ADAPTIVE SUBSAMPLING OF MULTIDOMAIN SIGNALS WITH PRODUCT GRAPHS

Théo Gnassounou, Pierre Humbert, and Laurent Oudre

Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, F-91190, Gif-sur-Yvette, France.
Université de Paris, CNRS, Centre Borelli, F-75005, Paris, France.

ABSTRACT

In this paper, we propose an adaptive subsampling method for multidomain signals based on the constrained learning of a product graph. Given an input multidomain signal, we search for a product graph on which the signal is bandlimited, i.e. have limited spectral occupancy. The subsampling procedure described in this article is composed of two successive steps. First, we use the input data to learn a graph that will be optimized to favor efficient sampling. Then, we derive an algorithm for choosing the best nodes and provide a sampling strategy for multidomain signals. Experiments on synthetic data and two real datasets show the efficiency of the proposed method and its relevance for multidomain data compression and storing.

***Index Terms*—** Graph signal processing, graph product, subsampling, graph learning, Laplacian matrix estimation

1. INTRODUCTION

In standard signal processing, the sampling theory states that a bandlimited signal sampled above its Nyquist rate can be perfectly reconstructed. This important property is at the cornerstone of the sampling of Euclidean signals. However, when signals are defined over a more complex domain, the design of adaptive sampling strategies is still an active topic of interest. In order to deal with signals residing on irregular domains, Graph Signal Processing (GSP) [1, 2] has emerged as a powerful alternative to standard approaches. In this formalism, the graph defines a support, and the signals, now called graph signals, are defined on this support. This allows to capture the structure on which a signal evolves, hence providing more information than considering the signal alone. By generalizing concepts and tools of signal processing to signals recorded over graphs, GSP has proven its success in many tasks such as filtering [3], reconstruction [4], and sampling [5].

For the later, one idea proposed in the univariate case is to reconstruct a graph signal from its measurement at some nodes by taking advantage of its underlying graph. This approach known as graph sampling set selection (or subset sampling) is now well investigated [6, 7, 8]. For instance, (in the noiseless setting) assuming that the graph signal is bandlimited one can show that a random selection of a reasonable number of samples/nodes is sufficient to enjoy a perfect reconstruction with a high probability [9]. Unfortunately, there exist some major limitations of such methods.

First, to date, most articles have focused on univariate signals. However, recent publications in GSP have argued the need for multidomain graph signal processing that allows to deal with tensor data instead or vector data [10, 11]. Indeed, in several application contexts such as sensor networks, data streams are recorded as multivariate time series that evolve on a network, thus defining at least three domains: space, time and dimension. Second, these methods are mostly based on the computation of the first eigenvectors of the Laplacian. This step is computationally prohibitive for large graphs. Third, as in many computational tasks such as spectral clustering, and semi-supervised learning, the availability of the underlying graph is a core assumption. However, in most situations, no natural graph can be derived or defined and this graph must be inferred from the available data.

In this paper, we propose to address these issues in order to provide an efficient adaptive subsampling method for multidomain signals. Our approach extensively relies on tensor algebra to take into account the multidomain property of signals. To circumvent the complexity problem, we exploit the product structure of large graphs by factorizing them as the product graph of smaller graphs. This is a standard technique used to lower the complexity of GSP algorithms [12, 13]. The adaptive property of our subsampling procedure is based on a graph learning step that will search for a graph on which the signals can be efficiently sampled. As this graph learning problem is inherently an ill-posed problem, most state-of-the-art approaches use additional constraints such as smoothness (minimal signal variability between adjacent nodes) in order to retrieve an adequate graph [14, 15]. In the context of adaptive sampling, which is the core of this article, a less-studied constraint naturally emerges: the bandlimitness [16, 17, 18]. Indeed, Graph Fourier Transform (GFT) theory insures that if a graph signal if K-bandlimited (i.e. has only K non null spectral coefficients), then a perfect recovery is possible from its values on only K well-chosen nodes [6]. In order to take advantage of this property, we will seek for a graph where the signals have a sparse spectral representation, which will in turn ensure that the signals will be easy to sample efficiently.

2. BACKGROUND

We now introduce the notations of this paper on tensor algebra and graph product (see [19, 20] for more complete introductions).

Tensor algebra. For $d_1, d_2, \dots, d_P \in \mathbb{N}_+$, let denote a tensor of order P by $\mathcal{Y} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_P}$. This tensor can be seen as a discretized multidomain signal with each of its entries indexed over P domains. The matrix $\mathbf{Y}^{(m)}$ in $\mathbb{R}^{d_m \times (d_1 \dots d_{m-1} \cdot d_{m+1} \dots d_P)}$ represents the tensor $\mathcal{Y}$ unfolded along the dimension m . The *mode-m* matrix product between a matrix $\mathbf{X} \in \mathbb{R}^{J \times d_m}$ and $\mathcal{Y}$ is defined as $\mathcal{Y} \times_m \mathbf{X} \iff \mathbf{X}\mathbf{Y}^{(m)}$. The operator $\otimes$ represent the Kronecker product. When multiple products are necessary, we use upper version of the notations.

