LEARNING LAPLACIAN MATRIX FROM BANDLIMITED GRAPH SIGNALS

Batiste Le Bars [⋆†◇] *Pierre Humbert* [⋆◇] *Laurent Oudre* [‡] *Argyris Kalogeratos* [⋆]

[⋆] CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France
[‡] L2TI, Université Paris 13, 93430 Villetaneuse, France
[†] Sigfox R&D, 31670 Labège, France

{lebars, humbert, kalogeratos}@cmla.ens-cachan.fr; laurent.oudre@univ-paris13.fr

ABSTRACT

In this paper, we present a method for learning an underlying graph topology using observed graph signals as training data. The novelty of our method lies on the combination of two assumptions that are imposed as constraints to the graph learning process: i) the standard assumption used in the literature that signals are *smooth* with respect to graph structure (i.e. small signal variation at adjacent nodes), with ii) the additional assumption that signals are *bandlimited*, which implies sparsity in the signals' representation in the spectral domain. The latter assumption affects the inference of the whole eigenvalue decomposition of the Laplacian matrix and leads to a challenging new optimization problem. The conducted experimental evaluation shows that the proposed algorithm to solve this problem outperforms a reference state-of-the-art method that is based only on the smoothness assumption, when compared in the graph learning task on synthetic and real graph signals.

Index Terms – Graph learning, graph signal processing, signal representation, bandlimited signals, sparse coding.

1. INTRODUCTION

Graph signal processing (GSP) [1, 2] is a new and emerging field manifesting the generalization of standard signal processing tools (for example sampling [3], filtering [4], recovery [5]) to signals recorded in complex environments. Such an environment comprises of multiple entities whose interrelations, or interactions, can be encoded in a graph, and specifically in the links between its nodes. In more formal terms, a *graph signal* is a function defined on the nodes of a graph and can be represented as a vector with one component per graph node. Note that, contrary to traditional signals that encode the variation over time, a graph signal only refers to a single time instance (e.g. its acquisition time) and hence encodes the variation of an instantaneous observation over a graph structure.

Despite GSP has provided a new way to approach several real-world applications [6, 7, 8, 9], its success relies on the availability and accuracy of the underlying graph. In some contexts the graph is a priori known, perfectly or to some extent, since it may have been engineered in the first place by experts (e.g. sensor and communication networks) or can be extracted by examining the intrinsic relations between the connected entities (e.g. by considering their geographical proximity). In most situations, though, it is hard to define a suitable and sufficiently accurate graph, and hence that needs to be learned from data. This task is referred to as *data-driven graph inference* and its aim is to find the graph that best explains a given set of observed graph signals.

A body of previous work on graph learning has been based on physical models, such as epidemic models or other statistical information propagation and interaction models [10, 11, 12]. This task has been also seen as the estimation of the parameters of a Markov random field [13]. In the case of Gaussian random fields, the graph estimation consists in the estimation of the inverse covariance matrix [14, 15]. Yet, unless a statistical model is assumed for the mechanism that generates the observed graph signals, the graph inference is inevitably an ill-posed problem since several graphs may be almost equally capable to explain the same set of graph signals. Thus, solving the problem requires the introduction of constraints that would narrow down the range of possible solutions. In most related works [16, 17, 18, 19, 20] the *smoothness* constraint has been used , assuming that only small local signal variation should be expected across a graph, i.e. among adjacent nodes.

In this article we introduce the additional assumption of *bandlimitedness* which implies the sparsity of the graph signals' representation in the spectral domain. This is also a well-known concept in the GSP theory, especially for sampling tasks [21], and a property observed in real-world data [22]. To the best of our knowledge, there exists only the work in [23] that considers the bandlimitedness for the graph inference task. However, our work is the first to fully combine this assumption with that of smoothness, leading to a novel and challenging optimization problem. Experimental evaluation on synthetic and real-world data, shows that the proposed graph learning method outperforms the reference work in [24] that assumes only the smoothness of the graph signals. The results also suggest that our direction of work is promising for the task of sampling of future signal observations, or interpolation.

◇ Authors with equal contribution to this work.

