Sampling Graph Signals with Sparse Dictionary Representation

Kaiwen Zhang, Mario Coutino and Elvin Isufi
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology, Delft, The Netherlands

***Abstract*—Graph sampling strategies require the signal to be relatively sparse in an alternative domain, e.g. bandlimitedness for reconstructing the signal. When such a condition is violated or its approximation demands a large bandwidth, the reconstruction often comes with unsatisfactory results even with large samples. In this paper, we propose an alternative sampling strategy based on a type of overcomplete graph-based dictionary. The dictionary is built from graph filters and has demonstrated excellent sparse representations for graph signals. We recognize the proposed sampling problem as a coupling between support recovery of sparse signals and node selection. Thus, to approach the problem we propose a sampling procedure that alternates between these two. The former estimates the sparse support via orthogonal matching pursuit (OMP), which in turn enables the latter to build the sampling set selection through greedy algorithms. Numerical results corroborate the role of key parameters and the effectiveness of the proposed method.**

***Index Terms*—Compressive sensing, graph signal sampling, graph signal processing, signal reconstruction, sparse sensing**

I. Introduction

Sampling strategies are ubiquitous for graph signals over e.g. sensor, social, and biological networks [1], [2]; to name a few. Different from the conventional temporal and spatial sampling, graph sampling requires accounting for the coupling between the signal and the underlying topology. This coupling is often expressed as a prior to obtain a sparse signal representation in an alternative domain. The typical approach in these cases is to consider the eigenvectors of a graph representative matrix, e.g., Laplacian, and represent the graph signal as a linear combination of a few eigenvectors that capture most of its energy; i.e. bandlimited graph signal representation [1]–[3].

Studies have been devoted to the sampling of bandlimited graph signals. Earlier works, e.g. [4], [5], and [2], have developed conditions for the exact recovery of bandlimited data in the noiseless scenarios, laying down the theoretical foundations. In [6], the uncertainty principle was developed for graph signals. Compared to the noiseless case, sampling and recovering graph signals in the noisy settings are considerably more complex. To address this, a series of works have been proposed based on intuitions related to optimal experiment design [1]. For instance, convex relaxation techniques have been used in [7], and greedy algorithms were also widely adopted as alternatives, e.g. [2], [8]. Furthermore, efforts have been put to facilitate sampling over large graphs. To avoid the eigendecomposition cost when working with bandlimited graph signals, the work in [9] considered the signal to be smooth and used Gershgorin discs to optimize a sampling criterion based on the smallest eigenvalue bound of the graph Laplacian. In [10], the author proposed to use the so-called spectral proxies as an alternative to the graph frequencies thus avoiding the computation of the eigenvectors. In addition, other types of sampling sampling methods have been considered. The aggregation sampling was proposed in [11] and built on the fact that each node has access to shifted versions of the signal. Probabilistic sampling has also been introduced to reduce the associated computational burden when sampling signals on large graphs [12], [13].

While the approximately-bandlimited assumption is often a safe choice for sampling graph signals, it often leads to non-sparse representations, ultimately, requiring a large number of samples to reconstruct the signal within a prescribed accuracy [10]. When a graph signal is not bandlimited, approximating it as such may lead to reconstruction artifacts that are difficult to mitigate even if almost all nodes are sampled. To still be able to sample graph signals in such situations, we propose a novel dictionary-based graph sampling framework that represents the graph signal as a sparse combination of atoms of a parametric graph dictionary [14]. Since this setting works with an underdetermined system of equations – this is contrarily to the overdetermined case of bandlimited graph signal sampling that the above works consider – we exploit the sparsity of the signal and introduce a combinatorial ℓ_0-minimization problem for jointly optimizing the sampling set and the signal sparse representation. By identifying the two subproblems in this task: (i) sparse recovery and (ii) subset selection, we devise an efficient approach to find a tractable solution for the sampling problem. Numerical experiments corroborate the proposed method and show it outperforms bandlimited sampling at moderate signal-to-noise ratio (SNR) and number of samples.

II. Preliminaries

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of N nodes and M edges respectively, and $\mathbf{W}$ is the weighted adjacency matrix. The interconnections between nodes are captured by the entries of a symmetric matrix $\mathbf{S}$ known as the graph shift operator (GSO) [15], whose off-diagonal entry $[\mathbf{S}]_{i,j} > 0$ if there exists an edge connecting tuple (i, j), and $[\mathbf{S}]_{i,j} = 0$ otherwise. Throughout the paper, we assume $\mathbf{S}$ is normal and has the eigenvalue decomposition $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\top}$. Choices for the GSO are the adjacency matrix $\mathbf{W}$, the graph Laplacian $\mathbf{L}$, and the normalized Laplacian $\mathbf{L}_{\text{n}}$

