Subsampling of Multivariate Time-Vertex Graph Signals

Pierre Humbert, *Student Member, IEEE,*
CMLA (CNRS, ENS Paris Saclay)
Paris, France
pierre.humbert@cmla.ens-cachan.fr

Laurent Oudre
L2TI (Université Paris 13)
Villetaneuse, France
laurent.oudre@univ-paris13.fr

Nicolas Vayatis
CMLA (CNRS, ENS Paris Saclay)
Cachan, France
nicolas.vayatis@cmla.ens-cachan.fr

Abstract—**This article presents a new approach for processing and subsampling multivariate time-vertex graph signals. The main idea is to model the relationships within each dimension (time, space, feature space) with different graphs and to merge these structures. A new technique based on tensor formalism is provided, which aims to identify the frequency support of the graph signal in order to preserve its content after subsampling. Results are provided on real EEG data for data interpolation and reconstruction.**

Index Terms—**Graph Signal Processing (GSP), sampling over graphs, tensors**

I. Introduction

Graph Signal Processing (GSP) [1], [2] has emerged as a new field for dealing with structured data, recorded for instance with sensor networks. By assuming that each sample of the signal lies on a graph node, complex spatial interactions or dependencies can be taken into account for several tasks such as sampling, filtering or reconstruction [3], [4], [5]. 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 node.

In most works, 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 an underlying graph structure. Therefore, very often the time variations are not taken into account in the processing of such signals: studies consider either one time-sample [6], [7], [8], where only the spatial dimension is analyzed, or an average on a time window. In order to deal with temporal graph signals, recent works have introduced the notion of *time-vertex* signal processing, where both spatial and temporal interactions are modeled [9]. In this context, a new Graph Fourier Transform (GFT), called Joint Fourier Transform (JFT) has been introduced and efficiently used in several examples such as video inpainting, seismic epicenter localization [9], and spread modelling [10].

In the frequent case of multivariate sensor networks or feature-based representations, one solution may consist in treating each feature or modality individually. However, the underlying assumption is that all variables are independent, which is not true in several typical situations, such as meteorological data which is composed of several correlated variables or features (atmospheric pressure, temperature, rainfalls) over time and space. For these reasons, a third graph layer is needed in order to also model the links between the different modalities. As a result, multivariate time-vertex graph signals should be modeled with three types of interactions in time, space and feature space. By building on the notion of graph product and on the tensor formalism, we show in this article that it is possible to extend the notion of GFT to these signals, and thus to provide efficient algorithms for processing them.

This article provides a framework for processing multivariate time-vertex graph signals, based on the notion of graph product and the definition of three graphs that each model the interactions within one dimension (time, space, feature space). By using the tensor formalism, several sparsity methods are provided, that can be specified so as to work only on one dimension (i.e. selection of the best time samples, sensors or features). These approaches are tested on real EEG signals in order to assess the sampling and interpolation performances of the proposed framework.

II. Background and notations

We first introduce the notations used in this paper on tensor algebra and graph product (see [11], [12] for a more complete introduction).