2. LEARNING GRAPHS FROM SIGNALS

2.1. Notations

Let a weighted graph $G=(\boldsymbol{\mathcal{V}},\mathcal{E})$ with nodes $\boldsymbol{\mathcal{V}}=\{1,...,N\}$, edges $\mathcal{E}=\{(i,j,w_{ij}),i,j\in\boldsymbol{\mathcal{V}}\}$, and weights $w_{ij}\in\mathbb{R}^{+}$. The Laplacian matrix $\boldsymbol{\mathcal{L}}$ of the graph is defined as $\boldsymbol{\mathcal{L}}=\boldsymbol{D}-\boldsymbol{W}$, where $\boldsymbol{D}$ is the degree matrix and $\boldsymbol{W}$ the weight matrix. Assuming that G is undirected, with no self-loops, and with a single connected component, then $\boldsymbol{\mathcal{L}}$ is a symmetric positive semi-definite matrix. Therefore, its eigenvalue decomposition can be written as $\boldsymbol{\mathcal{L}}=\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{\mathsf{T}}$, with $\boldsymbol{\Lambda}=\text{diag}(\lambda_1,...,\lambda_N)$ a diagonal matrix with the eigenvalues and $\boldsymbol{X}=(\boldsymbol{x}_1,...,\boldsymbol{x}_N)$ a matrix with the eigenvectors as columns. Note that for a graph with one connected component, it holds that $\lambda_1=0$ and $\boldsymbol{x}_1=\mathbf{1}_N$, where $\mathbf{1}_N$ is the constant unitary vector of size N.

A graph signal can be represented as a vector $\boldsymbol{y}\in\mathbb{R}^N$, where y_i is the function value at the i-th node. Using the Graph Fourier Transform (GFT) it is possible to create a spectral representation $\boldsymbol{h}$ for $\boldsymbol{y}$ defined as $\boldsymbol{h}=\boldsymbol{X}^{\mathsf{T}}\boldsymbol{y}$. The eigenvalues can be interpreted as distinct frequencies, the components of $\boldsymbol{h}$ as Fourier coefficients, and the eigenvectors as a decomposition basis. A graph signal $\boldsymbol{y}$ is said to be *smooth* if adjacent nodes tend to exhibit similar behavior. Smoothness can be quantified with various metrics, though probably the most common is:

$$\boldsymbol{y}^{\mathsf{T}}\boldsymbol{\mathcal{L}}\boldsymbol{y}=\frac{1}{2}\sum_{i,j}w_{ij}\left(y_i-y_j\right)^2. \tag{1}$$

A signal gets smoother, thus (1) lower, when its value at any two nodes gets closer as their edge weight gets larger. The smoothness property is naturally present in real-world graph signals [16] and it has consequently been widely considered for graph inference [17, 24].

A graph signal $\boldsymbol{y}$ is said to be ω-*bandlimited* when its Fourier coefficients $\boldsymbol{h}$, associated to frequencies λ_k which are higher than a real value ω, are assumed to be null, or formally: $h_i=0,\forall\lambda_i>\omega$. The bandlimitedness assumption can also be interpreted as a sparsity assumption on $\boldsymbol{h}$. This assumption on graph signals is very common, especially in GSP where it is the main hypothesis of several sampling methods [25, 26, 27, 3]. However, this work is the first to fully employ it for the graph learning task. Intuitively, this property refers to the fundamental principle of signal processing which suggests filtering out the band of high frequencies of a signal where there assumably lies mostly noise and little or no information.

Let us finally consider that n graph signals $\{\boldsymbol{y}^{(k)}\}_{k=1}^n$ of size N compose a matrix $\boldsymbol{Y}=[\boldsymbol{y}^{(1)},...,\boldsymbol{y}^{(n)}]\in\mathbb{R}^{N\times n}$.

2.2. Problem statement

The general task of *data-driven graph inference* aims at learning the graph G that best explains the structure of the observed graph signals $\boldsymbol{Y}$. What we propose in this article is to learn the graph that best explains the structure of the approximation of the signals $\boldsymbol{Y}$, which has to be smooth and bandlimited on G. To that end, we introduce the optimization problem:

$$\min_{\boldsymbol{H},\boldsymbol{X},\boldsymbol{\Lambda}}\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{H}\|_{\text{F}}^2+\alpha\|\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}\|_{\text{F}}^2+\beta\|\boldsymbol{H}\|_{2,1} \tag{2}$$

$$\text{s.t.}\begin{cases}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}=\boldsymbol{I}_N,\ \boldsymbol{x}_1=\mathbf{1}_N & \text{(a)}\\ (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{\mathsf{T}})_{i,j}\leq 0\quad i\neq j, & \text{(b)}\\ \boldsymbol{\Lambda}=\text{diag}(0,\lambda_2,...,\lambda_N)\succeq 0 & \text{(c)}\\ \text{tr}(\boldsymbol{\Lambda})=N\in\mathbb{R}_*^+. & \text{(d)}\end{cases}$$