Graph notations. Throughout the paper, let consider a weighted and undirected graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of N nodes and $\mathcal{E}$ is a set of edges. The combinatorial Laplacian matrix of this graph is a N by N matrix defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the degree matrix and $\mathbf{W}$ the weight matrix. Since G is an undirected

graph, $\mathbf{L}$ is a symmetric and positive semi-definite matrix verifying $\mathbf{L} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^\mathsf{T}$ with $\mathbf{\Lambda}$ the diagonal matrix of non-negative eigenvalues of $\mathbf{L}$ and $\mathbf{X}$ the matrix of the corresponding eigenvectors as columns. Assuming that G has one connected component, $\mathbf{L}$ has $\lambda_1 = 0$ for first eigenvalue associated with the eigenvectors $\mathbf{X}_{:,1} = \mathbf{1}_N/\sqrt{N}$ with $\mathbf{1}_N$ the unitary vector of size N. The space of such Laplacian matrices is given by:

$$\mathcal{L} = \{\mathbf{L} \in \mathbf{S}_+^N \mid \mathbf{L}\mathbf{1}_N = \mathbf{0}_N,\ \mathbf{L}_{ij} = \mathbf{L}_{ji} \geq 0,\ \forall i \neq j\}\ , \quad (1)$$

where $\mathbf{S}_+^N$ is the set of real, symmetric and positive semi-definite matrices of size $N \times N$.

Graph Signal Processing. A graph signal is a function $\mathbf{y} : \mathcal{V} \to \mathbb{R}^N$ assigning a scalar value to each node of a graph G. This function can be represented as a vector $\mathbf{y} \in \mathbb{R}^N$, where $\mathbf{y}_i$ is the function value at the i-th node. The eigenvectors of the Laplacian of G provide a Fourier-like basis for graph signals, allowing to decompose any signal into its spectral components. From this formalism, the Graph Fourier Transform (GFT) of $\mathbf{y}$ is defined by $\mathbf{h} = \mathbf{X}^\mathsf{T}\ \mathbf{y}$. A K-bandlimited graph signal is thus a signal for which $\mathbf{h}_i \neq 0$ in K entries [1, 2].

Cartesian graph product. Consider two weighted and undirected graphs $G_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $G_2 = (\mathcal{V}_2, \mathcal{E}_2)$ with N_1 and N_2 nodes. According to the foregoing there are $\mathbf{L}_1 = \mathbf{X}_2\mathbf{\Lambda}_1\mathbf{X}_1^\mathsf{T}$ and $\mathbf{L}_2 = \mathbf{X}_2\mathbf{\Lambda}_2\mathbf{X}_2^\mathsf{T}$. A graph G is said to be a Cartesian graph when it is the result of the Cartesian product between G_1 and G_2. The Laplacian matrix of G is then given by:

$$\mathbf{L} = (\mathbf{X}_1 \otimes \mathbf{X}_2)(\mathbf{\Lambda}_1 \oplus \mathbf{\Lambda}_2)(\mathbf{X}_1^\mathsf{T} \otimes \mathbf{X}_2^\mathsf{T})\ , \quad (2)$$

where $\oplus$ denotes the Kronecker sum i.e. for $\mathbf{A} \in \mathbb{R}^{N_1 \times N_1}$ and $\mathbf{B} \in \mathbb{R}^{N_2 \times N_2}$, $\mathbf{A} \oplus \mathbf{B} = \mathbf{I}_2 \otimes \mathbf{A} + \mathbf{B} \otimes \mathbf{I}_1$ with $\mathbf{I}_n$ the identity matrix of size N_n.

The notion of GFT and K-bandlimitedness can be extended in this multidomain context. The multidomain GFT of a graph signal $\mathbf{Y} \in \mathbb{R}^{N_1 \times N_2}$, is given by:

$$\mathbf{H} = \mathbf{X}_1\mathbf{Y}\mathbf{X}_2^\mathsf{T} \Longleftrightarrow \mathbf{H} = \mathbf{Y} \times_1 \mathbf{X}_1^\mathsf{T} \times_2 \mathbf{X}_2^\mathsf{T}\ . \quad (3)$$

Furthermore, given a vector $\mathbf{K} = (K_1, K_2)$, a $\mathbf{K}$-bandlimited graph signal is a signal for which $\mathbf{H}$ has K_1 rows and K_2 columns different than zero. In other word, a signal is $\mathbf{K}$-bandlimited when it is simultaneously K_i-bandlimited respectively in each domain.