A. Tensor algebra

Let $d_1, d_2, \ldots, d_p \in \mathbb{N}_*$ and $\mathbb{Y} = \mathbb{R}^{d_1} \times \ldots \times \mathbb{R}^{d_p} \triangleq \mathbb{R}^{d_1 \times \cdots \times d_p}$ be the product of p $\mathbb{R}$-vector spaces. An element of $\boldsymbol{\mathcal{Y}} \in \mathbb{Y}$ is called a tensor of order p. In the following, $\boldsymbol{\mathcal{Y}}$ will be used indifferently to denote the multilinear form in $\mathbb{Y}^*$ and its representation in the canonic base of $\mathbb{Y}$, the choice being clear from the context. The *mode-m matrix product* (see Figure 1) between a tensor $\boldsymbol{\mathcal{Y}}$ and a matrix $\boldsymbol{X} \in \mathbb{R}^{j \times d_m}$ in coordinate notation is $(\boldsymbol{\mathcal{Y}} \times_m \boldsymbol{X})_{i_1, \cdots, i_{m-1}, j, i_{m-1}, \cdots, i_p} \triangleq \sum_{k=1}^{d_m} \boldsymbol{\mathcal{Y}}_{i_1, \cdots, i_{m-1}, k, i_{m-1}, \cdots, n} \boldsymbol{X}_{j,k}$ and is equivalent to $\boldsymbol{\mathcal{Y}} \times_m \boldsymbol{X} \Leftrightarrow \boldsymbol{X} \boldsymbol{\mathcal{Y}}^{(m)}$ where $\boldsymbol{\mathcal{Y}}^{(m)}$ denotes the tensor $\boldsymbol{\mathcal{Y}}$ unfolded along axis m. The operator $\otimes$ represent the Kronecker product. When multiple products are necessary, we use the upper version of these notations, $\bigtimes$ and $\bigotimes$.

B. Product graph

Let a weighted graph $G = (\boldsymbol{\mathcal{V}}, \mathcal{E})$ with nodes $\boldsymbol{\mathcal{V}} = \{1, \cdots, N\}$, 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

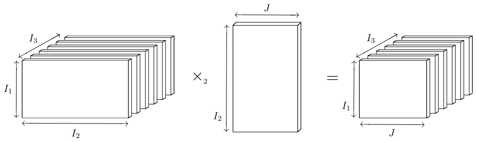

Fig. 1: Illustration of the mode-1 product.

weight matrix. For simplicity of the discussion, assume that $\boldsymbol{\mathcal{L}}$ is diagonalizable and its eigendecomposition is $\boldsymbol{\mathcal{L}} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$, with $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \dots, \lambda_N)$ a diagonal matrix with the eigenvalues and $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)$ a matrix with the eigenvectors as columns. If $\boldsymbol{\mathcal{L}}$ is not diagonalizable, Jordan decomposition into generalized eigenvectors is used.

Let $G_1 = (\boldsymbol{\mathcal{V}}_1, \mathcal{E}_1)$ and $G_2 = (\boldsymbol{\mathcal{V}}_2, \mathcal{E}_2)$ be two graphs with N_1 and N_2 vertices and Laplacian $\boldsymbol{\mathcal{L}}_1 = \boldsymbol{X}_1\boldsymbol{\Lambda}_1\boldsymbol{X}_1^{-1}$, $\boldsymbol{\mathcal{L}}_2 = \boldsymbol{X}_2\boldsymbol{\Lambda}_2\boldsymbol{X}_2^{-1}$, respectively. A product graph of G_1 and G_2 denoted with the symbol $\diamond$, is the graph with Laplacian operator related to the eigenvalue decompositions of each Laplacian matrix through

$$\boldsymbol{\mathcal{L}} = (\boldsymbol{X}_1 \otimes \boldsymbol{X}_2)\boldsymbol{\Lambda}_\diamond(\boldsymbol{X}_1^{-1} \otimes \boldsymbol{X}_2^{-1}), \tag{1}$$

where $\boldsymbol{\Lambda}_\diamond$ depends of the choice product [13].

C. Graph signal processing

A graph signal can be represented as a vector $\boldsymbol{Y} \in \mathbb{R}^{N_1 \times N_2}$, where $Y_{i,j}$ is the function value at the i-th node of G_1 and j-th node of $G_2^{(i)}$. Using the Graph Fourier Transform (GFT) it is possible to create a spectral representation $\boldsymbol{H}$ for $\boldsymbol{Y}$ defined as

$$\boldsymbol{H} = \boldsymbol{Y} \times_1 \boldsymbol{X}_1^{-1} \times_2 \boldsymbol{X}_2^{-1}. \tag{2}$$

The eigenvalues can be interpreted as distinct frequencies, the components of $\boldsymbol{H}$ as Fourier coefficients, and the eigenvectors as a decomposition basis. Notice that if G_1 is a cycle graph, $\boldsymbol{X}_1$ is the Discrete Fourier Transform matrix, and the GFT formula (2) is exactly the JFT. Hence, the JFT could be seen as a particular case of the multidimensional GFT.