This problem aims at learning conjointly the graph G (defined by its Laplacian matrix $\boldsymbol{\mathcal{L}}=\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{\mathsf{T}}$) and a smooth bandlimited approximation $\boldsymbol{X}\boldsymbol{H}$ of the noisy observed signals $\boldsymbol{Y}$. The three terms composing the objective function (2) are:

1. The first term corresponds to the quadratic approximation error of $\boldsymbol{Y}$ by $\boldsymbol{X}\boldsymbol{H}$, where $\|\cdot\|_F$ is the Frobenius norm.
2. The second term is a *smoothness regularization* imposed to the approximation $\boldsymbol{X}\boldsymbol{H}$. Rewriting equation (1) for the set of graph signals $\boldsymbol{X}\boldsymbol{H}$, we get $\|\boldsymbol{\mathcal{L}}^{1/2}\boldsymbol{X}\boldsymbol{H}\|_{\text{F}}^2=\|\boldsymbol{X}\boldsymbol{\Lambda}^{1/2}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{H}\|_{\text{F}}^2=\|\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}\|_{\text{F}}^2$. This kind of regularization is very common in graph learning [18, 20].
3. The last term $\|\boldsymbol{H}\|_{2,1}$ is a *sparsity regularization* that corresponds to the sum of the ℓ_2-norm of each row of $\boldsymbol{H}$. This is the main contribution of our work, which is motivated by the bandlimitedness assumption on the graph signals. Indeed, this hypothesis implies weights equal to zero at the same dimension for every constructed signal, and such norm will enforce that feature.

Finally, $\alpha,\beta>0$ are hyperparameters controlling respectively the smoothness and the sparsity of $\boldsymbol{H}$. Regarding the constraints, the first three (a, b, c) enforce $\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{\mathsf{T}}$ to be a proper combinatorial Laplacian with a single connected component. The last one, (d), was proposed in [24] to impose structure in the learned graph so that the trivial solution $\lambda=\mathbf{0}$ is avoided. A discussion about values other than N is made in [18].

3. SOLVING THE OPTIMIZATION PROBLEM

The proposed optimization algorithm to solve the problem (2) is an alternating minimization procedure [28] on $\boldsymbol{H}$, $\boldsymbol{X}$, and $\boldsymbol{\Lambda}$, knowing $\boldsymbol{Y}$. The overall optimization procedure is given by the *Graph Learning for Smooth and Sparse Spectral Representation* (**GL-3SR**) in Alg. 1. At the beginning, it requires the initialization of $\boldsymbol{X}$ and $\boldsymbol{\Lambda}$ which can be done by creating a graph with one connected component (e.g. a prior graph, a complete graph, or just a random graph) and then compute the Singular Value Decomposition (SVD) of its Laplacian matrix.

3.1. Optimization with respect to $\boldsymbol{H}$

For fixed $\boldsymbol{X}$, $\boldsymbol{\Lambda}$, the problem (2) can be written as:

$$\min_{\boldsymbol{H}}\left\|\begin{bmatrix}\boldsymbol{Y}\\ \mathbf{0}_{\boldsymbol{N}\times\boldsymbol{n}}\end{bmatrix}-\begin{bmatrix}\boldsymbol{X}\\ (\alpha\boldsymbol{\Lambda})^{1/2}\end{bmatrix}\boldsymbol{H}\right\|_{\text{F}}^2+\beta\|\boldsymbol{H}\|_{2,1}, \tag{3}$$

where $\mathbf{0}_{N\times n}$ is a zero matrix of size $N\times n$. The form of (3) is a particular case of Group Lasso [29] with equal weights for each group. Thus, it can be solved efficiently using a suited optimization procedure, such as the coordinate descent [30].