It is possible to generalize equations (2) and (3) to P graphs $(G_p)_{p=1}^P$ with (N_p) nodes. Denoting by $\mathbf{L}_p = \mathbf{X}_p\mathbf{\Lambda}_p\mathbf{X}_p^\mathsf{T}$ their respective Laplacian matrices, the Laplacian matrix of their Cartesian products is [20]:

$$\mathbf{L} = (\bigotimes_{p=1}^P \mathbf{X}_p)\bigoplus_{p=1}^P \mathbf{\Lambda}_p(\bigotimes_{p=1}^P \mathbf{X}_p^\mathsf{T})\ . \quad (4)$$

The multidomain GFT for the graph signal $\mathcal{Y} \in \mathbb{R}^{N_1 \times N_2 \times \ldots \times N_P}$ is then given by $\mathcal{H} = \mathcal{Y} \underset{p=1}{\overset{P}{\times}} \mathbf{X}_p^\mathsf{T}$. Recall that by definition of the mode product, we also have $\mathcal{Y} = \mathcal{H} \underset{p=1}{\overset{P}{\times}} \mathbf{X}_p$.

3. SUBSAMPLING OF MULTIDOMAIN SIGNALS

Let consider a multidomain signal $\mathcal{Y} \in \mathbb{R}^{N_1 \times \ldots \times N_P}$ and a vector $\mathbf{K} = (K_1, \ldots, K_P)$. We want to efficiently subsample $\mathcal{Y}$ by using only $\prod_{p=1}^P K_p$ samples. This problem can be decomposed into two main tasks. First, learn an adaptive multidomain graph representation where the signal is $\mathbf{K}$-bandlimited. Second, select the $\prod_{p=1}^P K_i$ nodes (i.e. samples) that allow to achieve the best reconstruction errors.

3.1. Learning the graph Fourier basis

In this step, our goal is to learn the P graph Fourier basis $\{\mathbf{X}_p\}_{p=1}^P$ of each subgraph composing the Cartesian graph together with the sparse spectral representation $\mathcal{H}$ of the signal on this graph.
The associated optimization problem writes as:

$$\begin{aligned} \underset{\{\mathbf{X}_p\}_p, \mathcal{H}}{\text{minimize}} \quad & \|\mathcal{Y} - \mathcal{H} \underset{p=1}{\overset{P}{\times}} \mathbf{X}_p\|_F^2 \\ \text{s.t.} \quad & \mathbf{X}_p^\mathsf{T}\mathbf{X}_p = \mathbf{I}, \mathbf{x}_{p_1} = \mathbf{1}_{N_p}/\sqrt{N_p},\ \forall p \\ & \|\mathbf{H}^{(p)}\|_{2,0} \leq K_p,\ \forall p\ , \end{aligned} \quad (5)$$

where $\|\cdot\|_F$ is the Frobenius norm and $\|\mathbf{H}^{(p)}\|_{2,0}$ simply counts the number of nonzero rows in the matrix $\mathbf{H}^{(p)}$. In this problem, the first constraints insures that the matrices $\{\mathbf{X}_p\}_p$ are proper graph Fourier basis and the second that the input signal is $\mathbf{K}$-bandlimited in the spectral domain induced by the learned basis. Indeed, the norm imposes zeros on the rows of the unfolding $\mathcal{H}$ which make it adapted for the bandlimitedness assumption. To solve (5) we use an alternating procedure on $\mathbf{X}_1, \ldots, \mathbf{X}_P$ and $\mathcal{H}$.

$\mathbf{X}_{\bar{p}}$ computation. To learn $\mathbf{X}_{\bar{p}}$ with fixed $(\{\mathbf{X}_p\}_{p \neq \bar{p}}, \mathcal{H})$, we reformulate the objective function of (5) into:

$$\begin{aligned} \|\mathcal{Y} - \mathcal{H} \underset{p=1}{\overset{P}{\times}} \mathbf{X}_p\|_F^2 &= \|\mathcal{Y} \times_{\bar{p}} \mathbf{X}_{\bar{p}}^\mathsf{T} - \mathcal{H} \underset{p=1, p\neq\bar{p}}{\overset{P}{\times}} \mathbf{X}_p^\mathsf{T}\|_F^2 \\ &= \|\mathbf{X}_{\bar{p}}^\mathsf{T}\mathbf{Y}^{(\bar{p})} - \bar{\mathbf{H}}^{(\bar{p})}\|_F^2\ , \end{aligned} \quad (6)$$

with $\bar{\mathbf{H}}^{(\bar{p})} \triangleq (\mathcal{H} \underset{p=1, p\neq\bar{p}}{\overset{P}{\times}} \mathbf{X}_p^\mathsf{T})^{(\bar{p})}$. This manipulation leads to the following optimization problem:

