eigenvectors.

between the centered features ($\sum_{i=1}^{N} x_k[i] = 0$) is directly proportional to the covariance matrix :

$$\bar{\Sigma}_{\mathbf{x}} = -\frac{1}{2} JDJ, \tag{21}$$

where $D[i,j] = \frac{1}{K}\sum_{k=1}^{K}\left(x_k[i] - x_k[j]\right)^2$ and $\bar{\Sigma}_{\mathbf{x}}[i,j] = \frac{1}{K}\sum_{k=1}^{K} x_k[i]x_k[j]$. The proof is given in Appendix E. The nearest-neighbors graph can be seen as an approximation of the original distance matrix, which pleads for using it as a good proxy destined at leveraging the spectral content of the covariance. Put differently, when using realizations of the signal as features and computing the k-NN graph we are connecting strongly correlated variables via strong edge weights.

7 Experiments

All experiments were performed with the GSPBox [34] and the UNLocBoX [35] two open-source softwares. The code to reproduce all figures of the paper can be downloaded at: `https://lts2.epfl.ch/rrp/stationarity/`. As the stationary signals are random, the reader may obtain slightly different results. However, conclusions shall remain identical. The models used in our comparisons are detailed in the Appendix A for completeness, where we also detail how the tuning of the parameters is done. All experiments are evaluated with respect to the Signal to Noise Ratio (SNR) measure:

$$\mathrm{SNR}(x, \dot{x}) = 10 \log \left(\frac{\mathrm{var}(x - \dot{x})}{\mathrm{var}(x)} \right)$$

7.1 Synthetic dataset

In order to obtain a first insight into applications using stationarity, we begin with some classical problems solved on a synthetic dataset. Compared to real data, this framework allows us to be sure that the signal is stationary on the graph.

Graph Wiener deconvolution We start with a de-convolution example on a random geometric graph. This can model an array of sensors distributed in space or simply a mesh. The signal is chosen with a low frequency band-limited PSD. To produce the measurements, the signal is convolved with the heat kernel $h(\lambda) = e^{-\tau\lambda}$. Additionally, we add some uncorrelated i.i.d Gaussian noise. The heat kernel is chosen because it simulates a heat diffusion process. Using de-convolution we aim at recovering the original signal before diffusion. For this experiment, we put ourselves in an ideal case and suppose that both the PSD of the input signal and the noise level are known.

Figure 8 presents the results. We observe that Wiener filtering is able to de-convolve the measurements. The second plot shows the reconstruction errors for three different methods: Tikhonov presented in problem (23), TV in (25) and Wiener filtering in (13). Wiener filtering performs clearly much better than the other methods because it has a much better prior assumption.

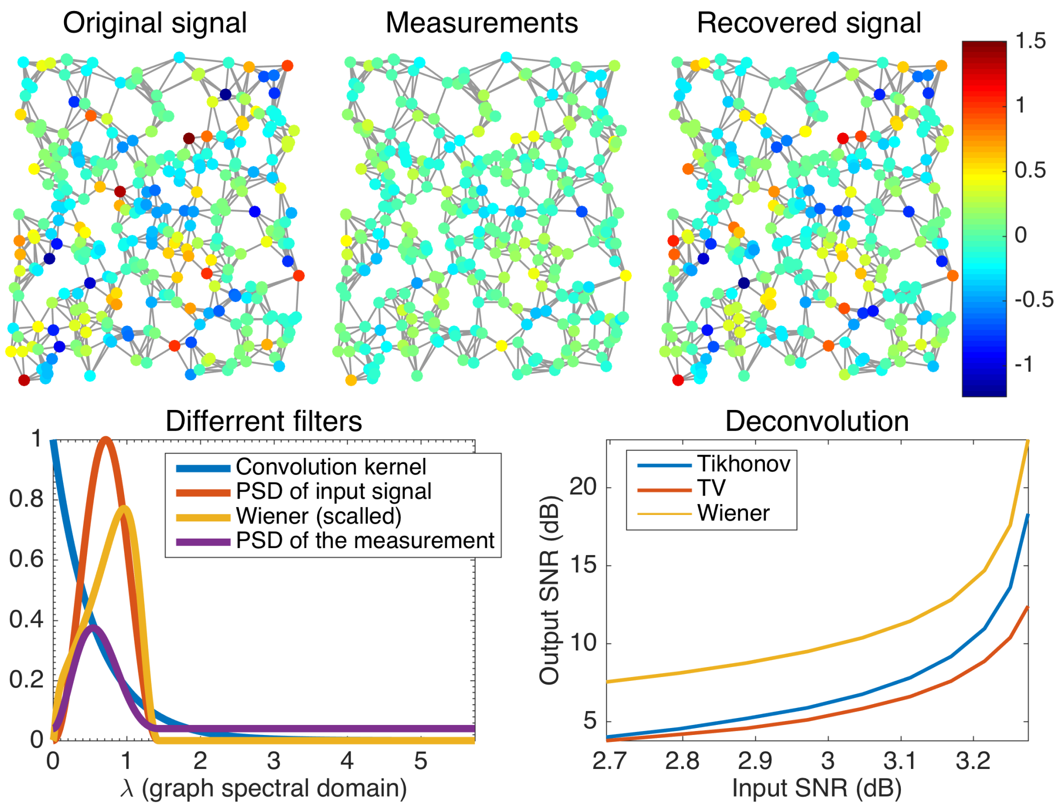


Figure 8: Graph de-convolution on a geometric random graph. The convolution kernel is $e^{-\frac{10x}{\lambda_{\max}}}$. Top: Signal and filters for a noise level of 0.16. Bottom: evolution of the error with respect of the noise.

Graph Wiener in-painting In our second example, we use Wiener optimization to solve an in-painting problem. This time, we suppose that the PSD of the input signal is unknown and we estimate it using 50 signals. Figure 9 presents quantitative results for the in-painting. Again, we compare three different optimization methods: Tikhonov (22), TV (25) and Wiener (16). Additionally we compute the classical MAP estimator based on the empirical covariance matrix (see [36] 2.23). Wiener optimization performs clearly much better than the other methods because it has a much better prior assumption. Even with 50 measurements, the MAP estimator performs poorly compared to graphs method. The reason is that the graph contains a lot of the covariance information. Note that the PSD estimated with only one measurement is sufficient to outperform Tikhonov and TV.

7.2 Meteorological dataset

We apply our methods to a weather measurements dataset, more precisely to the temperature and the humidity. Since intuitively these two quantities are correlated smoothly across space, it suggests that they are more or less stationary on a nearest neighbors geographical graph.