3.2. Optimization with respect to $\boldsymbol{X}$

With respect to $\boldsymbol{X}$, the optimization problem is not convex:

$$\min_{\boldsymbol{X}} \|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{H}\|_{\mathrm{F}}^2 \tag{4}$$

$$\text{s.t.} \begin{cases} \boldsymbol{X}^\mathsf{T}\boldsymbol{X}=\boldsymbol{I}_N,\ \boldsymbol{x}_1=\mathbf{1}_N & \text{(a)} \\ (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\mathsf{T})_{i,j}\leq 0 \quad i\neq j. & \text{(b)} \end{cases}$$

To our knowledge, there exists no method solving (4) with both the above constraints. In the literature, solutions have been proposed considering only the constraint $\boldsymbol{X}^\mathsf{T}\boldsymbol{X}=\boldsymbol{I}_N$ [31, 32]. Due to the difficulty of the problem and the lack of space, our proposed workaround is a closed-form solution to (4) under the constraint (a) only, which is explained in the following proposition.

Proposition 1. (*Closed-form solution for the relaxation of problem* (4).) – Consider the optimization with the constraint (a) only. Let $\boldsymbol{X}_0$ be any matrix that belongs to the constraints set (a), and $\boldsymbol{M}=(\boldsymbol{X}_0^\mathsf{T}\boldsymbol{Y}\boldsymbol{H}^\mathsf{T})_{1:,1:}$ the submatrix containing everything but the input's first row and first column. Finally, let $\boldsymbol{P}\boldsymbol{D}\boldsymbol{Q}^\mathsf{T}$ be the SVD of $\boldsymbol{M}$. Then, the problem admits the following closed form solution:

$$\boldsymbol{X}^*=\boldsymbol{X}_0\begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & \boldsymbol{P}\boldsymbol{Q}^\mathsf{T}\end{bmatrix}. \tag{5}$$

Proof. By replacing $\boldsymbol{X}$ with $\boldsymbol{X}_0\tilde{\boldsymbol{G}}^*$, one can observe that the relaxed optimization problem is equivalent to finding:

$$\boldsymbol{G}^*=\operatorname*{argmin}_{\boldsymbol{G}}\left\|\boldsymbol{Y}-\boldsymbol{X}_0\underbrace{\begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & \boldsymbol{G}\end{bmatrix}}_{\triangleq\tilde{\boldsymbol{G}}}\boldsymbol{H}\right\|_{\mathrm{F}}^2 \tag{6}$$

s.t. $\boldsymbol{G}^\mathsf{T}\boldsymbol{G}=I_{N-1}$. Solving (6) is equivalent to finding:

$$\boldsymbol{G}^*=\arg\max_{\boldsymbol{G}}\operatorname{tr}\left(\boldsymbol{H}\boldsymbol{Y}^\mathsf{T}\boldsymbol{X}_0\tilde{\boldsymbol{G}}\right)=\arg\max_{\boldsymbol{G}}\operatorname{tr}\left(\boldsymbol{M}^\mathsf{T}\boldsymbol{G}\right)$$

s.t. $\boldsymbol{G}^\mathsf{T}\boldsymbol{G}=I_{N-1}$. Then, as proved in [31], we finally have $\boldsymbol{G}^*=\boldsymbol{P}\boldsymbol{Q}^\mathsf{T}$, which completes the proof. □

The constraints (a) and (b) are equally important to obtain a valid Laplacian matrix at the end, however reducing the problem (4) by the use of the closed-form (5) does not ensure that the constraint (b) will be satisfied. Nevertheless, one may notice that for any $\boldsymbol{X}$ that satisfies (a) but not necessarily (b), there always exists a $\boldsymbol{\Lambda}$ that makes all constraints satisfied (it suffices to take $\lambda_2=\ldots=\lambda_N=\frac{N}{N-1}$). For these reason, we propose to use the closed-form solution of Proposition (1) to learn $\boldsymbol{X}$ and always optimize with respect to $\boldsymbol{\Lambda}$ right after.