$$\begin{aligned} \underset{\mathbf{X}_{\bar{p}} \in \mathbb{R}^{N_{\bar{p}} \times N_{\bar{p}}}}{\text{minimize}} \quad & \|\mathbf{X}_{\bar{p}}^\mathsf{T}\mathbf{Y}^{(\bar{p})} - \bar{\mathbf{H}}^{(\bar{p})}\|_F^2 \\ \text{s.t.} \quad & \mathbf{X}_{\bar{p}}^\mathsf{T}\mathbf{X}_{\bar{p}} = \mathbf{I}_{\bar{p}}, \mathbf{x}_{\bar{p}_1} = \mathbf{1}_{N_{\bar{p}}}/\sqrt{N_{\bar{p}}}\ , \end{aligned} \quad (7)$$

where a solution can be derived from [17]: let $\tilde{\mathbf{X}}$ be any matrix that verifies the constraint of (7). Introducing the matrix $\mathbf{M} = (\tilde{\mathbf{X}}^\mathsf{T}\mathbf{Y}^{(\bar{p})}\bar{\mathbf{H}}^{(\bar{p})}_{1:,1:})$ the submatrix containing everything except the first column and the first row. Let $\mathbf{P}\mathbf{D}\mathbf{Q}^\mathsf{T}$ be the singular value decomposition of $\mathbf{M}$. The problem accepts the following closed form:

$$\mathbf{X}_{\bar{p}} = \tilde{\mathbf{X}} \begin{bmatrix} 1 & \mathbf{0}_{N_{\bar{p}}-1}^\mathsf{T} \\ \mathbf{0}_{N_{\bar{p}}-1} & \mathbf{P}\mathbf{Q}^\mathsf{T} \end{bmatrix}. \quad (8)$$

This can be computed in $\mathcal{O}(N_{\bar{p}}^3)$ leading to a complexity of $\mathcal{O}(\sum_p N_p^3)$ to compute all $\{\mathbf{X}_p\}_p$. Note than the use of graph products is particularly interesting in this context, since the standard complexity (when the full graph is not supposed to be a Cartesian graph) is $\mathcal{O}((\prod_{p=1}^P N_p)^3)$.

$\mathcal{H}$ computation. To compute $\mathcal{H}$ with fixed $\{\mathbf{X}_p\}_p$, we use the strategy proposed in [21]. By fixing all dimensions except the $\bar{p}$-th, the solution of the subproblem is obtained by sorting the rows of the matrix $\left(\mathcal{Y} \underset{p=1}{\overset{P}{\times}} \mathbf{X}_p^\mathsf{T}\right)^{(\bar{p})}$ by their ℓ_2-norm and then by selecting the $K_{\bar{p}}$ rows with larger norms (row/column-wise hard thresholding) [22]. The complexity of this sorting process is $\mathcal{O}(\prod_{p=1}^P N_p + N_{\bar{p}}\log(N_{\bar{p}}))$ for each $\bar{p}$. In the following, we will denote by $\mathcal{K}_p$ the set of K_p indexes corresponding to the selected rows. The matrix $\mathbf{X}_p(:, \mathcal{K}_p)$, is thus $\mathbf{X}_p$ where only the column indexed by $\mathcal{K}_p$ are kept.

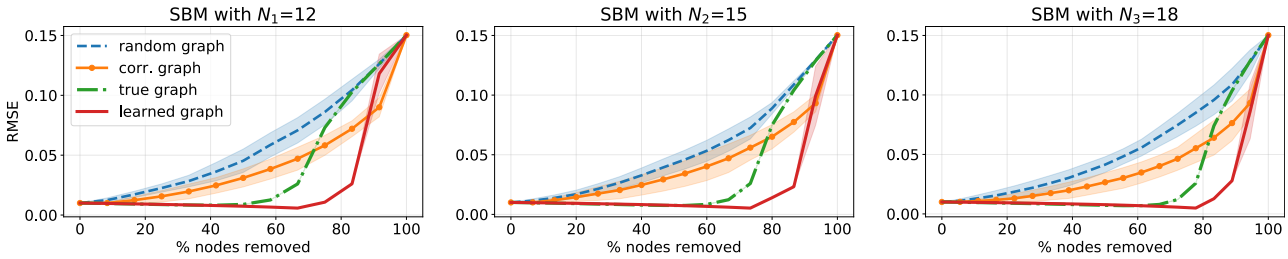

Fig. 1. Evolution of the RMSE ($\pm$ standard deviation) with the percentage of removed nodes for the three dimensions.

