



**HAL**  
open science

# Joint Graph and Vertex Importance Learning

Benjamin Girault, Eduardo Pavez, Antonio Ortega

► **To cite this version:**

Benjamin Girault, Eduardo Pavez, Antonio Ortega. Joint Graph and Vertex Importance Learning. 2023 31st European Signal Processing Conference (EUSIPCO), Sep 2023, Helsinki, Finland. pp.1858-1862, 10.23919/EUSIPCO58844.2023.10290030 . hal-04025968v2

**HAL Id: hal-04025968**

**<https://hal.science/hal-04025968v2>**

Submitted on 8 Jun 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

# JOINT GRAPH AND VERTEX IMPORTANCE LEARNING

Benjamin Girault<sup>\*</sup>, Eduardo Pavez<sup>†</sup>, Antonio Ortega<sup>†</sup>

<sup>\*</sup> Université de Rennes, ENSAI, CNRS, CREST-UMR 9194, Rennes, France

<sup>†</sup> Signal and Image Processing Institute, University of Southern California, Los Angeles, USA

## ABSTRACT

In this paper, we explore the topic of graph learning from the perspective of the Irregularity-Aware Graph Fourier Transform, with the goal of learning the graph signal space inner product to better model data. We propose a novel method to learn a graph with smaller edge weight upper bounds compared to combinatorial Laplacian approaches. Experimentally, our approach yields much sparser graphs compared to a combinatorial Laplacian approach, with a more interpretable model.

**Index Terms** — Graph signal processing, graph learning, graph signal Hilbert space

## 1. INTRODUCTION

The field of graph signal processing proposes a toolbox to analyze, process and transform data supported by arbitrary discrete structures [1, 2]. Examples include measurements collected by sensor networks, activities in a human neuronal network, or image processing [1]. However, for some applications, we have access such data, but the graph structure is either missing, noisy, or incomplete. In this case, it is important to learn a graph that provides a model that can be leveraged by the graph signal processing toolbox to process or analyze the data. To that end, graph learning is typically formulated as an optimization problem, where the goal is to obtain an algebraic representation of the graph that best captures the variability of the data. Popular objective functions are built on graph signal smoothness or stationarity [3, 4].

Following [5], graph learning can be formulated as an inverse covariance estimation problem with graph Laplacian constraints. In this context, the most common type of Laplacian is the *Combinatorial Graph Laplacian* (CGL). The CGL variation  $\Delta(\mathbf{x}) = \mathbf{x}^* \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{ij} w_{ij} |x_i - x_j|^2$  exhibits the desirable property of constant signals having zero variation, since only variations along edges are used [1]. In addition, the *Graph Fourier Transform* (GFT) decomposes a signal on an orthonormal basis that minimizes this variation [6]. Learning a CGL leads then to a consistent spectral interpretation given by the *graph Power Spectrum Density* (gPSD) [7]. Ultimately, learning a CGL and having access to such a gPSD allows for specific filter designs such as Wiener filters [8] or ARMA filters [9]. To account for the difficulty associated with singular CGL matrices in inverse covariance esti-

mation, the objective function is oftentimes modified [5, 10–13]. However, such an approach produces dense graphs, even if variables are weakly correlated (see Sec. 4 and [12]) because the modified objective function encourages well connected graphs [10]. This issue can be solved by incorporating non-convex sparse regularization [12, 14] at the expense of a more complex graph learning algorithm.

This strict CGL approach can be relaxed by allowing for the estimated inverse covariance to be a CGL plus a diagonal matrix [5]. The resulting matrices are the *Generalized Graph Laplacian* (GGL) and the *Diagonally Dominant Graph Laplacian* (DDGL). Many graph learning algorithms for GGL matrices have been proposed [5, 15–17], which have been shown not to produce the spurious connections that often arise in CGL approaches [18]. However, the graph variation loses its classical interpretations: due to the additional term weighting each vertex magnitude and summing over all vertices, constant signal exhibit non-zero variations.

In this work we propose a new formulation that allows us to learn CGLs without spurious connections while preserving interpretability, in contrast with the relaxed Laplacians above for the DDGL case. Our approach to the inverse covariance problem uses our recently introduced generalization of the *Graph Fourier Transform* (GFT) to arbitrary Hilbert spaces of graph signals: the *Irregularity Aware Graph Fourier Transform* [6]. In this generalization, the space of graph signals is equipped with an inner product other than the standard dot product, thus adapting the notion of *orthogonality* and *norm* of these graph signals to the application. This additional parameter to the GFT shows great promise in areas such as vertex sampling [19], image [20] or point cloud processing [21], and filter design [22]. In the context of graph learning, we propose to *jointly learn a CGL and an inner product*, which corresponds to a diagonal matrix whose diagonal terms reflect the relative importances of the vertices [6].

Our contributions are threefold: i) we formulate a joint CGL and inner product learning problem and show that it can be reduced to learning a DDGL (Sec. 3.1), allowing for the CGL and the vertex importance weights to be learned using any DDGL algorithm [5, 17], ii) we propose an efficient and scalable coordinate minimization algorithm similar to [10] for the CGL that updates one edge weight or vertex importance per iteration (Sec. 3.2), and iii) we prove that the proposed CGL and inner product solution is sparse and obtain a sharp upper bound for the non zero edge

weights (Sec. 3.3).