With this formalism, it is straightforward to extend the previous definitions to product graph with more than two related graphs. Given a collection of M graphs $(G_m)_{m=1}^M$ with $(N_m)_{m=1}^M$ vertices and Laplacian $(\boldsymbol{\mathcal{L}}_m = \boldsymbol{X}_m\boldsymbol{\Lambda}_m\boldsymbol{X}_m^{-1})_{m=1}^M$, the Laplacian of the (full) product graph is

$$\boldsymbol{\mathcal{L}} = (\bigotimes_{m=1}^{M} \boldsymbol{X}_m)\boldsymbol{\Lambda}_\diamond(\bigotimes_{m=1}^{M} \boldsymbol{X}_m^{-1}), \tag{3}$$

where $\boldsymbol{\Lambda}_\diamond$ is a matrix which depends of the choice product. As an example, if we choose the cartesian product, it is well known that $\boldsymbol{\Lambda}_\diamond = \bigoplus_{m=1}^{M} \boldsymbol{\Lambda}_m$ where $\bigoplus$ is the kronecker sum [14].

The GFT of a graph signal tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{N_1 \times \cdots \times N_M}$ is therefore

$$\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{Y}} \bigtimes_{m=1}^{M} \boldsymbol{X}_m^{-1}. \tag{4}$$

III. Method

In this section we propose to use the tensor framework to represent multivariate time-vertex graph signals. By using the extended version of GFT, we propose a subsampling technique that aims at recovering the whole signals by using a subset of time samples, sensors or components.

A. Framework for processing multivariate time-vertex graph signals

In the context of graph signals obtained from multivariate sensor networks, data can be stored in a tensor $\boldsymbol{\mathcal{Y}}$ in $\mathbb{R}^{F \times T \times S}$ where F is the number of features recorded by the sensor, T is the number of time samples, and S is the number of sensors. Interactions between the different dimensions can be modelled with three different graphs that each encodes the interactions for one dimension:

- G_F – This graph quantifies the similarity between the different features or modalities of the data. There are several techniques to build such a graph. An intuitive approach is to consider a weighted correlation graph where the weights between two nodes corresponds to the absolute Pearson correlation coefficient between the modalities or features. This graph has been widely used for studying correlated features in genomics such as gene expression [15].
- G_T – This graph controls the interactions between time samples. One common choice is to use a directed cycle graph of size T, which links each sample to the next sample. This type of dependencies can be seen as a Markov process where the value of a sample only depends on the previous sample. This graph is widely used in the GSP community since, for this graph, the definition of the classical Fourier Transform is the same than the Graph Fourier Transform. The weight (adjacency) matrix of G_T is a circulant matrix and is known to have as eigenvector matrice, the discrete Fourier transform matrix, and as t-th eigenvalue, $\boldsymbol{\lambda}_t = \exp(2\pi i(t-1)(T-1))/T))$ [9], [16].
- G_S – In a sensor network, this graph models the interactions between the sensors. When dealing with a physical network, this graph can be based on physical links that exist between the sensors. When these interactions are unknown, an intuitive choice consists in building the graph in order to reflect the spatial closeness of each sensor. In general case, this graph is undirected and the edge weights can be built with the Gaussian function

  $$\boldsymbol{W}_S(i,j) = \exp\Big(-\|s_i - s_j\|_2^2/\sigma^2\Big), \tag{5}$$

  where s_i is the spatial position of the i-th node of G_S. In the literature, this is one of the most common practices to construct edge weights from given data [17].