3.2. Selection of the best subset of nodes

In this second step, we want to choose the best subset of nodes. This choice is based on the frequency components kept in the previous step. Considering a graph from a graph product, selecting the best nodes of each graph is equivalent to selecting a subset of rows from each associated $\mathbf{X}_p$. To this end, we introduce the selection matrices $\{\mathbf{S}_p\}_{p=1}^P$, respectively in $\mathbb{R}^{K_p \times N_p}$, with elements satisfying: $\mathbf{S}_p(i,j) \in \{0,1\}, \forall i, \sum_j \mathbf{S}_p(i,j) = 1$, and $\forall j, \sum_i \mathbf{S}_p(i,j) \leq 1$ [5]. The sampled graph signal $\mathcal{Y}_s$ lies in $\mathbb{R}^{K_1 \times \cdots \times K_P}$ instead of $\mathbb{R}^{N_1 \times \cdots \times N_P}$ and is defined as $\mathcal{Y}_s = \mathcal{Y} \overset{P}{\underset{p=1}{\times}} \mathbf{S}_p$. $\mathcal{Y}$ can be approximated from $\mathcal{Y}_s$ according to the following equation:

$$\hat{\mathcal{Y}} = \mathcal{Y}_s \overset{P}{\underset{p=1}{\times}} \mathbf{X}_p(:,\mathcal{K}_p)(\mathbf{S}_p\mathbf{X}_p(:,\mathcal{K}_p))^{-1} \,. \tag{9}$$

The only thing left to do is to properly choose the $\{\mathbf{S}_p\}_p$. For each dimension, one related optimization problem is:

$$\underset{\mathbf{S}_p \in \mathbb{R}^{K_p \times N_p}}{\text{minimize}} \; \|\mathcal{Y} - \mathcal{Y}_s \overset{P}{\underset{p=1}{\times}} \mathbf{X}_p(:,\mathcal{K}_p)(\mathbf{S}_p\mathbf{X}_p(:,\mathcal{K}_p))^{-1}\|_F^2 \,. \tag{10}$$

This optimization program is computationally expensive because of the particular structure of $\mathbf{S}_p$. For multi-domain graphs, all methods in the literature for the selection of the rows are approximate: these iterative methods select one node per iteration in a greedy procedure. There are different existing works such as D-optimality[23] and frame potential [24]. Following these approaches, we use a low-complexity greedy algorithm to sample signals that reside on the nodes of a product graph [25].

3.3. Learning a Cartesian graph structure

Although the recovery of the underlying graph is not mandatory for subsampling, it can help to visualize the data in a compact form and to assess the relevance of the selected nodes in an application perspective. In this context, since the $\mathbf{X}_p$ have been computed in the optimization problem (5), the only missing information is linked to the eigenvalues $\mathbf{\Lambda}_p$. In the univariate case, a procedure for learning a Laplacian matrix from the graph Fourier basis and a sparse spectral representation has been introduced by Sardellitti et al. [16]. We propose here to learn a Cartesian graph with the $\{\mathbf{X}_p\}_p$ as eigenvectors, hence extending this strategy to the case of multidomain signals. For the multidomain framework, we cast this problem as:

$$\begin{aligned} \underset{\{\mathbf{L}_p\}_p, \{\mathbf{C}_p\}_p}{\text{minimize}} \; & \langle \mathcal{H}_\mathbf{K}, \sum_{p=1}^P \mathcal{H}_\mathbf{K} \times_p \mathbf{C}_p \rangle + \sum_{p=1}^P \mu_p \|\mathbf{L}_p\|_F \\ \text{s.t.} \quad & \mathbf{L}_p \in \mathcal{L}, \; \text{tr}(\mathbf{L}_p) = N_p \quad \forall p \\ & \mathbf{L}_p\mathbf{X}_p(:,\mathcal{K}_p) = \mathbf{X}_p(:,\mathcal{K}_p)\mathbf{C}_p, \mathbf{C}_p \succeq 0 \quad \forall p \,, \end{aligned} \tag{11}$$

where $\mathcal{H}_\mathbf{K}$ is the tensor $\mathcal{H}$ where the "non-selected rows" of each dimension are removed, and each $\mathbf{C}_p \in \mathbb{R}^{K_p \times K_p}$ is a matrix relative to the sparsity of $\mathcal{H}$. Note that, $\langle \mathcal{H}_\mathbf{K}, \sum_{p=1}^P \mathcal{H}_\mathbf{K} \times_p \mathbf{C}_p \rangle = \text{vec}(\mathcal{Y})^\mathsf{T} \bigoplus_{p=1}^P \mathbf{L}_p \text{vec}(\mathcal{Y})$, where $\text{vec}(\cdot)$ denotes the vectorization. Hence, we look for a Cartesian graph on which $\text{vec}(\mathcal{Y})$ is smooth. For more details on the optimization program and the additional matrices $\{\mathbf{C}_p\}_p$, the readers shall refer to the aforementioned paper.