Algorithm 1 - Graph Learning with **GL-3SR**

1: **Input :** the signals $\boldsymbol{Y}$, α, β
2: **Output :** the sparse spectral signals $\boldsymbol{H}$, $\boldsymbol{X}$, $\boldsymbol{\Lambda}$
3: **Initialization :** $\boldsymbol{X}, \boldsymbol{\Lambda}$
4: **for** $t=1,2,\ldots$ **do**
5: **Update** $\boldsymbol{H}$: Solve the lasso problem (3).
6: **Update** $\boldsymbol{X}$: Compute the closed-form solution (5).
7: **Update** $\boldsymbol{\Lambda}$: Solve the linear program (7).
8: **end for**

3.3. Optimization with respect to $\boldsymbol{\Lambda}$

With respect to $\boldsymbol{\Lambda}$, the optimization problem (2) becomes:

$$\min_{\boldsymbol{\Lambda}}\alpha\operatorname{tr}(\boldsymbol{H}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{H}) \tag{7}$$

$$\text{s.t.} \begin{cases} (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\mathsf{T})_{i,j}\leq 0 \quad i\neq j, & \text{(b)} \\ \boldsymbol{\Lambda}=\operatorname{diag}(0,\lambda_2,\ldots,\lambda_N)\succeq 0 & \text{(c)} \\ \operatorname{tr}(\boldsymbol{\Lambda})=N\in\mathbb{R}_*^+, & \text{(d)} \end{cases}$$

which is a linear program that can be solved efficiently using linear cone programs.

4. EXPERIMENTS

In this section we use synthetic and real data to compare the proposed graph learning **GL-3SR** [1] method against **GL-SigRep** [24], which is a reference state-of-the-art method in the related literature.

Results on synthetic data. Having access to the ground truth of synthetic data, we provide visual and quantitative comparisons of the learned $\boldsymbol{H}$ and $\boldsymbol{W}$ using the *Squared Error* and the F*-measure* (in particular, the F_1-measure), often used in this type of experimental evaluation [34]. In each experiment and for both methods, the best set of hyperparameters were first found with regards to the considered evaluation metrics, while also achieving a similar error in reconstructing the true graph signals. Furthermore, as explained in [34], since F-measure is defined for binary and relatively sparse variables, we evaluated on a thresholded version of the learned $\boldsymbol{W}$ each time.

Graphs: We carried out experiments on graphs with 20 vertices, following: i) a Random Geometric Graph (RGG - see [24] for details) model with a truncated Gaussian kernel of width size 0.5, where weights smaller than 0.75 were set to 0; ii) an Erdős-Rényi (ER) model with edge probability 0.2.

Graph signals: Given a graph, our sampling process for graph signals was very similar to the one in [24], with $\boldsymbol{y}=\boldsymbol{X}\boldsymbol{h}+\varepsilon$, and $\boldsymbol{h}$, ε following Gaussian distributions. The difference brought by the bandlimitedness assumption is that $\boldsymbol{h}$ is supposed to follow a multivariate Gaussian distribution with degenerate values at some dimensions. Its precision matrix is defined as the eigenvalue matrix of the graph Laplacian $\boldsymbol{\mathcal{L}}$

[1] The source code of our implementations is available at `https://github.com/pierreHmbt/GL-3SR`. We used the scikit-learn's [33] function `MultiTaskLasso` to solve the optimization problem (3) and the Python's `CVXOPT` package to solve the linear program (7).

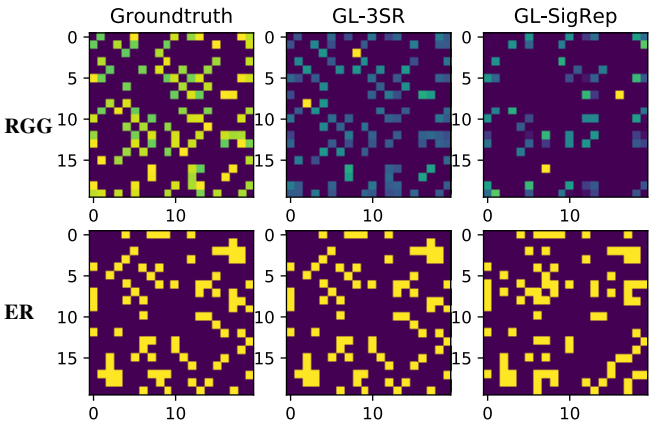

Fig. 1: Comparison of the ground truth, and the adjacency matrices learned (after thresholding) by the competing methods (columns), for the considered random graph models (rows).