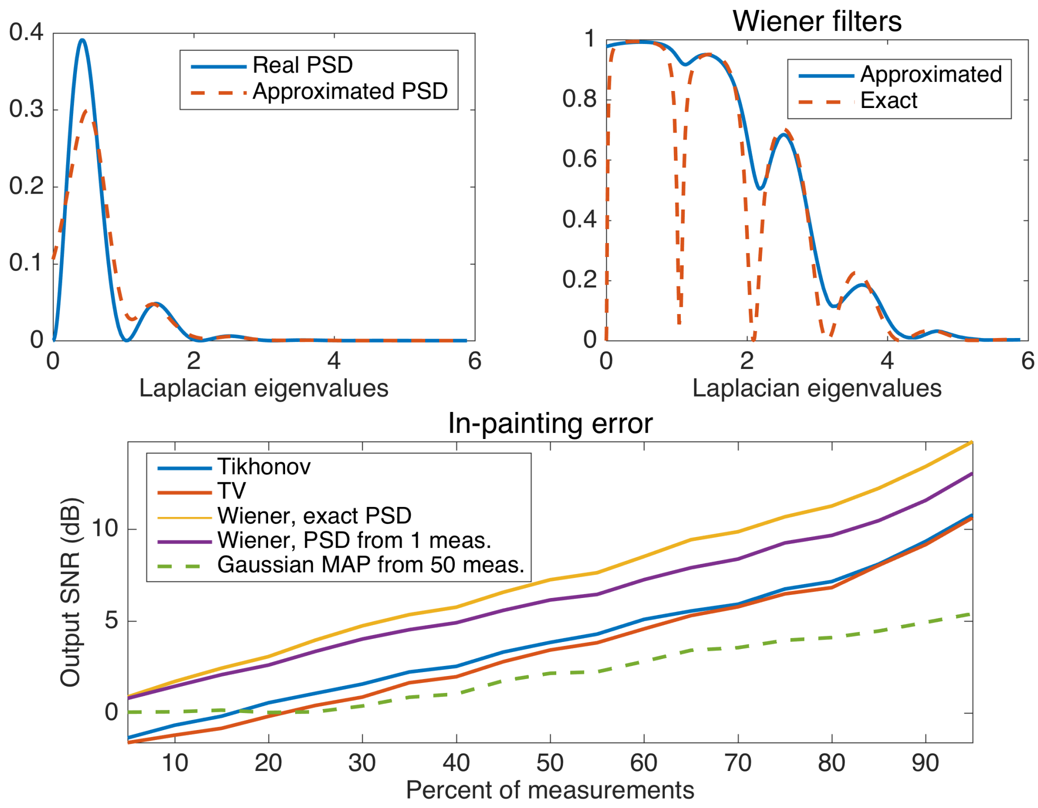


Figure 9: Wiener in-painting on a geometric graph of 400 nodes. Top: true VS approximated PSD and resulting Wiener filters. Bottom: in-painting relative error with respect to number of measurements.

The French national meteorological service has published in open access a dataset[6] with hourly weather observations collected during the Month of January 2014 in the region of Brest (France). From these data, we wish to ascertain that our method still performs better than the two other models (TV and Tikhonov) on real measurements. The graph is built from the coordinates of the weather stations by connecting all the neighbors in a given radius with a weight function $w(i,j) = e^{-d^2\tau}$ where τ is adjusted to obtain a average degree around 3 (τ, however, is not a sensitive parameter). For our experiments, we consider every time step as an independent realization of a GWSS signal. As sole pre-processing, we remove the temperature mean of each station independently. This is equivalent to removing the first moment. Thanks to the 744 time observation, we can estimate the covariance matrix and check whether the signal is stationary on the graph.

Prediction - Temperature The result of the experiment with temperatures is displayed in Figure 10. The covariance matrix shows a strong correlation between the different weather stations. Diagonalizing it with the Fourier basis of the graph shows that the meteorological instances are not really stationary within the distance graph as the resulting matrix is not really diagonal. However, even in this case, Wiener optimization still outperforms graph TV and Tikhonov models, showing the robustness of the proposed method. In our experiment, we solve an prediction problem with a mask operator covering 50 per cent of measurements and an initial average SNR of 13.4 dB . We then average the result over 744 experiments (corresponding to the 744 observations) to obtain the curves displayed in Figure 10. We observe that Wiener optimization performs always better than the two other methods.

Prediction - Humidity Using the same graph, we have performed another set of experiments on humidity observations. In our experiment, we solve an prediction problem with a mask operator covering 50% of measurements and various amount of noise. The rest of the testing framework is identical as for the temperature and the conclusions are similar.

[6]Access to the raw data is possible directly through our code or through the link `https://donneespubliques.meteofrance.fr/donnees_libres/Hackathon/RADOMEH.tar.gz`