We reformulate the first term of the objective function as:

$$\langle \mathcal{H}_\mathbf{K}, \sum_{p=1}^P \mathcal{H}_\mathbf{K} \times_p \mathbf{C}_p \rangle = \sum_{p=1}^P \text{tr}(\mathbf{H}_\mathbf{K}^{(p)^\mathsf{T}} \mathbf{C}_p \mathbf{H}_\mathbf{K}^{(p)}) \,, \tag{12}$$

where $\mathbf{H}_\mathbf{K}^{(p)}$ is the tensor $\mathcal{H}_\mathbf{K}$ unfolded on the dimension p. Thus, it turns out that because of the use of a Cartesian graph, the full objective function naturally splits into P separable functions i.e. P separable optimization problems for $(\mathbf{L}_{\bar{p}}, \mathbf{C}_{\bar{p}})$ with fixed $\{(\mathbf{L}_p, \mathbf{C}_p)\}_{p \neq \bar{p}}$. Therefore, we can simply solve each independent problem:

$$\begin{aligned} \underset{\mathbf{L}_{\bar{p}}, \mathbf{C}_{\bar{p}} \in \mathbb{R}^{K_{\bar{p}} \times K_{\bar{p}}}}{\text{minimize}} \; & \text{tr}(\mathbf{H}_\mathbf{K}^{(\bar{p})^\mathsf{T}} \mathbf{C}_{\bar{p}} \mathbf{H}_\mathbf{K}^{(\bar{p})}) + \mu \|\mathbf{L}_{\bar{p}}\|_F^2 \\ \text{s.t.} \quad & \mathbf{L}_{\bar{p}} \in \mathcal{L}, \; \text{tr}(\mathbf{L}_{\bar{p}}) = N_{\bar{p}} \\ & \mathbf{L}_{\bar{p}}\mathbf{X}_{\bar{p}}(:,\mathcal{K}_{\bar{p}}) = \mathbf{X}_{\bar{p}}(:,\mathcal{K}_{\bar{p}})\mathbf{C}_{\bar{p}}, \mathbf{C}_{\bar{p}} \succeq 0 \,. \end{aligned} \tag{13}$$

This last formulation is the same as in Sardellitti et al. [16] (univariate case) where they proposed to solve it with CVXPY [26].

4. EXPERIMENTS

Our method is now tested on synthetic and real data and compared to other non-adaptive sampling strategies. For each simulation, we compute the Root-Mean-Square-Error (RMSE) of the reconstruction as a function of the percentage of removed nodes. The code to reproduce all the subsequent results is available at https://github.com/Tgnassou/adaptive_subsampling_multidomain_graph.

4.1. Synthetic data

We consider a Cartesian graph constructed from three sub-graphs G_1, G_2, and G_3 of different sizes. As the bandlimited property is relative to the clusters of a graph [16], we generate them using the Stochastic Block Model (SBM), one of the most widely studied generative model that exhibits community structure [27]. The sizes of the clusters, and the edge densities connecting clusters are parameters of SBM. In this experiment, each graph is composed of 3 clusters of respectively 4, 5 and 6 nodes. Dimensions of the three graphs are thus respectively $N_1 = 12$, $N_2 = 15$ and $N_3 = 18$. Two nodes of the same group have a probability of 0.9 to be connected together and two nodes of different group have a probability of 0.2 to be connected together. Then, we generate noisy compressed bandlimited signals. They are known to provide more flexibility for the low-frequency component, hence reflecting better the data structure exhibited in several application contexts [7]. Graph signals are generated as $\mathcal{Y} = \mathcal{H} \times_1 \mathbf{X}_1 \times_2 \mathbf{X}_2 \times_3 \mathbf{X}_3 + \mathcal{E}$ where $\mathcal{E} \sim \mathcal{N}(0, 0.01)$ and $\mathcal{H}(k_1, k_2, k_3) \sim \mathcal{N}(1, 0.5)$ for all $k_1 \leq K_1 = 3$, $k_2 \leq K_2 = 3$

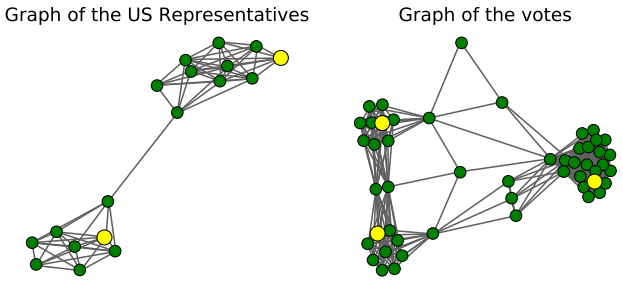

Fig. 2. Learned Graphs of the representatives and the votes, where the larger yellow nodes are the selected nodes.

and $k_3 \leq K_3 = 3$. For the others indexes, $\mathcal{H}(k_1, k_2, k_3)$ is equal to the outer product of three vectors $\mathbf{w}_1$,$\mathbf{w}_2$ and $\mathbf{w}_3$, where $\mathbf{w}_i(k_i) = 1$ for all $k_i \leq K_i$ and $\mathbf{w}_i(k_i) = (K_i/k_i)^{2\beta}$ for all $k_i > K_i$ with $\beta = 2$ and $i = \{1, 2, 3\}$. With this construction, the values on frequencies upper than (K_1, K_2, K_3) drastically decrease to zero.