Our experiments with sampled intrinsic stationary continuous signals highlight the key benefits of our approach, including spatial consistency and a substantial increase of sparsity compared to learning a CGL (Sec. 4).

## 2. BACKGROUND

### 2.1. Graph Signal Processing

A graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$  is defined by a set of vertices  $\mathcal{V}$ , a set of edges  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  and an edge weight function  $w : \mathcal{E} \rightarrow \mathbb{R}_+$ . We denote by  $N$  (*resp.*  $M$ ) the number of vertices (*resp.* edges). In this paper we are interested in undirected graphs where for any edge  $e = (i, j) \in \mathcal{E}$ , its opposite  $(j, i)$  is also an edge in  $\mathcal{E}$  with identical weight  $w(i, j) = w(j, i)$ .

Algebraic representations of such graphs include the classical adjacency or weight matrix  $\mathbf{A}$  such that  $\mathbf{A}_{ij} = w(i, j)$  if  $(i, j) \in \mathcal{E}$ , 0 otherwise. The degree of a vertex is defined as the sum  $d_i = \sum_{ij \in \mathcal{E}} w(i, j)$  of its incident edges, and these degrees are collected into the diagonal degree matrix  $\mathbf{D} = \text{diag}(d_1, \dots, d_N)$ . This allows to define the combinatorial Laplacian matrix  $\mathbf{L} = \mathbf{D} - \mathbf{A}$  of the graph.

Using the unweighted incidence matrix  $\mathbf{B} = [\mathbf{b}_1 \cdots \mathbf{b}_M]$  such that for any edge  $e = (i, j) \in \mathcal{E}$ , such that  $i < j$ , then  $\mathbf{B}_{i,e} = 1$  and  $\mathbf{B}_{j,e} = -1$ , and  $\mathbf{B}$  is zero elsewhere, we have  $\mathbf{L} = \mathbf{B}\mathbf{W}\mathbf{B}^T$ , where  $\mathbf{W} = \text{diag}(w_1, \dots, w_M)$  is the diagonal matrix collecting edge weights.

A graph signal  $x : \mathcal{V} \rightarrow \mathbb{C}$  is a function mapping vertices to scalar values. We assumed an indexing of the vertices with integers  $\{1, \dots, N\}$  (and of the edges with  $\{1, \dots, M\}$ ) to define the algebraic representations above. A graph signal  $\mathbf{x}$  is then represented by a column vector  $[\mathbf{x}_1, \dots, \mathbf{x}_N]^T \in \mathbb{C}^N$ .

### 2.2. Irregularity Aware Graph Fourier Transform

The Irregularity Aware Graph Fourier Transform is a parametric generalisation of the orthonormal Graph Fourier Transforms (GFT) using two parameters [6]: the graph variation operator  $\Delta : \mathbb{C}^N \rightarrow \mathbb{R}_+$  mapping any graph signal  $\mathbf{x}$  to its non-negative variation  $\Delta(\mathbf{x}) \geq 0$ , and an inner product  $\langle \cdot, \cdot \rangle_{\mathbf{Q}} : \mathbb{C}^N \times \mathbb{C}^N \rightarrow \mathbb{R}$  with Hermitian positive definite matrix  $\mathbf{Q}$ , such that  $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{Q}} = \mathbf{y}^* \mathbf{Q} \mathbf{x}$ . We denote it as the  $(\Delta, \mathbf{Q})$ -GFT. With  $\Delta(\mathbf{x}) = \mathbf{x}^* \mathbf{M} \mathbf{x}$  and  $\mathbf{M}$  a Hermitian semi-definite positive matrix, its graph Fourier modes  $\{\mathbf{u}_l\}_l$  verify  $\mathbf{M} \mathbf{u}_l = \lambda_l \mathbf{Q} \mathbf{u}_l$ . The graph Fourier basis  $\{\mathbf{u}_l\}_l$  is then orthonormal with respect to the  $\mathbf{Q}$ -inner product. We collect all the eigenvalues in the diagonal matrix  $\mathbf{\Lambda} = \text{diag}(\lambda_0, \dots, \lambda_{N-1})$ , and the graph Fourier modes in  $\mathbf{U} = [\mathbf{u}_0 \dots \mathbf{u}_{N-1}]$  (such that  $\mathbf{U}^* \mathbf{Q} \mathbf{U} = \mathbf{I}$ ). We obtain the inverse and forward  $(\Delta, \mathbf{Q})$ -GFT of a graph signal  $\mathbf{x}$  with  $\mathbf{x} = \mathbf{F}^{-1} \hat{\mathbf{x}} = \mathbf{U} \hat{\mathbf{x}}$  and  $\hat{\mathbf{x}} = \mathbf{F} \mathbf{x} = \mathbf{U}^* \mathbf{Q} \mathbf{x}$ .

In this paper, we use the combinatorial Laplacian variation  $\Delta(\mathbf{x}) = \mathbf{x}^* \mathbf{L} \mathbf{x}$  ( $\mathbf{M} = \mathbf{L}$ ), and further assume that the inner product matrix is diagonal  $\mathbf{Q} = \text{diag}(q_1, \dots, q_N)$ . This effectively adds vertex *importances* to the model alongside edge

weights. Non-diagonal inner product matrices  $\mathbf{Q}$  in the context of graph learning will be studied in a future communication.