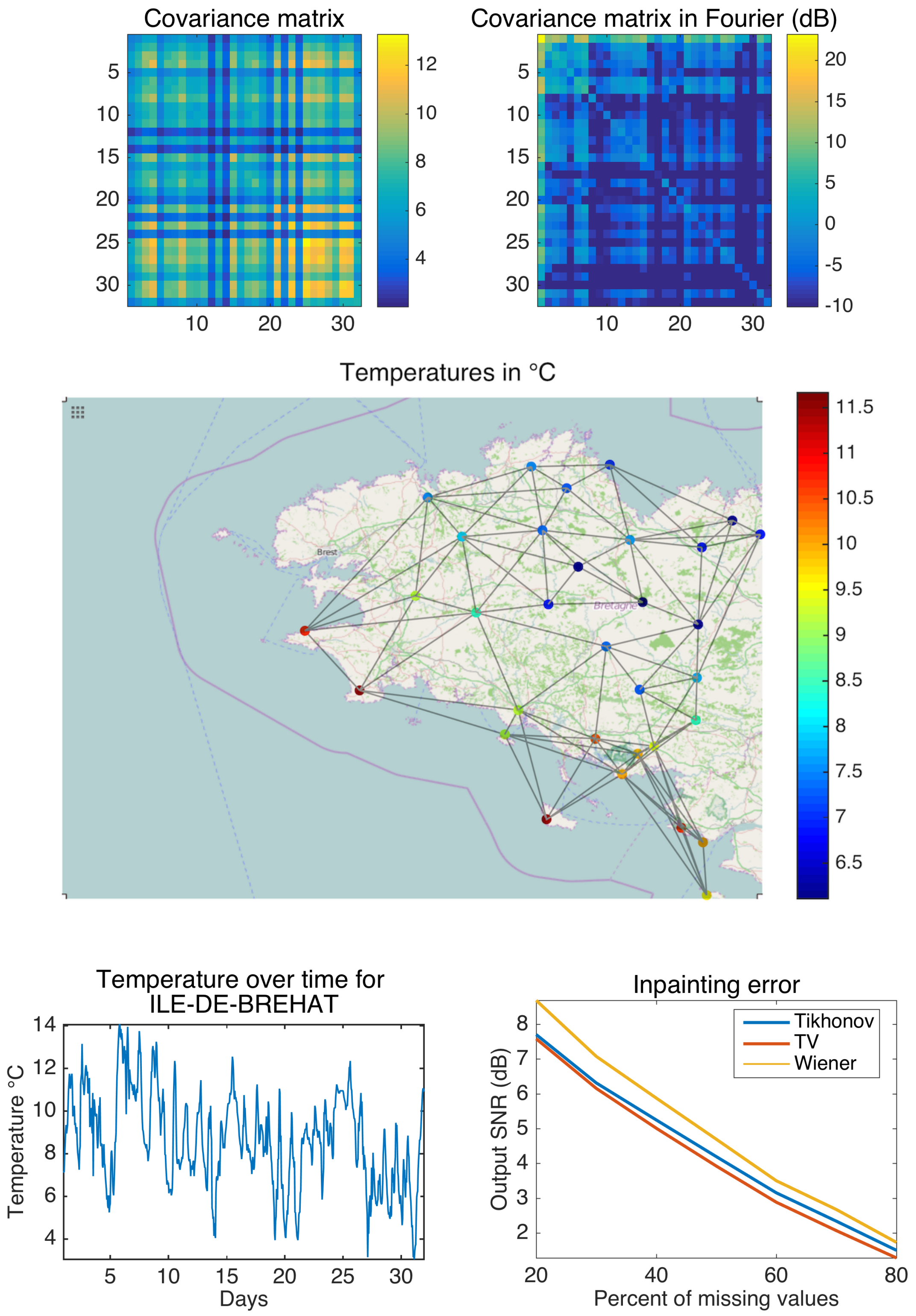


Figure 10: Top: Covariance matrices. Bottom left: A realization of the stochastic graph signal (first measure). Bottom center: the temperature of the Island of Brehat. Bottom right: Recovery errors for different noise levels.

7.3 USPS dataset

We perform the same kind of in-painting/de-noising experiment with the USPS dataset. For our experiments, we consider every digit as an independent realization of a GWSS signal. As sole pre-processing,

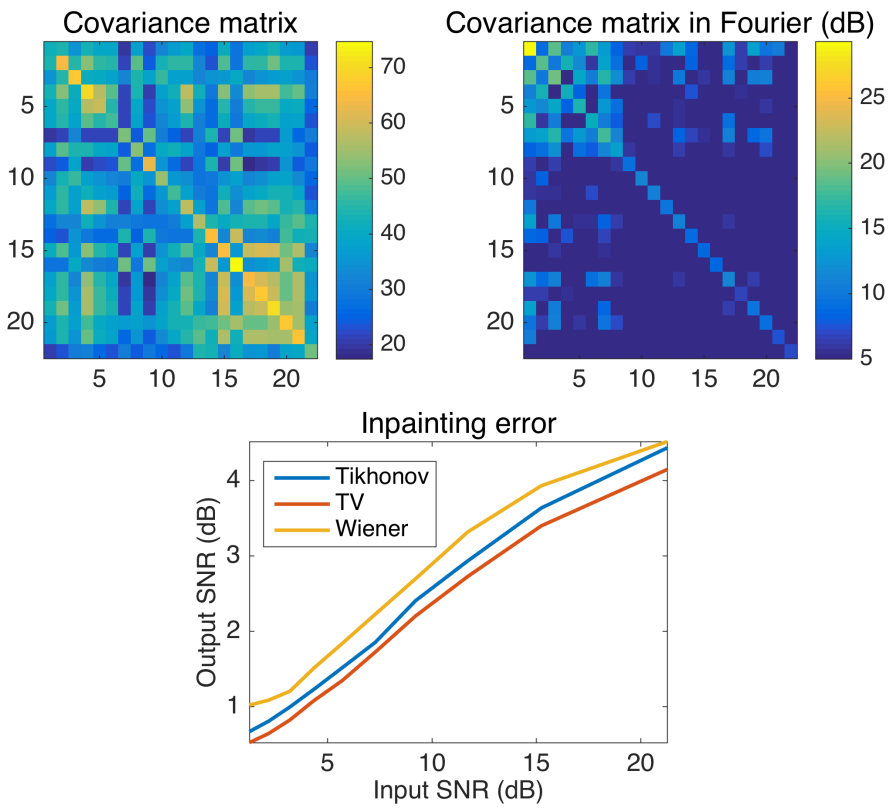


Figure 11: Top: Covariance matrices. Bottom: Recovery errors for different noise levels.

we remove the mean of each pixel separately. This ensures that the first moment is 0. We create the graph[7] and estimate the PSD using only the first 20 digits and we use 500 of the remaining ones to test our algorithm. We use a mask covering 50% of the pixel and various amount of noise. We then average the result over 500 experiments (corresponding to the 500 digits) to obtain the curves displayed in Figure 12[8]. For this experiment, we also compare to traditional TV de-noising [37] and Tikhonov de-noising. The optimization problems used are similar to (22). Additionally we compute the classical MAP estimator based on the empirical covariance matrix for the solution see ([36] 2.23). The results presented in Figure 12 show that graph optimization is outperforming classical techniques meaning that the grid is not the optimal graph for the USPS dataset. Wiener once again outperforms the other graph-based models. Moreover, this experiment shows that our PSD estimation is robust when the number of signals is small. In other words, using the graph allows us for a much better covariance estimation than a simple empirical average. When the number of measurements increases, the MAP estimator improves in performance and eventually outperforms Wiener because the data is close to stationary on the graph.