Notice that since G_F and G_S are undirected, with no self-loops, and with a single connected component, their Laplacian are symmetric positive semi-definite with $\boldsymbol{X}_F^{-1} = \boldsymbol{X}_F^\intercal$, and $\boldsymbol{X}_S^{-1} = \boldsymbol{X}_S^\intercal$.

B. Identifying the support of the tensor graph signal

Most graph signal subsampling techniques are based on the assumption that the signal representation in the GFT domain is sparse [18], [19], [20]. A graph signal with this property is called bandlimited with respect to its graph. When the frequency support of a graph signal is not known, we need to identify it in order to design a proper sampling and interpolation procedure. This problem leads to the following sparse signal reconstruction minimization

$$\min_{\boldsymbol{\mathcal{H}}} \quad \|\boldsymbol{\mathcal{Y}} \bigtimes_{m=1}^{M} \boldsymbol{X}_m^{-1} - \boldsymbol{\mathcal{H}}\|_F^2 + \Omega(\boldsymbol{\mathcal{H}}), \tag{6}$$

where Ω is a regularization function imposing some sparsity on $\boldsymbol{\mathcal{H}}$. There are several valid choices for Ω. However, to obtain bandlimited tensor graph signal, we need to design a function which imposes sparsity on slices. We propose to use the two following functions, illustrated on Figure 2:

1) **General Sparsity (GS) constraint:**

$$\Omega : (\boldsymbol{\mathcal{H}}, \lambda) \longmapsto \lambda \|\boldsymbol{\mathcal{H}}\|_0. \tag{7}$$

This function is the equivalent of the vectorial zero semi-norm for tensor object. When using this "norm", the solution of (6) is given by the hard-threshold operator P_λ

$$\boldsymbol{\mathcal{H}}^* = P_\lambda\Big(\boldsymbol{\mathcal{Y}} \bigtimes_{m=1}^{M} \boldsymbol{X}_m^{-1}\Big). \tag{8}$$

The complexity of the sorting process is $\mathcal{O}(\prod_{k=1}^{M} N_k \log(\prod_{k=1}^{M} N_k))$. Although this sparsity constraint is very simple to implement, it does not allow us to control in which dimension the sparsity occurs. In particular, this behaviour is not adapted to the bandlimitedness assumption.

2) **Controlled Sparsity (CS) constraint:**

$$\Omega : (\boldsymbol{\mathcal{H}}, (\lambda_m)_{m=1}^M) \longmapsto \sum_{m=1}^{M} \lambda_m \|\boldsymbol{\mathcal{H}}^{(m)}\|_{2,0}. \tag{9}$$

This function imposes zeros on the rows of the unfolding $\boldsymbol{\mathcal{H}}$ which make it more adapted for the bandlimitedness assumption (see Fig. 2). Considering each norm independently, the solution of the subproblem is obtained by sorting the rows of $\big(\boldsymbol{\mathcal{Y}} \bigtimes_{m=1}^{M} \boldsymbol{X}_m^{-1}\big)^{(m)}$ by their ℓ_2-norm and then selecting the rows with norms lower than λ_m (row/column-wise hard thresholding) [21]. Denoting by $N_{m^-} = \prod_{k=1, k\neq m}^{M} N_k$, the complexity of this sorting process is $\mathcal{O}(N_{m^-} N_m + N_{m^-} \log(N_{m^-}))$. Following this observation, we propose the following optimization problem and the algorithm 1 to solve it

$$\min_{\boldsymbol{\mathcal{H}}} \quad \|\boldsymbol{\mathcal{Y}} \bigtimes_{m=1}^{M} \boldsymbol{X}_m^{-1} - \boldsymbol{\mathcal{H}}\|_F^2 \tag{10}$$

$$\text{s.t.} \qquad \Big(\|\boldsymbol{\mathcal{H}}^{(m)}\|_{2,0} \leq K_m\Big)_{m=1}^{M}, \tag{11}$$

where each $K_m \in \mathbb{R}$ control the sparsity of the m-th dimension. Contrary to the previous constraint, this one is therefore adapted to the bandlimited assumption; thanks to the parameters K_m it is possible to impose different sparsity constraints for the three different domains (time, space, feature space).