### 2.3. Graph Wide Sense Stationarity (gWSS)

Our goal is to obtain a stochastic model for the data at hand. To that end, we use as class of models the framework of Graph Wide Sense Stationarity (gWSS) we previously introduced and its spectral characterization [7]: A stochastic graph signal is *gWSS* if and only if its mean is a DC component and its spectral components are uncorrelated. We denote by  $\mathbf{\Sigma}$  the covariance matrix of the graph signal,  $\mathbf{\Gamma}$  its spectral covariance matrix, and  $\boldsymbol{\gamma}$  its diagonal, called the graph Power Spectrum Density (gPSD) [7]. In other words,  $\gamma_l$  is the variance of the  $l^{\text{th}}$  spectral component  $\hat{\mathbf{x}}_l$  of the signal  $\mathbf{x}$ .

### 2.4. Graph Learning through Coordinate Minimization

Assume that the graph signals are realizations of a zero-mean Gaussian gWSS with gPSD  $\gamma(0) = 0$  and  $\gamma(\lambda) = 1/\lambda$  when  $\lambda > 0$ . Using the  $(\mathbf{L}, \mathbf{I})$ -GFT, this translates into the covariance matrix  $\mathbf{\Sigma} = \mathbf{L}^\dagger$ . Under this setting we obtain the maximum likelihood estimator proposed by [5] (see Sec. 1), which minimizes the following cost function:

$$F(\mathbf{L}) = -\log \det(\mathbf{L} + 1/N \mathbf{J}) + \text{tr}(\mathbf{L} \mathbf{S}), \quad (1)$$

where  $\mathbf{J}$  is the all-one matrix (necessary, since a valid  $\mathbf{L}$  is always singular), and  $\mathbf{S} = \frac{1}{K} \sum_k \mathbf{x}^{(k)} \mathbf{x}^{(k)*}$  is the empirical covariance matrix. The coordinate minimization approach of [10] minimizes  $F$  iteratively for all edge weights. The optimal update (fixing all other weights)  $\delta_e$  to edge weight  $w_e$  is:

$$\delta_e = \max(-w_e, 1/h_e - 1/r_e), \quad (2)$$

with the edge cost  $h_e = \mathbf{b}_e^T \mathbf{S} \mathbf{b}_e$ , and the effective resistance  $r_e = \mathbf{b}_e^T (\mathbf{L} + 1/N \mathbf{J})^{-1} \mathbf{b}_e$  for an edge  $e$ . This graph learning algorithm updates iteratively all edge weights using (2), and stops when the update to  $F$  after all the edge weight updates is below a stopping threshold. The optimal graph weights are upper bounded by  $w_e \leq 1/h_e$  [10].

## 3. PROPOSED GRAPH LEARNING APPROACH

### 3.1. Data Model

We assume that the data we have is i.i.d. multivariate Gaussian. Our goal is to model the data with a graph given by its combinatorial graph Laplacian  $\mathbf{L}$  and the graph signal inner product matrix  $\mathbf{Q} = \text{diag}(\mathbf{q})$ . To that end, we learn this graph such that the data is gWSS on the graph, with gPSD  $\gamma(\lambda) = (1 + \lambda)^{-1}$  when using the  $(\mathbf{L}, \mathbf{Q})$ -GFT. Such a gPSD allows for a continuous gPSD around frequency 0 compared to the classical  $\gamma(\lambda) = 1/\lambda$ .

**Theorem 1.** The covariance matrix of a stochastic graph signal with gPSD  $\gamma(\lambda) = (1 + \lambda)^{-1}$  using the  $(\mathbf{L}, \mathbf{Q})$ -GFT is:

$$\mathbf{\Sigma} = [\mathbf{Q} + \mathbf{L}]^{-1}. \quad (3)$$

*Proof.* We first observe that the spectral covariance matrix verifies  $\Gamma = [\mathbf{I} + \mathbf{A}]^{-1}$ . Using the  $(\mathbf{L}, \mathbf{Q})$ -GFT, we obtain  $\Sigma = \mathbb{E}[\mathbf{x}\mathbf{x}^*] = \mathbf{F}^{-1}\Gamma\mathbf{F}\mathbf{Q}^{-1} = [\mathbf{I} + \mathbf{Q}^{-1}\mathbf{L}]^{-1}\mathbf{Q}^{-1} = [\mathbf{Q} + \mathbf{L}]^{-1}$ . Given that  $\mathbf{Q}$  is Hermitian positive definite and  $\mathbf{L}$  is Hermitian positive semi-definite, then  $\mathbf{Q} + \mathbf{L}$  is definite, hence invertible and  $\Sigma$  is well-defined.  $\square$

Using maximum likelihood, combining the Gaussian assumption and Thm. 1 leads to the following convex [5] cost function to minimize:

$$F(\mathbf{L}, \mathbf{Q}) = -\log\det(\mathbf{Q} + \mathbf{L}) + \text{tr}((\mathbf{Q} + \mathbf{L})\mathbf{S}), \quad (4)$$

with  $\mathbf{S}$  the empirical covariance matrix of the data. Interestingly, the matrix  $\mathbf{Q} + \mathbf{L}$  in (4) also corresponds to the *Diagonally Dominant Graph Laplacian* proposed in [5], but with strict dominance of the diagonal. However, our spectral interpretation to this model using the  $(\mathbf{L}, \mathbf{Q})$ -GFT is different.