7.4 ORL dataset

For this last experiment, we use the ORL faces dataset. We have a good indication that this dataset is close to stationary since CMUPIE (a smaller faces dataset) is also close to stationary. Each image has $112 \times 92 = 10304$ pixels making it complicated to estimate the covariance matrix and to use a Gaussian MAP estimator. Wiener optimization on the other hand does not necessitate an explicit computation of the covariance matrix. Instead, we estimate the PSD using the algorithm presented in Section 4. A detailed experiment is performed in Figure 13. After adding Gaussian noise to the image, we remove randomly a percentage of the pixels. We consider the obtained image as the measurement and we

[7]The graph is created using patches of pixels of size 5×5. The pixels' patches help because we have only a few digits available. When the size of the data increases, a nearest neighbor graph performs even better.

[8]All parameters have been tuned optimally in a probabilistic way. This is possible since the noise is added artificially. The models presented in Appendix A have only one parameter to be tuned: ϵ which is set to $\epsilon = \sigma\sqrt{\#y}$, where σ is the variance of the noise and $\#y$ the number of elements of the vector y. In order to be fair with the MAP estimator, we construct the graph with the only 20 digits used in the PSD estimation.

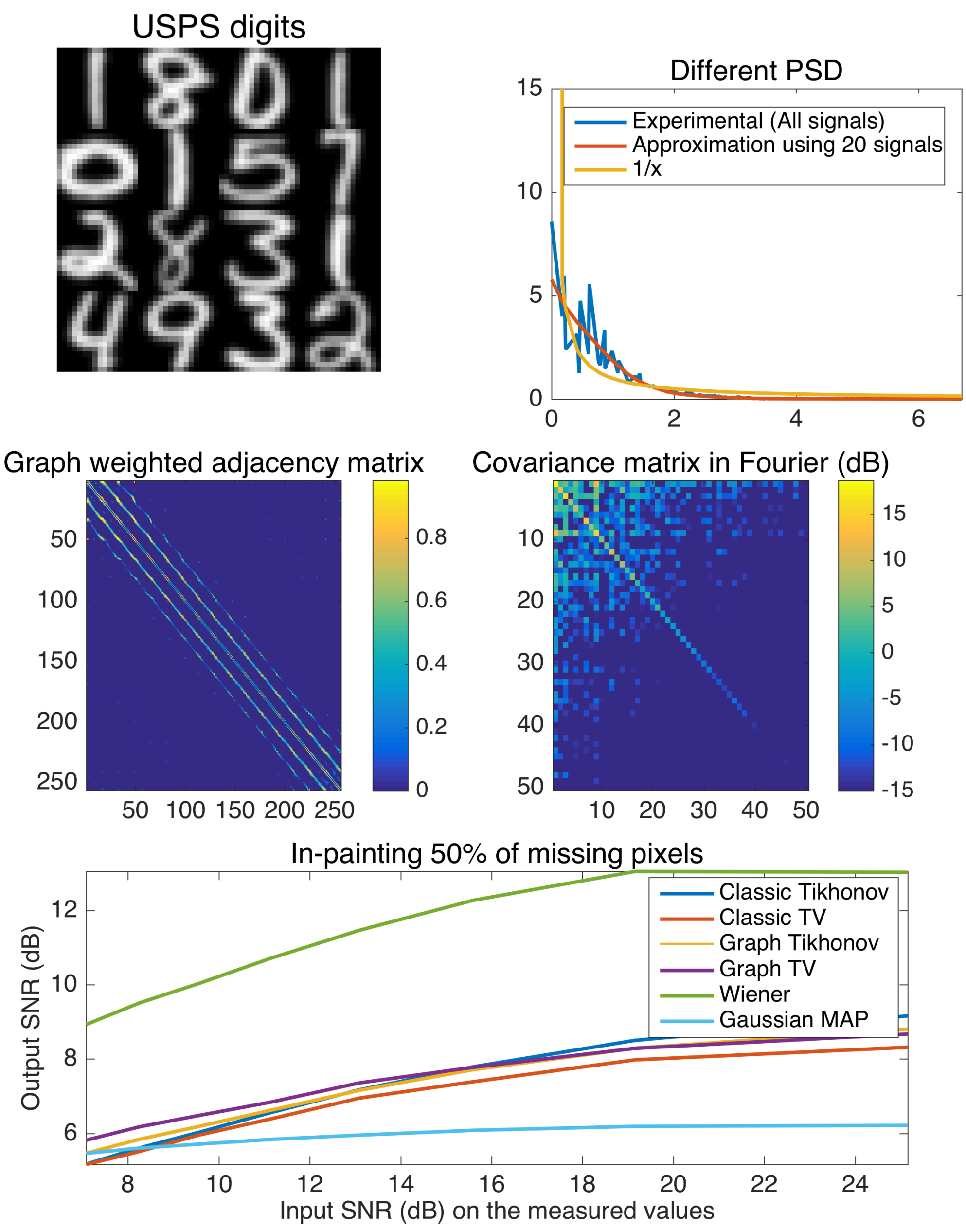


Figure 12: Top left: Some digits of the USPS dataset. Top right: Different PSDs. Compared to $\frac{1}{\lambda}$, the approximation is a smoothed version of the experimental PSD. Middle left: Weights matrix of the 10 nearest neighbors (patch) graph (The diagonal shape indicates the grid base topology of the graph). Middle right: spectral covariance matrix for the first 50 graph frequencies. Since we only use 20 digits for the graph construction, the stationarity level is low. Nevertheless, Wiener optimization outperforms other methods. Bottom: Recovery errors for different noise levels. Methods using the graph perform better. Even if the data is not stationary on the graph, the stationarity assumption helps a lot in the recovery.

reconstruct the original image using TV, Tikhonov and Wiener priors. In Figure 14, we display the reconstruction results for various noise levels. We create the graph with 300 faces[9] and estimate the PSD with 100 faces. We test the different algorithms on the 100 remaining faces.

[9]We build a nearest neighbor graph based on the pixels values.