Method. We compare the reconstruction performances to those obtained by a correlation graph, the real graph (that was used for data generation), and a random graph (Erdős-Rényi). The quality of the reconstruction is measured by the RMSE between the learned signal and the noiseless real signal $(\mathcal{Y} - \mathcal{E})$. For each method, we reconstruct the signal with a fixed number of nodes and compare the RMSE performances. For sake of clarity, we display the results on Figure 1 dimension per dimension. Note that when the graph is known (which is the case for the three competitor methods), the optimization problem (5) reduces to the learning of $\mathcal{H}$.

Results. Evolution of the RMSE with respect to the percentage of removed nodes is displayed in Figure 1 (average on 10 simulations). The importance of the structure of the graph clearly appears. The random graph and the correlation graph do not allow the suppression of nodes since they do not capture the structure of the data. For these graphs, the RMSE grows linearly with the percentage of removed nodes. On the other hand, our adaptive procedure efficiently learns a graph structure that is appropriate for efficient subsampling. Thanks to the learned graph, it is possible to remove more than 60% and still have a good reconstruction with a RMSE close to zero. In fact, for each dimension our learned graph allows to select only 3 nodes (reflecting the 3 clusters in the graphs and the K_i parameters), and still have a RMSE equals to 0.03. These curves underline another surprising property: the RMSE obtained with our graph is sometimes actually better than the one obtained with the true graph. This may be due to the fact that the bandlimitness constraint used in our method does not necessary learns the original graph but rather a graph structure that is optimised for sampling. Note that during our experiments, we also tried with other graphs such as Erdős-Rényi or Barabási graphs and we obtained similar results.

4.2. Real data

Illinois House of Representation. In this part, we measure the effectiveness of our method on a real dataset consisting on the votes of the Illinois House of Representatives during 2015-2017. This dataset is available at voteview.com. There are $N_1 = 18$ seats at the Illinois House of representatives, one for each district. Each seat corresponds to a US Representative belonging either to The Democratic or The Republican parties. In our study, we used the $N_2 = 50$ first available votes. Each vote is represented by a vector of size 18 with values +1 (*Yes*) and -1 (*No*).

In this case we have two dimensions, one for the US Representatives, another for the votes. We use our algorithm to compute the Fourier basis, select the nodes and learn the graph structure. Figure (2) shows the two graphs (one per dimension). Two clusters appear in the US representatives graph corresponding to Democrats and Republicans. For the votes graph, three clusters appear corresponding to questions with the same answer distribution (e.g. questions where all Republicans answer *Yes* and all Democrats answer *No*). As expected, we see that the selected nodes (in yellow) are coherent because we selected a node in each cluster: two nodes for the first graph and three nodes for the second graph.

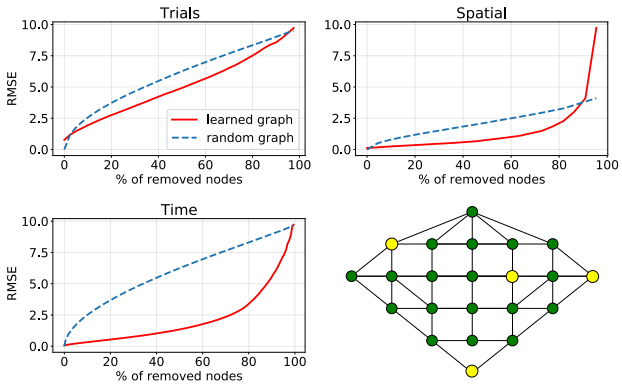

Fig. 3. Evolution of the RMSE with the percentage of removed nodes for the dimension of trials, the spatial dimension and the time dimension (top and bottom right). Electrodes learned graph (bottom right). In yellow, the four selected nodes/channels.

Brain Computer Interface. This data set contains ElectroEncephaloGram data (EEG) recorded during a Brain Computer Interface (BCI) experiment consisting on 4 different motor imagery: movement of left hand, movement of right hand, movement of both feet, and movement of tongue. Each action is imagined and repeated 10 times (leading to $N_1 = 4 \cdot 10 = 40$ trials). The EEG is recorded with $N_2 = 22$ electrodes and each time signal contains $N_3 = 188$ samples. For a much more complete description of the data set see [28]. The signal is therefore a three dimensional tensor.