Note also that the cost function (4) is actually a generalization of [23] for any positive definite matrix  $\mathbf{Q}$  instead of only scaled versions of the identity matrix  $\frac{1}{\sigma^2}\mathbf{I}$ . The additional  $\ell_1$  penalty term of [23] and the difference between  $\mathbf{Q}$  and  $\frac{1}{\sigma^2}\mathbf{I}$  will be studied in a future communication.

### 3.2. Proposed Coordinate Minimization Approach

We propose to solve the following graph learning problem:

$$\min_{\mathbf{w} \geq \mathbf{0}, \mathbf{q} > \mathbf{0}} F(\mathbf{B} \text{diag}(\mathbf{w})\mathbf{B}^\top, \text{diag}(\mathbf{q})). \quad (5)$$

Note that the non-negativity of  $\mathbf{w}$  and positivity of  $\mathbf{q}$  are enough to ensure that  $\mathbf{L}$  is Hermitian semi-definite positive and  $\mathbf{Q}$  is Hermitian definite positive, such that the  $(\mathbf{L}, \mathbf{Q})$ -GFT is well-defined. However, for our implementation, we need to introduce a hyperparameter  $q_{\min}$  and change the positivity constraint to  $\forall i, q_i \geq q_{\min}$ . This is justified below, in the update formula for  $q_i$ .

Coordinate minimization iterates through all edges and vertices and updates these weights according to Thm. 2.

**Theorem 2** (Coordinate Minimization Update). (5) is solved by, at iteration  $t$ , updating either  $w_e$  or  $q_i$  using:

$$\begin{aligned} \delta_e^{(t)} &= \max(-w_e^{(t)}, \frac{1}{r_e} - \frac{1}{r_e^{(t)}}) \\ \delta_i^{(t)} &= \max(q_{\min} - q_i^{(t)}, \frac{1}{p_i} - \frac{1}{u_i^{(t)}}) \end{aligned}$$

with edge cost  $h_e = \mathbf{b}_e^\top \mathbf{S} \mathbf{b}_e$ , effective resistance  $r_e^{(t)} = \mathbf{b}_e^\top (\mathbf{Q}^{(t)} + \mathbf{L}^{(t)})^{-1} \mathbf{b}_e$ , vertex cost  $p_i = \mathbf{S}_{ii}$ , vertex effective importance  $u_i^{(t)} = [(\mathbf{Q}^{(t)} + \mathbf{L}^{(t)})^{-1}]_{ii}$ , and  $q_{\min} > 0$ .

*Proof.* Notice first how the trace term in (4) can be decomposed as a sum  $\text{tr}(\mathbf{L}\mathbf{S}) + \text{tr}(\mathbf{Q}\mathbf{S})$ . Any update to  $w_e$  (resp.  $q_i$ ) will only modify the first (resp. second) trace in this sum.

The derivation for edge weight update  $\delta_e$  is identical to [10] by changing  $\frac{1}{N}\mathbf{J}$  to  $\mathbf{Q}$  in the effective resistance  $r_e$  definition.

For the update  $\delta_i$  to vertex importance  $q_i$ , we solve:

$$\Delta F = -\log(1 + \delta_i [(\mathbf{Q} + \mathbf{L})^{-1}]_{ii}) + \delta_i \mathbf{S}_{ii},$$

where the first term is derived from the log-determinant variation and the second from the trace variation. Let  $p_i = \mathbf{S}_{ii}$ , and  $u_i = [(\mathbf{L} + \mathbf{Q})^{-1}]_{ii}$ . Using the Lagrangian of the problem minimizing the variation of  $F$  requires weak inequalities for the constraints. Therefore, we relax  $q_i > 0$  into  $q_i \geq q_{\min}$  for some  $q_{\min} > 0$ . The KKT conditions lead to:

$$\frac{-u_i}{1 + \delta_i u_i} + p_i - \lambda_i = 0, \lambda_i \geq 0, q_i^{(t+1)'} \geq 0, \lambda_i q_i^{(t+1)'} = 0,$$

with  $q_i^{(t+1)'} = q_i^{(t+1)} - q_{\min}$ . These conditions are satisfied in the proposed update.  $\square$

$u_i$  and  $r_e$  depend on  $\mathbf{L}$  and  $\mathbf{Q}$  and need to be updated after each update. We use the Sherman-Morrison formula to write the update to  $\Phi^{(t)} = (\mathbf{Q} + \mathbf{L})^{-1}$  [10]:

$$\Delta \Phi^{(t)} = \begin{cases} \frac{\delta_e^{(t)} (\Phi^{(t)} \mathbf{b}_e) (\Phi^{(t)} \mathbf{b}_e)^\top}{1 + \delta_e^{(t)} \mathbf{b}_e^\top \Phi^{(t)} \mathbf{b}_e} & \text{if } w_e \text{ is updated} \\ \frac{\delta_i^{(t)} [\Phi^{(t)}]_{ii} [\Phi^{(t)}]_{ii}^\top}{1 + \delta_i^{(t)} [\Phi^{(t)}]_{ii}} & \text{if } q_i \text{ is updated.} \end{cases}$$

Using  $\Delta \Phi^{(t)} = \Phi^{(t+1)} - \Phi^{(t)}$ , the updates to  $r_f$  ( $f \in \mathcal{E}$ ), and to  $q_j$  ( $j \in \mathcal{V}$ ) can then be easily obtained through:

$$\Delta r_f^{(t)} = \mathbf{b}_f^\top \Delta \Phi^{(t)} \mathbf{b}_f \quad \Delta u_j^{(t)} = [\Delta \Phi^{(t)}]_{jj}$$

which is efficient to implement when  $\Phi^{(t)} = (\mathbf{Q} + \mathbf{L})^{-1}$  is kept in memory and updated after each iteration.