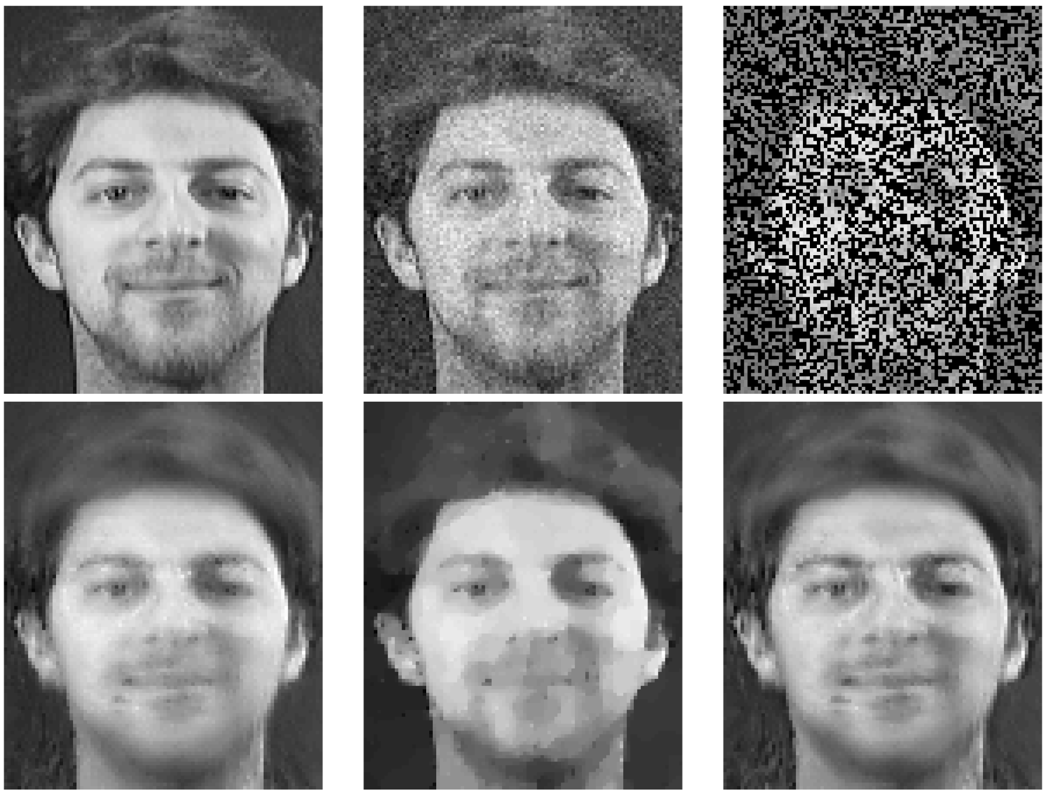


Figure 13: ORL dataset, single in-painting experiment. Top left: Original image. Top center: Noisy image (SNR 12.43 dB). Top right: Measurements 50% of the noisy image. Bottom left: Reconstruction using Tikhonov prior (SNR 12.12 dB). Bottom center: Reconstruction using classic TV prior (SNR 13.53 dB). Bottom right: Reconstruction using Wiener optimization (SNR 14.42 dB).

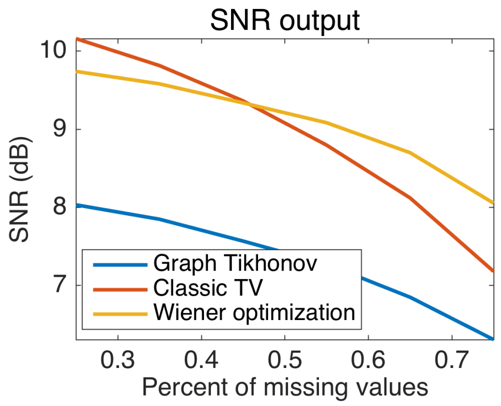


Figure 14: Inpainting experiment on ORL dataset. Left: some images of the dataset. Right: reconstruction error.

8 Conclusion

In this contribution, we have extended the common concept of stationarity to graph signals. Using this statistical model, we proposed a new regularization framework that leverages the stationarity hypothesis by using the Power Spectral Density (PSD) of the signal. Since the PSD can be efficiently estimated, even for large graphs, the proposed Wiener regularization framework offers a compelling way to solve traditional problems such as denoising, regression or semi-supervised learning. We believe that stationarity is a natural hypothesis for many signals on graphs and showed experimentally that it is deeply connected with the popular nearest neighbor graph construction. As future work, it would be very interesting to clarify this connection and explore if stationarity could be used to infer the graph structure from training signals, in the spirit of [38].

Acknowledgment

We thank the anonymous reviewers for their constructive comments as well as Andreas Loukas, Vassilis Kalofolias and Nauman Shahid for their suggestions.

This work has been supported by the Swiss National Science Foundation research project *Towards Signal Processing on Graphs*, grant number: 2000_21/154350/1.

A Convex models

Convex optimization has recently become a standard tool for problems such as de-noising, de-convolution or in-painting. Graph priors have been used in this field for more than a decade [6, 7, 8]. The general assumption is that the signal varies smoothly along the edges, which is equivalent to saying that the signal is low-frequency-based. Using this assumption, one way to express mathematically an in-painting problem is the following:

$$\bar{x} = \arg\min_x x^T Lx \qquad \text{s.t.} \qquad \|Mx - y\|_2 \leq \epsilon \tag{22}$$

where M is a masking operator and ϵ a constant computed thanks to the noise level. We could also rewrite the objective function as $x^T Lx + \gamma \|Mx - y\|_2^2$, but this implies a greedy search of the regularization parameter γ even when the level of noise is known. For our simulations, we use Gaussian i.i.d. noise of standard deviation n. It allows us to optimally set the regularization parameter $\epsilon = n\sqrt{\#y}$, where $\#y$ is the number of elements of the measurement vector.

Graph de-convolution can also be addressed with the same prior assumption leading to

$$\bar{x} = \arg\min_x x^T Lx \qquad \text{s.t.} \qquad \|h(L)x - y\|_2 \leq \epsilon \tag{23}$$

where h is the convolution kernel. To be as generic as possible, we combine problems (22) and (23) together leading to a model capable of performing de-convolution, in-painting and de-noising at the same time:

$$\bar{x} = \arg\min_x x^T Lx \qquad \text{s.t.} \qquad \|Mh(L)x - y\|_2 \leq \epsilon. \tag{24}$$

When the signal is piecewise smooth on the graph, another regularization term can be used instead of $x^T Lx = \|\nabla_{\mathcal{G}} x\|_2^2$, which is the ℓ_2-norm of the gradient on the graph[10]. Using the ℓ_1-norm of the gradient favors a small number of major changes in signal and thus is better for piecewise smooth signals. The resulting model is:

$$\bar{x} = \arg\min_x \|\nabla_{\mathcal{G}} x\|_1 \qquad \text{s.t.} \qquad \|Mh(L)x - y\|_2 \leq \epsilon \tag{25}$$

In order to solve these problems, we use a subset of convex optimization tools called proximal splitting methods. Since we are not going to summarize them here, we encourage a novice reader to consult [26, 27] and the references therein for an introduction to the field.