Algorithm 1 CS constraint

```
Input : $\boldsymbol{\mathcal{Y}}$, $(\boldsymbol{L}_m)_m^M$, and $(K_m)_m^M$.
Output : $\boldsymbol{\mathcal{H}}$
for $m = 1, \cdots, M$ do
    $\boldsymbol{X}_m \longleftarrow$ eigen$(\boldsymbol{L}_m)$
end for
$\boldsymbol{\mathcal{H}} \longleftarrow \boldsymbol{\mathcal{Y}} \bigtimes_{m=1}^{M} \boldsymbol{X}_m^{-1}$
for $m = 1, \cdots, M$ do
    for $j = 1, \cdots, N_{m^-}$ do
        $\boldsymbol{y}_j \longleftarrow \|\boldsymbol{\mathcal{H}}_{:,j}^{(m)}\|_F^2$
    end for
    $s \longleftarrow$ argsort$(\boldsymbol{y})[0 : K_m]$
    $\boldsymbol{\mathcal{H}}^{(m)}[s] \longleftarrow 0$
end for
```

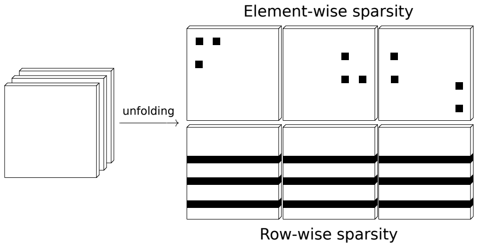

Fig. 2: Illustration of the difference between the two sparsity norms. On the top, the element-wise sparsity norm. On the bottom, the row-wise sparsity norm.

C. Selecting the best nodes

The sparsity in the frequency domain allows to subsample graph signals by selecting few elements from each graph domain: this task is often referred to as sparse sampling. Sampling a subset of nodes from multiple graph $(G_m)_{m=1}^M$ is equivalent to selecting a subset of rows and columns from each associated $\boldsymbol{X}_m$. Fortunately, as the support of the tensor graph signal is known (see previous section), the columns which need to be kept are known and we only need to select the best subset of rows for each $\boldsymbol{X}_m$. When only one graph is considered, several methods exist in order to efficiently find a proper subset. For high-dimensional data, greedy methods (algorithms that select one node at a time) are very useful. Several authors have proved submodularity of different optimality criteria such as D-optimality [22], and frame potential [23]. Following theses approaches, we use a low-complexity greedy algorithm to sample signals that reside on the vertices of a product graph [24].

IV. Results

In this section, we test our different subsampling strategies on real EEG data.

A. Data

Dataset: The dataset consists of $S = 32$ EEG signals collected at 250 Hz during a general anesthesia with electrodes

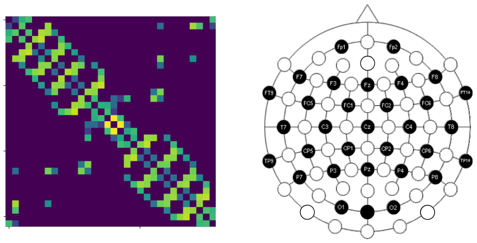

Fig. 3: Weight matrix of the graph G_S and template 2-D layouts of the sensors.