**Connectedness** Compared to [10], edge weight updates can disconnect the graph since the matrix  $\mathbf{Q} + \mathbf{L}$  is always non-singular, whereas  $\mathbf{L} + \frac{1}{N}\mathbf{J}$  would become singular. As a consequence, edge weight updates are not restricted, thus allowing to fully adapt to the data.

**Stopping Criterion** Coordinate minimization is stopped whenever the maximum number of epochs ( $N + M$  updates) is reached, or improvement of the cost function  $F$  is below a predefined threshold after an epoch.

### 3.3. Optimal Graph Weight Properties

Similarly to the Generalized Graph Laplacian case of [11], we can upper bound edge weights, but only between vertices whose importances are larger than  $q_{\min}$ .

**Theorem 3.** Let  $\rho_{ij} = \mathbf{S}_{ij} / \sqrt{\mathbf{S}_{ii}\mathbf{S}_{jj}}$  be the sample correlation coefficient between vertices  $i$  and  $j$ . For any non zero graph weight  $w_{ij} > 0$  with  $q_i > q_{\min}$  and  $q_j > q_{\min}$ , the optimal solution of problem (5) verifies:

$$w_{ij} \leq \frac{1}{|\mathbf{S}_{ij}|} \frac{\rho_{ij}^2}{1 - \rho_{ij}^2} \quad (6)$$

*Proof.* Similar to [11, Appendix A], but using the additional constraints  $q_{\min} - [\Theta]_{ii} \leq 0$ .  $\square$

In addition, any negative correlation between two vertices leads to no edge between them, thus allowing us to remove those edges before performing coordinate minimization:

**Theorem 4.** The optimal graph satisfies:

$$\mathcal{E} = \{(i, j) \in \mathcal{V}^2 : w_{ij} > 0\} \subset \{(i, j) : \mathbf{S}_{ij} > 0\}. \quad (7)$$

*Proof.* Using the fact that  $\Theta$  is a generalized Laplacian (M-matrix) and the KKT conditions for the strict inequality.  $\square$

#### 4. EXPERIMENTS

To experimentally validate our method, we use a synthetic stochastic signal whose statistics are defined by an underlying Euclidean space. Our goal is to show that our approach can reliably extract meaningful properties of the Euclidean space from the learnt graph, without relying on where the observations are made in Euclidean space.

More precisely, we consider intrinsic stationary 2D signals [24]. In this experiment, we use an isotropic exponential variogram with no *nugget* (i.e. no measurement noise), a *sill* (variance) of 10 for all data points, and varying values of the *range*  $r$ :

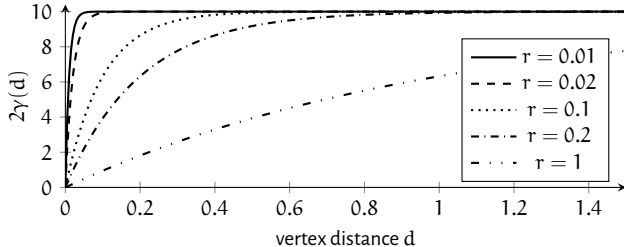
$$2\gamma(d) = 10(1 - \exp(-d/r)),$$

where  $d$  is the Euclidean distance between any two points. To observe the behavior of our proposed method, we choose 5 values of range (see Fig. 1). Smaller ranges correspond to signals being correlated only for a short distance (almost white noise), while larger ranges being highly correlated throughout the space (almost constant signal). A range of 0.1 corresponds to the interesting middle case where there is substantial correlation between somewhat close locations, but almost no correlation between locations far away.

We then uniformly sample the 2D space ( $[0, 1] \times [0, 1]$  Euclidean plane) at  $N = 50$  locations. To study the consistency of the learnt graph, we generate  $K = 50$  sets of random node locations. For a given spatial sampling and range  $r$ , we obtain the exact covariance matrix  $\mathbf{S}$  of the data with  $\mathbf{S}_{ii} = 10$  and  $\mathbf{S}_{ij} = 10 \exp(-d_{ij}/r), \forall i \neq j$ . Using Thm. 3, we obtain the following edge weight upper bound (when  $q_i > q_{\min} < q_j$ ):

$$w_{ij} \leq 0.1 \left( e^{d_{ij}/r} - e^{-d_{ij}/r} \right)^{-1}. \quad (8)$$

Thm. 4 does not allow to remove any edge before performing coordinate minimization since  $\mathbf{S}_{ij} > 0$  for any two vertices  $i$  and  $j$ . The corresponding upper bound for [10] is  $w_{ij} \leq 0.05(1 - \exp(-d_{ij}/r))^{-1}$ . These bounds are shown on Fig. 2.