B Proof of Theorem 3

Proof. The proof is a classic development used in Bayesian machine learning. By assumption $\mathbf{x}$ is a sample of a Gaussian random multivariate signal $\mathbf{x} \sim \mathcal{N}\left(m_{\mathbf{x}}, s^2(L)\right)$. The measurements are given by

$$\mathbf{y} = H\mathbf{x} + \mathbf{w}_n,$$

where $\mathbf{w}_n \sim \mathcal{N}\left(0, \sigma^2\right)$ and thus have the following first and second moments: $\mathbf{y}|\mathbf{x} \sim \mathcal{N}\left(H\mathbf{x}, \sigma^2 I)\right)$. For simplicity, we assume $s^2(L)$ to be invertible. However this assumption is not necessary. We can write the probabilities of $\mathbf{x}$ and $\mathbf{y}|\mathbf{x}$ as:

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z_{H\mathbf{x}}} e^{-\|s^{-1}(L)(\mathbf{x}-m_{\mathbf{x}})\|_2^2} = \frac{1}{Z_{H\mathbf{x}}} e^{-\|s^{-1}(L)\tilde{\mathbf{x}}\|_2^2},$$

[10]The gradient on the graph is defined as $\nabla_{\mathcal{G}} x(e) = \frac{1}{2}\sqrt{W(i,j)}\,(x(i) - x(j))$, where e is the index corresponding the the edge linking the nodes i and j.

$$\mathbb{P}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{s\mathbf{x}}} e^{-\sigma^2 \|(H\mathbf{x}-\mathbf{y})\|_2^2}.$$

Using Bayes law we find

$$\mathbb{P}(\mathbf{x}|\mathbf{y}) = \frac{\mathbb{P}(\mathbf{y}|\mathbf{x})\mathbb{P}(\mathbf{x})}{\mathbb{P}(\mathbf{y})}.$$

The MAP estimator is

$$\begin{aligned}
\bar{\mathbf{x}}|\mathbf{y} &= \arg\max_{\mathbf{x}} \mathbb{P}(\mathbf{x}|\mathbf{y}) \\
&= \arg\max_{\mathbf{x}} \log\left(\mathbb{P}(\mathbf{x}|\mathbf{y})\right) \\
&= \arg\min_{\mathbf{x}} -\log\left(\mathbb{P}(\mathbf{y}|\mathbf{x})\right) - \log\left(\mathbb{P}(\mathbf{x})\right) + \log\left(\mathbb{P}(\mathbf{y})\right) \\
&= \arg\min_{\mathbf{x}} \|s^{-1}(L)\tilde{\mathbf{x}}\|_2^2 + \sigma^{-2}\|(H\mathbf{x}-\mathbf{y})\|_2^2 \\
&= \arg\min_{\mathbf{x}} \|w(L)\tilde{\mathbf{x}}\|_2^2 + \|(H\mathbf{x}-\mathbf{y})\|_2^2,
\end{aligned}$$

where $w(L) = \sigma s^{-1}(L)$. □

C Proof of Theorem 5

The following is a generalization of the classical proof. For simplicity, we assume that $m_{\tilde{\mathbf{x}}} = 0$, i.e $\tilde{\mathbf{x}} = \mathbf{x}$.

Proof. Because, by hypothesis $H = h(L) = Uh(\Lambda)U^*$, we can rewrite the optimization problem (16) in the graph Fourier domain using the Parseval identity $\|\mathbf{x}\|_2 = \|U\mathbf{x}\|_2 = \|\hat{\mathbf{x}}\|_2$:

$$\hat{\bar{\mathbf{x}}}|\hat{\mathbf{y}} = \arg\min_{\hat{\mathbf{x}}} \|w(\Lambda)\hat{\mathbf{x}}\|_2^2 + \|h(\Lambda)\hat{\mathbf{x}} - \hat{\mathbf{y}}\|_2^2.$$

Since the matrix Λ is diagonal, the solution of this problem satisfies for all graph eigenvalue λ_ℓ

$$w^2(\lambda_\ell)\bar{\mathbf{x}}[\ell] + h^2(\lambda_\ell)\bar{\mathbf{x}}[\ell] - h(\lambda_\ell)\hat{\mathbf{y}}[\ell] = 0. \tag{26}$$

For simplicity, we drop the notation (λ_ℓ) and $[\ell]$. The previous equation is transformed in

$$\bar{\mathbf{x}} = \frac{h}{w^2+h^2}\hat{\mathbf{y}}.$$

As a next step, we use the fact that $\hat{\mathbf{y}} = h\hat{\mathbf{x}} + \hat{\mathbf{w}}_n$ to find:

$$\hat{\bar{\mathbf{x}}} = \frac{h^2\hat{\mathbf{x}} + h\hat{\mathbf{w}}_n}{w^2+h^2}.$$

The error performed by the algorithm becomes

$$\hat{\mathbf{e}} = \hat{\bar{\mathbf{x}}} - \hat{\mathbf{x}} = \frac{-w^2\hat{\mathbf{x}}}{w^2+h^2} + \frac{h\hat{\mathbf{w}}_n}{w^2+h^2}.$$

The expectation of the error can thus be computed:

$$\begin{aligned}
\mathbb{E}\left(\hat{\mathbf{e}}^2\right) &= \frac{w^4\mathbb{E}\left(\hat{\mathbf{x}}^2\right)}{\left(w^2+h^2\right)^2} + \frac{h^2\mathbb{E}\left(\hat{\mathbf{w}}_n^2\right)}{\left(w^2+h^2\right)^2} - \frac{hw^2\mathbb{E}\left(\hat{\mathbf{x}}\hat{\mathbf{w}}_n\right)}{\left(w^2+h^2\right)^2} \\
&= \frac{w^4s^2 + h^2n}{\left(w^2+h^2\right)^2},
\end{aligned}$$

with s^2 the PSD of x_o and n the PSD of the noise w_n. Note that $\mathbb{E}\left(\hat{x}_o\hat{w}_n\right) = 0$ because x and w_n are uncorrelated. Let us now substitute w^2 by z and minimize the expected error (for each λ_ℓ) with respect to z:

$$\begin{aligned}
\frac{\partial}{\partial z}\mathbb{E}\left(\hat{\mathbf{e}}^2\right) &= \frac{\partial}{\partial z}\frac{z^2s^2 + h^2n}{\left(z+h^2\right)^2} \\
&= \frac{2zs^2\left(z+h^2\right) - 2\left(z^2s^2 + h^2n\right)}{\left(z+h^2\right)^3} = 0.
\end{aligned}$$