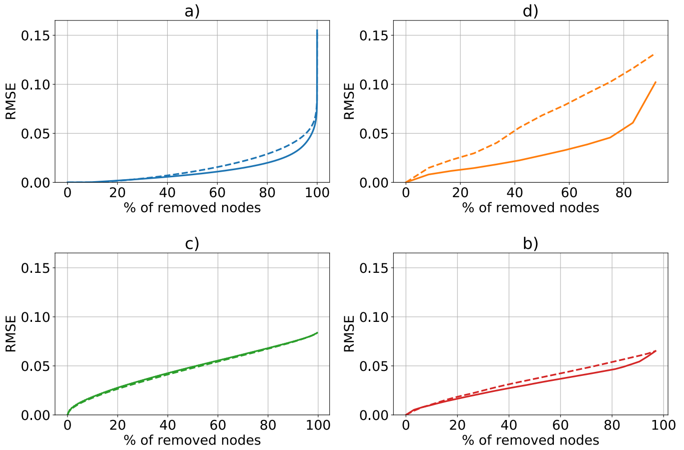

Fig. 4: Evolution of the RMSE with the percentage of removed nodes for a) General Sparsity (GS) or Controlled Sparsity (CS) on the b) feature space F c) time T d) spatial S dimensions. The dotted plots correspond to configurations where one or all graphs have been replaces by random ER graphs.

attached on the brain of a patient. For each EEG signal, we compute the spectrogram through Short-Time Fourier-Transform with time-windows of 256 samples and with 50% overlap. Then, we compute the energies in $F = 12$ frequency bands equally spaced between 0.1 Hz and 12 Hz (in order to retrieve the delta, theta and alpha waves that are relevant for anesthesia [25]). The final tensor graph signal $\boldsymbol{\mathcal{Y}}$ is in $\mathbb{R}^{F\times T\times S} = \mathbb{R}^{12\times 233\times 32}$.

Graph construction: As explained in the previous section, we construct three graphs, G_F, G_T, G_S, respectively as a weighted correlation graph, a cycle graph, and a spatially weighted graph. The graph G_S related to the spatial component is constructed with the spatial position of each channels in 3D space (Fig. 3).

B. Subsampling and reconstruction

The different subsampling techniques described in this article are tested on the EEG data. Then, for a given percentage of removed nodes, we reconstruct the data and compute the Root Mean Square Error (RMSE).

Results with four different sparsity constraints are displayed in solid lines in Figure 4:

- General Sparsity (GS) with a total of 74560 nodes. The conserved nodes can appear in any time/space/feature space positions.
- Controlled Sparsity (CS) on the feature space dimension F with a total of 12 nodes. Only a few energy signals are kept and other are reconstructed by using the correlations between the modalities.
- Controlled Sparsity (CS) on the time dimension T with a total of 233 nodes. Only a few time samples are kept for the reconstruction: this task is linked with signal interpolation and to the classical definition of signal subsampling.
- Controlled Sparsity (CS) on the spatial dimension S with a total of 32 nodes. Only a few EEG sensors are used to reconstruct others, based on their spatial interactions.

The performances of the General Sparsity (GS) constraint configuration are very satisfactory since it is possible to reconstruct the whole data set with a 0.02 RMSE by removing up to 80% of the nodes. This means that the data is actually very sparse in the frequency domain and that the information can be well represented in sparse domains. However, the main drawback of this approach is that, since the selected nodes can appear in any domain (time, space, feature space), such subsampling may be difficult to implement in practice.

The results obtained with Controlled Sparsity (CS) constraints are very contrasted. To obtain a 0.04 RMSE, it is equivalent to remove 30% of the time samples or 60% of the sensors or of the frequency bands. It therefore appears that the graph structure is especially relevant in the subsampling process for this two last dimensions. For the S and F dimensions, results are similar up to 50% of removed nodes but differ for larger percentages. Removing more sensors seems to have a stronger effect than removing more modalities (before 80%). This is probably due to the fact that the main phenomena occurring during anesthesia appear in the alpha band between 8 Hz and 12 Hz which spans several of the 10 considered frequency bands. Therefore, a strong correlation exists between modalities that enables a fairly good reconstruction. For the T dimension, the RMSE increases linearly with the number of removed nodes, which is probably due to the relatively weak interactions modeled in the G_T graph. Although the performances of the Controlled Sparsity configuration appear to be worse than the General Sparsity, it is interesting to notice that for this configuration, the subsampling experiment can directly be used to select sensors, lower the sampling frequency or to choose the relevant frequency band to monitor during anesthesia.