**Fig. 1:** Variograms being considered for our experiments.

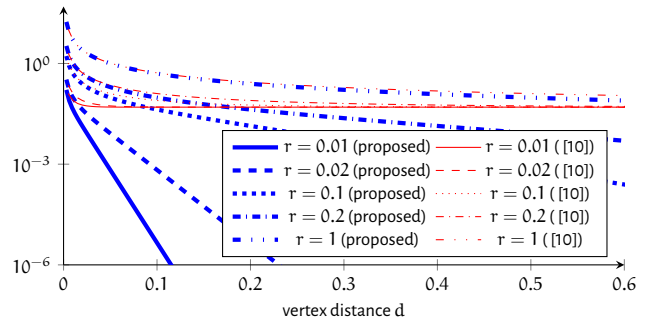
For each of these  $K$  samplings, and each of the 5 ranges, we learn i) a combinatorial Laplacian from the data covariance matrix  $\mathbf{S}$  using [10], and ii) an inner product matrix and a combinatorial Laplacian using our proposed approach. In both cases, we use the same graph initialization (graph weights between vertices obtained from a Gaussian kernel of the distance between vertices, with  $\sigma$  chosen as one third of the average distance) and let the algorithm run until the cost function is not changed by more than  $10^{-10}$ . Averages for key metrics are shown in Tab. 1.

We first study how vertex importance  $\mathbf{q}$  changes with range  $r$ . We are interested in two key quantities: how many of the vertices get the minimum importance  $q_{\min}$ ? and what is the average importance of the remaining vertices? We define the first quantity as  $u(\mathbf{q}) = |\{q_i : q_i = q_{\min}\}|/N$  ( $u$  for "unimportant"). Tab. 1 shows that all vertices are important for smaller ranges, and fewer and fewer vertices are important as range increases. Intuitively, for lower ranges, correlation between neighbors fades very quickly, as shown on Fig. 1, therefore, neighbors are not enough to explain the value of a vertex and all vertices are important to model the signal. However, for larger ranges, the value on a given vertex can be accurately inferred from its neighbors values, due to high correlations: only a small fraction of vertices are important to accurately model the whole signal.

For the vertices that are important ( $q_i > q_{\min}$ ), we also observe [25] that importance is higher for smaller ranges, while decreasing with range. The average importance  $\bar{q} = \langle \{q_i : q_i > q_{\min}\} \rangle$  in Tab. 1 shows the relation between importance and range. This follows from the cost function (4) for which, compared to the cost (1) of [10],  $\mathbf{Q}$  captures the information of  $\mathbf{S}$  not captured by  $\mathbf{L}$ . For larger ranges,  $\mathbf{L}$  is a good model using [10], and vertex importances remain low as they are not needed, while for smaller ranges, vertex importances are larger because  $\mathbf{L}$  is not enough for a good model.

These observations are especially important in the context of sampling on graphs [19], where the goal is to sample so as to minimize the  $\mathbf{Q}$ -norm of the error. With our proposed approach, the resulting importances target specific vertices whose values are important to keep to reconstruct accurately the graph signal.

We expect from the edge weight bounds in Fig. 2 that the graphs learnt with [10] should be denser when the signal has lower correlation, which corresponds here to lower ranges. We



**Fig. 2:** Edge weight upper bounds for both [10] and our proposed approach, and for varying ranges of the variogram.

**Table 1:** Average key properties of the learnt graph between [10] and our proposed approach. Bold values are better.

Method	$r$	$u(\mathbf{q})$	$\bar{q}$	$\epsilon(\mathbf{w})$	Time
[10]	0.01			0%	299.2s
	0.02			3.3e-3%	292.9s
	0.1			10.6%	295.9s
	0.2			71.0%	177.4s
	1			90.5%	<b>40.8s</b>
Proposed	0.01	0%	9.7e-2	<b>51.8%</b>	<b>58.7s</b>
	0.02	0%	9.1e-2	<b>66.5%</b>	<b>76.1s</b>
	0.1	0.3%	3.2e-2	<b>85.1%</b>	<b>71.2s</b>
	0.2	11%	1.5e-2	<b>88.6%</b>	<b>59.8s</b>
	1	75%	1.3e-2	<b>90.7%</b>	53.2s

define sparsity as the proportion of edges having 0 weight leading to:  $\epsilon(\mathbf{w}) = \|\mathbf{w}\|_0 / \frac{N(N-1)}{2}$ . This metric confirms that graphs are sparser with our proposed approach (see Tab. 1).

Our supplementary materials also show that edge weights are generally much smaller for the smaller ranges of 0.01 and 0.02 [25], by almost 4 orders of magnitude. We also remark that many of those edge weights violate the upper bound in Thm. 3, and are thus interpreted as numerical errors in the resolution. Trimming edges based on this could actually lead to even sparser graphs without sacrificing precision. We will study this in a future communication.