From the numerator, we get:

$$2zs^2h^2 - 2h^2n = 0$$

The three possible solution for z are $z_1 = \frac{n}{s^2}$, $z_2 = \infty$ and $z_3 = -\infty$. z_3 is not possible because z is required to be positive. z_2 leads to $\dot{x} = 0$ which is optimal only if $s^2 = 0$. The optimal solution is therefore $z(\lambda_\ell) = \frac{n(\lambda_\ell)}{s^2(\lambda_\ell)}$, resulting in

$$w(\lambda_\ell) = \sqrt{\frac{n(\lambda_\ell)}{s^2(\lambda_\ell)}}.$$

This finishes the first part of the proof. To show that the solution to (16) is a Wiener filtering operation, we replace $w^2(\lambda_\ell)$ by $\frac{n(\lambda_\ell)}{s^2(\lambda_\ell)}$ in (26) and find

$$\hat{\mathbf{x}}[\ell] = \frac{s^2(\lambda_\ell)h(\lambda_\ell)}{h^2(\lambda_\ell)s^2(\lambda_\ell) + n(\lambda_\ell)}\hat{\mathbf{y}}[\ell],$$

which is the Wiener filter associated to the convolution $h(L) = H$. □

D Proof of Theorem 4

Proof. Let $\mathbf{x}$ be GWSS with covariance matrix $\Sigma_{\mathbf{x}} = s^2(L)$ and mean $m_{\mathbf{x}}$. The measurements satisfy

$$\mathbf{y} = H\mathbf{x} + \mathbf{w}_n,$$

where $\mathbf{w}_n$ is i.i.d noise with PSD σ^2. The variable $\mathbf{y}$ has a covariance matrix $\Sigma_{\mathbf{y}} = Hs^2(L)H^* + \sigma^2 I$ and a mean $m_{\mathbf{y}} = Hm_{\mathbf{x}}$. The covariance between $\mathbf{x}$ and $\mathbf{y}$ is $\Sigma_{\mathbf{xy}} = \Sigma_{\mathbf{yx}}^* = s^2(L)H^*$. For simplicity, we assume $s^2(L)$ and $Hs^2(L)H^* + \sigma^2 I$ to be invertible, however this assumption is not necessary. The Wiener optimization framework reads:

$$\bar{\mathbf{x}} = \arg\min_{\mathbf{x}} \|H\mathbf{x} - \mathbf{y}\|_2^2 + \sigma^2\|s^{-1}(L)(\mathbf{x} - m_{\mathbf{x}})\|_2^2.$$

We perform the following change of variable $\tilde{\mathbf{x}} = \mathbf{x} - m_{\mathbf{x}}$, $\tilde{\mathbf{y}} = \mathbf{y} - m_{\mathbf{x}}$ and we obtain:

$$\bar{\tilde{\mathbf{x}}} = \arg\min_{\tilde{\mathbf{x}}} \|H\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|_2^2 + \sigma^2\|s^{-1}(L)\tilde{\mathbf{x}})\|_2^2.$$

The solution of the problem satisfies

$$H^*H\bar{\tilde{\mathbf{x}}} - H^*\tilde{\mathbf{y}} + \sigma^2 s^{-2}(L)\bar{\tilde{\mathbf{x}}} = 0.$$

From this equation we get $\bar{\tilde{\mathbf{x}}}$ and transform it as:

$$\begin{aligned}
\bar{\tilde{\mathbf{x}}} &= \left(H^*H + \sigma^2 s^{-2}(L)\right)^{-1} H^*\tilde{\mathbf{y}} \\
&= s(L)\left(\sigma^2 I + s(L)H^*Hs(L)\right)^{-1} s(L)H^*\tilde{\mathbf{y}} \\
&= \left(\tfrac{1}{\sigma^2}s^2(L)H^* - \tfrac{1}{\sigma^2}s^2(L)H^*\left(\sigma^2 I + H^*s^2(L)H\right)^{-1}Hs^2(L)H^*\right)\tilde{\mathbf{y}} \qquad (27) \\
&= \frac{1}{\sigma^2}s^2(L)H^*\left(I - \left(\sigma^2 I + H^*s^2(L)H\right)^{-1}Hs^2(L)H^*\right)\tilde{\mathbf{y}} \\
&= s^2(L)H^*\left(\sigma^2 I + H^*s^2(L)H\right)^{-1}\tilde{\mathbf{y}} \\
&= \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1}\tilde{\mathbf{y}}
\end{aligned}$$

where (27) follows from the Woodbury, Sherman and Morrison formula. The linear estimator of $\mathbf{x}$ corresponding to Wiener optimization is thus:

$$\begin{aligned}
\bar{\mathbf{x}} &= \bar{\mathbf{x}} + m_{\mathbf{x}} \\
&= \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1}(\mathbf{y} - Hm_{\mathbf{x}}) + m_{\mathbf{x}} \\
&= \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1}\mathbf{y} + \left(I - \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1}H\right)m_{\mathbf{x}} \\
&= Q\mathbf{y} + (I - QH)\,m_{\mathbf{x}}
\end{aligned}$$

We observe that it is equivalent to the solution of the linear minimum mean square error estimator:

$$\arg\min_{Q,b} \mathbb{E}\left(\|Q\mathbf{y} + b - \bar{x}\|\right)^2$$

with $\mathbf{y} = H\mathbf{x} + \mathbf{w}_n$. See [39, Equation 12.6]. □

Using similar arguments, we can prove that

$$\bar{\mathbf{x}} = \arg\min_x \|g^{-1}(L)(\mathbf{x} - m_{\mathbf{x}}\|_2^2 \quad \text{s.t. } y = Xx$$

leads to

$$\bar{\mathbf{x}} = s^2(L)\left(Hs^2(L)H^*\right)^{-1}\mathbf{y} + \left(I - s^2(L)\left(Hs^2(L)H^*\right)^{-1}H\right)m_{\mathbf{x}}$$

and is thus a linear minimum mean square estimator too.

E Developement of equation 21

Proof. Let us denote the matrix of squared distances $D[i,j] = \frac{1}{K}\sum_k |x_k[i] - x_k[j]|^2$ for the samples $\{x_1, x_2, \dots x_K\}$ of the random multivariate variable $\mathbf{x}$ on a N vertices graph. Let us assume further that $m[k] = \sum_{n=1}^N x_k[n] = 0$. We show then that $\bar{\Sigma}_{\mathbf{x}} = -\frac{1}{2}JD_{\mathbf{x}}J$ where $\bar{\Sigma}_{\mathbf{x}}$ is the covariance (Gram) matrix defined as $\bar{\Sigma}_{\mathbf{x}}[i,j] = \frac{1}{K}\sum_{k=1}^K x_k[i]x_k[j]$ and J is centering matrix $J[k,l] = \delta_k[l] - \frac{1}{N}$.