Metric	Method	ER	RGG
Error (L)	**GL-3SR**	$\mathbf{0.113(\pm 0.032)}$	$\mathbf{0.075(\pm 0.028)}$
	GL-SigRep	$0.862(\pm 0.301)$	$0.342(\pm 0.074)$
F-measure (W)	**GL-3SR**	$\mathbf{0.868(\pm 0.074)}$	$\mathbf{0.895(\pm 0.032)}$
	GL-SigRep	$0.677(\pm 0.117)$	$0.807(\pm 0.037)$
F-measure (H)	**GL-3SR**	$\mathbf{0.478(\pm 0.065)}$	$\mathbf{0.816(\pm 0.045)}$
	GL-SigRep	$0.314(\pm 0.017)$	$0.616(\pm 0.015)$

Table 1: Comparison of the two learning methods on synthetic data, according to three quality metrics (average $\pm$ std). The best performance is indicated in bold.

where the largest values were set to 0. As in [24], the error term ε follows a multivariate Gaussian distribution with zero mean and covariance $\sigma_\varepsilon^2 \boldsymbol{I}_N$. The probability density functions are thus given by: $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 I_p)$ and $\boldsymbol{h} \sim \mathcal{N}(\mu, \boldsymbol{\Lambda}_K^\dagger)$, where for $0 < K \leq N$, $\text{diag}(\boldsymbol{\Lambda}_K) = (\lambda_1, ..., \lambda_K, 0, ..., 0)$ and $\dagger$ stands for the Moore-Penrose pseudo inverse. The mean value of each signal was set to 0, the variance of the noise was set to 0.5^2 and the parameter K that enforces the bandlimitedness property was set to 10 (i.e. half the size of $\boldsymbol{\mathcal{V}}$).

For each type of graph, we ran 10 experiments with 1000 graph signals sampled as explained earlier, i.e. $\boldsymbol{Y} \in \mathbb{R}^{20 \times 1000}$. The average value of the evaluation metrics and their standard deviation are compared in Tab. 1; the learned weighted adjacency matrices (before and after thresholding) are displayed in Fig. 1. These indicate that in this sampling process the proposed **GL-3SR** method managed to infer a better graph, closer to the one of the ground truth, compared to **GL-SigRep**. This example can be considered as a proof of concept, since it shows clearly that if the (denoised) graph signals are bandlimited, then our method performs well.

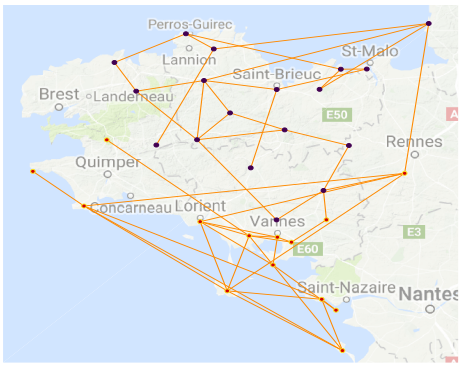

Fig. 2: Graph segmentation in two parts (red and black nodes) with spectral clustering using the Laplacian matrix learned by the **GL-3SR** algorithm.

Results on real-world data. We used hourly temperature measurements on 32 weather stations in Brittany, France, during a period of 31 days. The dataset contains 744 graph signals in total, i.e. $\boldsymbol{Y} \in \mathbb{R}^{32 \times 744}$, which are assumed to be smooth and bandlimited on the unknown underlying graph. Our objective is to infer that graph using **GL-3SR**. The two hyperparameters were chosen to maximize both smoothness and sparsity, while not deteriorating the recovery of the true signals.

We first show the capacity of the learned graph to allow efficient subsampling and interpolation. More specifically, the learned graph is in accordance with the one found in [20]. The important difference is, however, that in this graph the signals are 17-bandlimited. This property allows to: i) discard more than half of the nodes (subsampling); ii) reconstruct the true signals with an RMSE$= 0.087$ when only 15 values on nodes are used (interpolation). In addition, as displayed in Fig. 2, the spectral clustering algorithm [35], using the five eigenvectors associated to the five lowest eigenvalues of the resulting Laplacian, segments the learned graph in two parts corresponding to the north and the south of Britanny.

5. CONCLUSIONS

We presented a data-driven graph learning approach by employing a combination of two assumptions. The first is standard in the related literature and concerns the *smoothness* of graph signals with respect to the underlying graph structure. The second is the *bandlimitedness* assumption which implies sparsity in the signals' representation in the spectral domain. We then presented the **GL-3SR** method to solve the derived new and challenging optimization problem. The findings of our empirical evaluation showed that the proposed approach outperforms the reference state-of-the-art **GL-SigRep** method in synthetic and real-world data The results also suggest that our method learns graphs that are promising in the context of graph subsampling and signal interpolation, which we plan to investigate further.