## 5. CONCLUSIONS AND PERSPECTIVES

In this paper, we proposed a graph learning approach based on a different graph signal spectral model, and based on learning jointly vertex importances and edge weights. As shown experimentally with continuous intrinsically stationary signals, such an approach allows for dramatically sparser graphs in the low correlation regime, more interpretable weights, and vertex importances highlighting which vertices are enough to model the signal. Adding the freedom of choosing vertex importances, this effectively lessens overfitting through a richer, more accurate, graph signal model space. Future work will include proposing edge screening similar to [10], regularization, evaluation with more adverse synthetic data or real data, theoretical guarantees, and alternative gPSD assumptions.

## 6. REFERENCES

- [1] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph Signal Processing: Overview, Challenges, and Applications," *Proc. of the IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
- [2] A. Ortega, *Introduction to Graph Signal Processing*, Cam Uni P, 2022.
- [3] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, 2019.
- [4] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the Dots: Identifying Network Structure via Graph Signal Processing," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 16–43, May 2019.
- [5] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph Learning From Data Under Laplacian and Structural Constraints," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 825–841, Sept. 2017.
- [6] B. Girault, A. Ortega, and S. S. Narayanan, "Irregularity-Aware Graph Fourier Transforms," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5746–5761, Nov. 2018.
- [7] B. Girault, "Stationary Graph Signals using an Isometric Graph Translation," in *2015 23rd Europ. Sig. Proc. Conf. (EUSIPCO)*, 2015.
- [8] B. Girault, P. Gonçalves, É. Fleury, and A. S. Mor, "Semi-supervised learning for graph to signal mapping: A graph signal wiener filter interpretation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014.
- [9] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary Graph Processes and Spectral Estimation," *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 5911–5926, Nov 2017.
- [10] E. Pavez and A. Ortega, "An Efficient Algorithm for Graph Laplacian Optimization Based on Effective Resistances," in *2019 53rd Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2019, pp. 51–55.
- [11] E. Pavez, "Laplacian constrained precision matrix estimation: Existence and high dimensional consistency," in *Proc. of the 25th International Conference on Artificial Intelligence and Statistics*, Mar 2022.
- [12] J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "Nonconvex sparse graph learning under laplacian constrained graphical model," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7101–7113, 2020.
- [13] J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "Minimax estimation of laplacian constrained precision matrices," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 3736–3744.
- [14] T. Koyakumar, M. Yukawa, E. Pavez, and A. Ortega, "Learning sparse graph with minimax concave penalty under gaussian markov random fields," *IEICE Trans. on Fund. of Elec., Comm. and Comp.r Sc.*, 2023.
- [15] E. Pavez and A. Ortega, "Generalized Laplacian precision matrix estimation for graph signal processing," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2016, pp. 6350–6354, IEEE.
- [16] M. Slawski and M. Hein, "Estimation of positive definite m-matrices and structure learning for attractive gaussian markov random fields," *Linear Algebra and its Applications*, vol. 473, pp. 145–179, 2015.
- [17] J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "Adaptive estimation of mtp2 graphical models," *arXiv preprint arXiv:2210.15471*, 2022.
- [18] E. Pavez, H. E. Egilmez, and A. Ortega, "Learning Graphs With Monotone Topology Properties and Multiple Connected Components," *IEEE Trans. on Signal Processing*, vol. 66, no. 9, pp. 2399–2413, May 2018.
- [19] B. Girault, A. Ortega, and S. S. Narayanan, "Graph Vertex Sampling with Arbitrary Graph Signal Hilbert Spaces," in *2020 IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, May 2020.
- [20] K.-S. Lu, A. Ortega, D. Mukherjee, and Y. Chen, "Perceptually Inspired Weighted MSE Optimization Using Irregularity-Aware Graph Fourier Transform," in *2020 IEEE Int. Conf. on Im. Proc.*, 2020.
- [21] E. Pavez, B. Girault, A. Ortega, and P. A. Chou, "Region Adaptive Graph Fourier Transform for 3D Point Clouds," in *2020 IEEE Int. Conf. on Im. Proc.*, Oct. 2020.
- [22] E. Pavez, B. Girault, A. Ortega, and P. A. Chou, "Two Channel Filter Banks on Arbitrary Graphs with Positive Semi Definite Variation Operators," Feb. 2023, arXiv: 2203.02858.
- [23] B. Lake and J. Tenenbaum, "Discovering structure by learning sparse graphs," in *Proc. of the Ann. Meet. of the Cog. Sc. Soc.*, 2010.
- [24] N. A. C. Cressie, *Statistics for spatial data*, Wiley, Jan. 1993.
- [25] "Supplementary materials," [https://www.benjamin-girault.com/pages/graph\\_learning.html](https://www.benjamin-girault.com/pages/graph_learning.html).