Figure 3 displays the evolution of the RMSE as a function of the percentage of removed nodes for each dimension for our method and for a random graph (naive sampling). The sampling performances differ according to the dimension of interest (trials, space/electrodes, time samples). For the trial dimension, subsampling is tricky since as soon as a node is removed, the reconstruction is largely degraded, which is probably due to the fact that we work on the raw signals that do not allow to correctly represent the different types of movements. In contrast, it is possible to remove 82% of the electrodes and 75% of the time samples and to obtain a global RMSE of 2.8 (which corresponds to the inflexion point on the plots). The graph of electrodes learned in this configuration is displayed on the bottom right of Figure 3. The nodes which are selected for sampling are in yellow.

5. CONCLUSION

In this paper, we proposed a method to adaptively sample multidomain signals. By merging results from GSP, tensors, and graph products, we showed that it is possible to learn a graph structure that allows sampling and a useful data modelling and visualization. Using the learned structure we were able to deduce the number of nodes that could be deleted, select the best remaining nodes and accurately reconstruct the graph signal. This method also offers nice perspectives for the processing of datasets of N multidomain signals or size $N_1 \times \ldots \times N_P$. Indeed, by constructing a new tensor of dimension $N \times N_1 \times \ldots \times N_P$ (where the first dimension of the tensor serves to store the different signals), our algorithm can directly be used.

6. REFERENCES

[1] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre 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] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[3] Aliaksei Sandryhaila and José MF Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[4] Siheng Chen, Aliaksei Sandryhaila, José MF Moura, and Jelena Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4609–4624, 2015.

[5] Antonio G Marques, Santiago Segarra, Geert Leus, and Alejandro Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1832–1843, 2016.

[6] Siheng Chen, Rohan Varma, Aliaksei Sandryhaila, and Jelena Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Transactions on Signal Processing*, vol. 63, no. 24, pp. 6510–6523, 2015.

[7] Siheng Chen, Rohan Varma, Aarti Singh, and Jelena Kovacević, "Signal recovery on graphs: Random versus experimentally designed sampling," in *International Conference on Sampling Theory and Applications (SampTA)*. IEEE, 2015, pp. 337–341.

[8] Mikhail Tsitsvero, Sergio Barbarossa, and Paolo Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, 2016.

[9] Han Shomorony and A Salman Avestimehr, "Sampling large data on graphs," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2014, pp. 933–936.

[10] Aliaksei Sandryhaila and Jose MF Moura, "Big data analysis with signal processing on graphs," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, 2014.

[11] Rohan Varma and Jelena Kovačević, "Random sampling for bandlimited signals on product graphs," in *13th International conference on Sampling Theory and Applications (SampTA)*. IEEE, 2019, pp. 1–5.

[12] Sai Kiran Kadambari and Sundeep Prabhakar Chepuri, "Learning product graphs from multidomain signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5665–5669.

[13] Muhammad Asad Lodhi and Waheed U Bajwa, "Learning product graphs underlying smooth graph signals," *arXiv preprint arXiv:2002.11277*, 2020.

[14] Xiaowen Dong, Dorina Thanou, Michael Rabbat, and Pascal Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, 2019.

[15] Vassilis Kalofolias, "How to learn a graph from smooth signals," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016, pp. 920–929.

[16] Stefania Sardellitti, Sergio Barbarossa, and Paolo Di Lorenzo, "Graph topology inference based on sparsifying transform learning," *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1712–1727, 2019.

[17] Pierre Humbert, Batiste Le Bars, Laurent Oudre, Argyris Kalogeratos, and Nicolas Vayatis, "Learning laplacian matrix from graph signals with sparse spectral representation," https://www.researchgate.net/publication/339553098_Learning_Laplacian_Matrix_from_Graph_Signals_with_Sparse_Spectral_Representation, 2019.

[18] Batiste Le Bars, Pierre Humbert, Laurent Oudre, and Argyris Kalogeratos, "Learning laplacian matrix from bandlimited graph signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2937–2941.

[19] Tamara G Kolda and Brett W Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[20] Richard Hammack, Wilfried Imrich, and Sandi Klavžar, *Handbook of product graphs*, CRC press, 2011.

[21] Pierre Humbert, Laurent Oudre, and Nicolas Vayatis, "Subsampling of multivariate time-vertex graph signals," in *27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[22] Richard G Baraniuk, Volkan Cevher, Marco F Duarte, and Chinmay Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[23] Manohar Shamaiah, Siddhartha Banerjee, and Haris Vikalo, "Greedy sensor selection: Leveraging submodularity," in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 2572–2577.

[24] Andreas Krause, Ajit Singh, and Carlos Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *The Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[25] Guillermo Ortiz-Jiménez, Mario Coutino, Sundeep Prabhakar Chepuri, and Geert Leus, "Sparse sampling for inverse problems with tensors," *IEEE Transactions on Signal Processing*, vol. 67, no. 12, pp. 3272–3286, 2019.

[26] Steven Diamond and Stephen Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[27] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983.

[28] Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, and Gert Pfurtscheller, "BCI competition 2008–Graz data set A," *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, vol. 16, pp. 1–6, 2008.