6. REFERENCES

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

[2] P.M. Djuric and C. Richard, Eds., *Cooperative and Graph Signal Processing – Principles and Applications*, Elsevier, 2018.

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

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

[5] S. Chen, A. Sandryhaila, J.M.F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *Trans. Signal Processing*, vol. 63, no. 17, pp. 4609–4624, 2015.

[6] G. Cheung, E. Magli, Y. Tanaka, and M.K. Ng, "Graph spectral image processing," *Proc. of the IEEE*, vol. 106, no. 5, pp. 907–930, 2018.

[7] R. Wagner, H. Choi, R. Baraniuk, and V. Delouille, "Distributed wavelet transform for irregular sensor network grids," in *IEEE/SP Workshop on Statistical Signal Processing*, 2005, pp. 1196–1201.

[8] R.K. Jain, J.M.F. Moura, and C.E. Kontokosta, "Big data+ big cities: Graph signals of urban air pollution [exploratory sp]," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 130–136, 2014.

[9] D.M. Mohan, M.T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Wavelets on graphs with application to transportation networks," in *Proc. of the Int. Conf. on Intelligent Transportation Systems*. IEEE, 2014, pp. 1707–1712.

[10] M.G. Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," in *Proc. of the Int. Conf. on Machine Learning*, 2011, pp. 561–568.

[11] N. Du, L. Song, M. Yuan, and A.J. Smola, "Learning networks of heterogeneous influence," in *Advances in Neural Information Processing Systems*, 2012, pp. 2780–2788.

[12] R. Lemonnier, K. Scaman, and A. Kalogeratos, "Multivariate hawkes processes for large-scale inference," in *Proc. of the AAAI Conf. on Artificial Intelligence*, 2013, pp. 3384–3391.

[13] D. Koller, N. Friedman, and F. Bach, *Probabilistic graphical models: principles and techniques*, MIT press, 2009.

[14] O. Banerjee, L. El Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data," *J. of Machine Learning Research*, vol. 9, no. Mar, pp. 485–516, 2008.

[15] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[16] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *arXiv preprint arXiv:1806.00848*, 2018.

[17] S.I. Daitch, J.A. Kelner, and D.A. Spielman, "Fitting a graph to vector data," in *Proc. of the Int. Conf. on Machine Learning*, 2009, pp. 201–208.

[18] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. of the conf. on Artificial Intelligence and Statistics*, 2016, pp. 920–929.

[19] H.E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under structural and laplacian constraints," *arXiv preprint arXiv:1611.05181*, 2016.

[20] S.P. Chepuri, S. Liu, G. Leus, and A.O. Hero, "Learning sparse graphs under smoothness prior," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2017, pp. 6508–6512.

[21] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Processing*, vol. 63, no. 24, pp. 6510–6523, 2015.

[22] D. Valsesia, G. Fracastoro, and E. Magli, "Sampling of graph signals via randomized local aggregations," *arXiv preprint arXiv:1804.06182*, 2018.

[23] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, "Graph topology inference based on transform learning," in *Proc. of the Global Conf. on Signal and Information Processing*, 2016, pp. 356–360.

[24] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.

[25] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs.," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2014, pp. 3864–3868.

[26] S.K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2013, pp. 5445–5449.

[27] S. Chen, A. Sandryhaila, and J. Kovačević, "Sampling theory for graph signals," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2015, pp. 3392–3396.

[28] J. De Leeuw, F.W. Young, and Y. Takane, "Additive structure in qualitative data: An alternating least squares method with optimal scaling features," *Psychometrika*, vol. 41, no. 4, pp. 471–503, 1976.

[29] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[30] S.J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[31] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[32] C. Bao, J-F. Cai, and H. Ji, "Fast sparsity-based orthogonal dictionary learning for image restoration," in *Proc. of the IEEE Int. Conf. on Computer Vision*, 2013, pp. 3384–3391.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, and B. et al. Thirion, "Scikit-learn: Machine learning in Python," *J. of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[34] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M.G. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Trans. Signal and Information Processing over Networks*, 2017.

[35] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, 2001, pp. 849–856.