## 7. EXTENDED PROOFS

*Proof of Thm. 3.* This proof follows the approach of [11, Appendix A], but using an additional constraint enforcing positive  $q_i$ . Let  $\Theta = \mathbf{Q} + \mathbf{L}$  be the precision matrix we are estimating, such that  $\Theta_{ii} = q_i + d_i$  and  $\Theta_{ij} = -w_{ij}$ . After relaxing the constraint  $q_i > 0$ , Problem (5) is then equivalent to:

$$\min_{\substack{\Theta_{ij} \leq 0, i \neq j \\ q_{\min} \mathbf{1} - \Theta \mathbf{1} < \mathbf{o}}} -\log \det(\Theta) + \text{tr}(\Theta \mathbf{S}),$$

where we use the property that  $[\Theta \mathbf{1}]_i = q_i$ . Its Lagrangian uses the symmetric Lagrange multipliers  $\lambda_{ij}$  for the constraints on  $\Theta_{ij}$  and  $\mu_i$  for the constraint on the row sums of  $\Theta$ , leading to the following KKT conditions:

$$\begin{aligned} -\Theta^{-1} + \mathbf{S} + \mathbf{\Lambda} + \mathbf{M} &= \mathbf{o} & \Theta &\succcurlyeq 0 \\ \lambda_{ij} \Theta_{ij} &= 0, \forall i \neq j & \mu_i (q_{\min} - [\Theta \mathbf{1}]_i) &= 0, \forall i \\ \Theta_{ij} &\leq 0, \forall i \neq j & q_{\min} - [\Theta \mathbf{1}]_i &\leq 0, \forall i \\ \lambda_{ij} &\geq 0, \forall i \neq j & \mu_i &\geq 0, \forall i \end{aligned}$$

where the Lagrange multiplier matrices verify  $\mathbf{\Lambda}_{ii} = 0$ ,  $\mathbf{\Lambda}_{ij} = \mathbf{\Lambda}_{ji} = \lambda_{ij}$ ,  $\mathbf{M}_{ii} = -\mu_i$ , and  $\mathbf{M}_{ij} = -\mu_i - \mu_j$ . Considering the subset of vertices  $\mathcal{S} = \{i, j\}$ , and using the same technique as [11], the first condition (gradient is 0) leads to:

$$\Theta_{\mathcal{S}, \mathcal{S}} \geq (\mathbf{S}_{\mathcal{S}, \mathcal{S}} + \mathbf{\Lambda}_{\mathcal{S}, \mathcal{S}} + \mathbf{M}_{\mathcal{S}, \mathcal{S}})^{-1}. \quad (9)$$

Inverting the  $2 \times 2$  matrix on the r.h.s, and considering an edge  $ij$  such that its optimal weight  $w_{ij} > 0$  leads to:

$$\begin{aligned} \Theta_{ij} &\geq \frac{-(\mathbf{S}_{ij} + \lambda_{ij} - \mu_i - \mu_j)}{(\mathbf{S}_{ii} - \mu_i)(\mathbf{S}_{jj} - \mu_j) - (\mathbf{S}_{ij} + \lambda_{ij} - \mu_i - \mu_j)^2} \\ &\geq - \left[ \frac{(\mathbf{S}_{ii} - \mu_i)(\mathbf{S}_{jj} - \mu_j)}{\mathbf{S}_{ij} - \mu_i - \mu_j} - (\mathbf{S}_{ij} - \mu_i - \mu_j) \right]^{-1} \\ &\geq - \left[ \frac{\mathbf{S}_{ii} \mathbf{S}_{jj}}{\mathbf{S}_{ij}} - \mathbf{S}_{ij} \right]^{-1} \end{aligned}$$

where the second inequality is obtained using  $\lambda_{ij} = 0$  since  $w_{ij} > 0$ , and the last using  $\mu_i = 0 = \mu_j$  since  $q_i > q_{\min}$  and  $q_j > q_{\min}$ .  $\square$

*Proof of Thm. 4.* To prove the result, we prove that for non-zero edge weight  $w_{ij} > 0$  of the optimal graph we have  $\mu_i + \mu_j < \mathbf{S}_{ij}$ . The result will then follow by observing that  $\mu_i \geq 0$  and  $\mu_j \geq 0$ . Let  $i, j$  such that  $w_{ij} > 0$ . From the first equation in the KKT conditions we have that:

$$0 \leq (\Theta^{-1})_{ij} = \mathbf{S}_{ij} + \lambda_{ij} - \mu_i - \mu_j. \quad (10)$$

The inequality comes because  $\Theta$  is a generalized Laplacian ( $M$ -matrix). Since  $\lambda_{ij} = 0$ , then  $\mathbf{S}_{ij} - \mu_i - \mu_j \geq 0$ . For the strict inequality, we further assume that  $\mu_i + \mu_j = \mathbf{S}_{ij}$  and show a contradiction. Using (9), we obtain:

$$\Theta_{ij} \geq \frac{-\lambda_{ij}}{(\mathbf{S}_{ii} - \mu_i)(\mathbf{S}_{jj} - \mu_j) - \lambda_{ij}^2}.$$

Since  $\lambda_{ij} = 0$ , then  $0 > -w_{ij} = \Theta_{ij} \geq 0$ , contradiction.  $\square$

## 8. CONVERGENCE

Problem (5) is a convex optimization problem [5], and a number of standard convex optimization algorithms are guaranteed to converge to its minimum. Unfortunately, coordinate minimization with the cyclic variable selection rule is not guaranteed to converge [S1]. Problems for which non convergence appears are however degenerate cases that rarely appear in practice, and switching to a random variable selection rule when this happens can solve this issue [S1].

In a future communication, we will further study the impact of edge and vertex selection rules (the variables of the problem), with respect to both theoretical guarantees and computational efficiency.

## 9. SUPPLEMENTARY REFERENCES

[S1] S.J. Wright. Coordinate descent algorithms. *Math. Prog.*, 151(1):3–34, 2015.