We have

$$(JDJ)[i,j] = D[i,j] + N^{-2}\sum_{k,l=1}^N D[k,l] - N^{-1}\sum_{k=1}^N (D[i,k] + D[k,j])$$

Let us substitute $D[i,j] = \bar{\Sigma}_{\mathbf{x}}[i,i] + \bar{\Sigma}_{\mathbf{x}}[j,j] - 2\bar{\Sigma}_{\mathbf{x}}[i,j]$, then we find

$$\begin{aligned} & (JDJ)[i,j] \\ = \; & \bar{\Sigma}_{\mathbf{x}}[i,i] + \bar{\Sigma}_{\mathbf{x}}[j,j] - 2\bar{\Sigma}_{\mathbf{x}}[i,j] + N^{-2}\left(2N\sum_{n=1}^N \bar{\Sigma}_{\mathbf{x}}[n,n] - 2m^*m\right) \\ - \; & N^{-1}\left(N\bar{\Sigma}_{\mathbf{x}}[i,i] + N\bar{\Sigma}_{\mathbf{x}}[j,j] + 2\sum_{n=1}^N \bar{\Sigma}_{\mathbf{x}}[n,n] - 2m^*(x[j] + x[i])\right) \\ = \; & -2\bar{\Sigma}_{\mathbf{x}}[i,j] - 2N^{-2}m^*m + 2N^{-1}m^*(x[j] + x[i]). \end{aligned}$$

Under the assumption $m[k] = \sum_{n=1}^N x_k[n] = 0$, we recover the desired result $\bar{\Sigma}_{\mathbf{x}} = -\frac{1}{2}JD_{\mathbf{x}}J$. □

References

[1] C. Williams, “Prediction with Gaussian processes: From linear regression to linear prediction and beyond,” *Learning in graphical models*, 1998.

[2] B. Girault, “Signal processing on graphs-contributions to an emerging field,” Ph.D. dissertation, Ecole normale supérieure de lyon-ENS LYON, 2015.

[3] P. Welch, “The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms,” *IEEE Transactions on audio and electroacoustics*, pp. 70–73, 1967.

[4] M. S. Bartlett, “Periodogram analysis and continuous spectra,” *Biometrika*, pp. 1–16, 1950.

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

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

[7] D. Zhou and B. Schölkopf, “A regularization framework for learning from graph data,” 2004.

[8] G. Peyré, S. Bougleux, and L. Cohen, “Non-local regularization of inverse problems,” in *Computer Vision–ECCV 2008.* Springer, 2008, pp. 57–68.

[9] A. J. Smola and R. Kondor, “Kernels and Regularization on Graphs,” in *Proc. Ann. Conf. Comp. Learn. Theory*, B. Schölkopf and M. Warmuth, Eds. Springer, 2003, pp. 144–158.

[10] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[11] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, 2016.

[12] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, pp. 1644–1656, 2013.

[13] A. Gadde and A. Ortega, "A probabilistic interpretation of sampling theory of graph signals," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on.* IEEE, 2015, pp. 3257–3261.

[14] C. Zhang, D. Florêncio, and P. A. Chou, "Graph signal processing–a probabilistic framework," 2015.

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

[16] B. Girault, P. Goncalves, E. Fleury, and A. S. Mor, "Semi-supervised learning for graph to signal mapping: A graph signal wiener filter interpretation," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on.* IEEE, 2014, pp. 1115–1119.

[17] F. R. Chung, *Spectral graph theory.* AMS Bookstore, 1997, vol. 92.

[18] F. Chung, "Laplacians and the cheeger inequality for directed graphs," *Annals of Combinatorics*, vol. 9, no. 1, pp. 1–19, 2005.

[19] G. Strang, "The discrete cosine transform," *SIAM review*, vol. 41, no. 1, pp. 135–147, 1999.

[20] A. Susnjara, N. Perraudin, D. Kressner, and P. Vandergheynst, "Accelerated filtering on graphs using lanczos method," *arXiv preprint arXiv:1509.04537*, 2015.

[21] N. Perraudin, B. Ricaud, D. Shuman, and P. Vandergheynst, "Global and local uncertainty principles for signals on graphs," *arXiv preprint arXiv:1603.03030*, 2016.

[22] D. I. Shuman, C. Wiesmeyr, N. Holighaus, and P. Vandergheynst, "Spectrum-adapted tight graph wavelet and vertex-frequency frames," *Signal Processing, IEEE Transactions on*, vol. 63, no. 16, pp. 4223–4235, 2015.

[23] N. Wiener, "Generalized harmonic analysis," *Acta mathematica*, vol. 55, no. 1, pp. 117–258, 1930.

[24] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes.* Tata McGraw-Hill Education, 2002.

[25] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series.* MIT press Cambridge, MA, 1949, vol. 2.

[26] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering.* Springer, 2011, pp. 185–212.

[27] N. Komodakis and J.-C. Pesquet, "Playing with duality: An overview of recent primal? dual approaches for solving large-scale optimization problems," *Signal Processing Magazine, IEEE*, vol. 32, no. 6, pp. 31–54, 2015.

[28] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[29] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[30] B. Julesz, "Visual Pattern Discrimination," *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 84–92, 1962.

[31] N. L. Roux, Y. Bengio, P. Lamblin, M. Joliveau, and B. Kégl, "Learning the 2-d topology of images," in *Advances in Neural Information Processing Systems*, 2008, pp. 841–848.

[32] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 3, pp. 328–340, 2005.

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

[34] N. Perraudin, J. Paratte, D. Shuman, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, Aug. 2014.

[35] N. Perraudin, D. Shuman, G. Puy, and P. Vandergheynst, "UNLocBoX A matlab convex optimization toolbox using proximal splitting methods," *ArXiv e-prints*, Feb. 2014.

[36] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.

[37] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical imaging and vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[38] V. Kalofolias, "How to learn a graph from smooth signals," in *Journal of Machine Learning Research (JMLR)*, no. EPFL-CONF-214936, 2016.

[39] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory.* Prentice-Hall, Inc., 1993.