C. Importance of the graph structure

Intuitively, the structure of the graphs used for sampling and reconstruction is crucial. To prove this point, we propose in this experiment to replace one or all of the graphs G_F, G_T, G_S with a random Erdős–Rényi (ER) graph. For the General Sparsity (GS) constraint, all graphs are random and for the Controlled Sparsity (CS) constraint only the graph of interest is changed. The resulting reconstruction performances are displayed in dotted lines on Figure 4.

For the General Sparsity constraint, the performances decrease with the use of random graphs: for 80% of removed nodes, the RSME is now 0.03 instead of 0.02.

As far as the Controlled Sparsity constraints are concerned, and as seen in the previous subsection, the graph structure is especially important for the F and S dimensions. In particular, when considering the spatial dimension, the RMSE is significantly larger with the random graph, which shows that the proposed spatial modeling is here useful for the sampling/reconstruction process. Interestingly, although the directed cycle graph has been a very common model for dealing with the temporal aspects of time-vertex signals [9], [16], it here appears that this graph does not bring the necessary structure for the sampling task: results are here similar when this graph is replaced by a random graph. Instead of the simple Markov formulation, a more structured graph could have better modeled the relationships between time samples.

V. Conclusion

This paper proposed a novel subsampling framework for multivariate time-vertex graph signals based on tensor formalism. We imposed with three different graphs relationships between the three dimensions, time, space, and feature space. Then, to sample the tensor graph signal, we provided an efficient algorithm to identify its support. The results showed the importance of the graphs in this algorithm and support for the relevance of the controlled sparsity contraints to recover multivariate bandlimited signals.

References

[1] D. Shuman, S. 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. Djuric and C. Richard, Eds., *Cooperative and Graph Signal Processing – Principles and Applications*. Elsevier, 2018.

[3] A. 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. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

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

[6] 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.

[7] R. Jain, J. Moura, and C. 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.

[8] D. Mohan, M. 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.

[9] F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud, "A time-vertex signal processing framework: Scalable processing and meaningful representations for time-series on graphs," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 817–829, 2018.

[10] A. Loukas and N. Perraudin, "Stationary time-vertex signal processing," *arXiv preprint arXiv:1611.00255*, 2016.

[11] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[12] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[13] A. Sandryhaila and J. M. Moura, "Big data analysis with signal processing on graphs," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, 2014.

[14] R. Merris, "Laplacian graph eigenvectors," *Linear algebra and its applications*, vol. 278, no. 1-3, pp. 221–236, 1998.

[15] S. Horvath, *Weighted network analysis: applications in genomics and systems biology*. Springer Science & Business Media, 2011.

[16] A. Loukas and D. Foucard, "Frequency analysis of time-varying graph signals," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2016, pp. 346–350.

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

[18] 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.

[19] S. 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.

[20] 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.

[21] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *arXiv preprint arXiv:0808.3572*, 2008.

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

[23] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Transactions on signal processing*, vol. 62, no. 5, pp. 1135–1146, 2014.

[24] G. Ortiz-Jiménez, M. Coutino, S. P. Chepuri, and G. Leus, "Sampling and reconstruction of signals on product graphs," *arXiv preprint arXiv:1807.00145*, 2018.

[25] P. L. Purdon, E. T. Pierce, E. A. Mukamel, M. J. Prerau, J. L. Walsh, K. F. K. Wong, A. F. Salazar-Gomez, P. G. Harrell, A. L. Sampson, A. Cimenser *et al.*, "Electroencephalogram signatures of loss and recovery of consciousness from propofol," *Proceedings of the National Academy of Sciences*, vol. 110, no. 12, pp. E1142–E1151, 2013.