To each node i a signal $x_i \in \mathbb{R}$ is associated, and the graph signal vector $\mathbf{x} = [x_1, \ldots, x_N]^{\top}$ collects the values of all nodes. As for temporal signals, filters are the tools to process graph signals [16]. Graph filters are functions of the GSO, i.e. $\mathbf{H}(\mathbf{S}) = h(\mathbf{S})$, and the filtering is performed by multiplying the graph filtering matrix $\mathbf{H}(\mathbf{S})$ with the graph signal $\mathbf{x}$, namely $\mathbf{x}_{\text{filt}} = \mathbf{H}(\mathbf{S})\mathbf{x}$. Specifically, the finite impulse response (FIR) graph filter is defined as matrix polynomials of

E-mails: {kzhang10; m.a.coutinominguez; e.isufi-1}@tudelft.nl; Mario Coutino is partially supported by CONACYT.

the GSO, i.e. $\mathbf{H}(\mathbf{S}) = \sum_{k=0}^{K} h_k \mathbf{S}^k$, where K is the filter order and $\mathbf{h} = [h_0, ..., h_K]^\top$ are the filter coefficients. The output of this FIR graph filter

$$\mathbf{x}_{\text{filt}} = \sum_{k=0}^{K} h_k \mathbf{S}^k \mathbf{x} \tag{1}$$

is the weighted sum of the K-hop neighbours' shifted signals $\{\mathbf{S}^k\mathbf{x}\}_k$ around the nodes. Thus, the FIR filter captures the local behaviors of graph signals for up to a radius K from a node.

A parametric graph dictionary (PGD) [14] forms a dictionary comprising of S FIR graph filters, i.e., $\mathbf{H}_{1:S}(\mathbf{S}) = [\mathbf{H}_1(\mathbf{S}), \ldots, \mathbf{H}_S(\mathbf{S})]$, where to each filter $\mathbf{H}_s(\mathbf{S})$ [cf. (1)] it is referred to as a sub-dictionary. Given the link with the FIR graph filters, a PGD is inherently defined by the coefficients of all S sub-dictionaries $\mathbf{h}_{1:S} = [\mathbf{h}_1^\top, \ldots, \mathbf{h}_S^\top]^\top \in \mathbb{R}^{(K+1)S\times 1}$, where $\mathbf{h}_s = [h_{0s}, \ldots, h_{Ks}]^\top$ are the coefficients of the sth sub-dictionary filter (1). Training the PGD dictionary $\mathbf{H}_{1:S}(\mathbf{S})$ reduces to estimating the filter coefficients $\mathbf{h}_{1:S}$. To identify the latter, we start with a set of T graph signals collected in the matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_T] \in \mathbb{R}^{N\times T}$. Then, we aim to find a collection of T sparse vectors $\mathbf{z}_\tau \in \mathbb{R}^{NS}$ with $\tau = 1, ..., T$, each having at most S_0 non-zero entries, i.e., $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_T] \in \mathbb{R}^{NS\times T}$ and the coefficients $\mathbf{h}_{1:S}$. Formally, this task translates into solving the optimization problem

$$\begin{aligned}
\underset{\mathbf{h}_{1:S},\mathbf{Z}}{\text{argmin}} \quad & \|\mathbf{X} - \mathbf{H}_{1:S}(\mathbf{S})\mathbf{Z}\|_F^2 + \gamma\|\mathbf{h}_{1:S}\|_2^2 \\
\text{subject to} \quad & \|\mathbf{z}_\tau\|_0 \leq S_0,\ \tau = 1, \ldots, T, \\
& \mathbf{H}_s(\mathbf{S}) = \sum_{k=0}^{K} h_{ks}\mathbf{S}^k, s = 1, \ldots, S, \\
& 0\mathbf{I}_N \preceq \mathbf{H}_s(\mathbf{S}) \preceq \delta\mathbf{I}_N, s = 1, \ldots, S, \\
& (\delta - \delta_1)\mathbf{I}_N \preceq \sum_{s=1}^{S} \mathbf{H}_s(\mathbf{S}) \preceq (\delta + \delta_2)\mathbf{I}_N.
\end{aligned} \tag{2}$$

Problem (2) minimizes the Frobenius norm distance between the signals in $\mathbf{X}$ and their sparse dictionary reconstruction $\mathbf{H}_{1:S}(\mathbf{S})\mathbf{Z}$, while regularizing with a ℓ_2-norm of the coefficients $\gamma\|\mathbf{h}_{1:S}\|_2^2$. The ℓ_0-norm constraint $\|\mathbf{z}_\tau\|_0$ forces each vector $\mathbf{z}_\tau$ to be at most S_0-sparse. The other constrains impose the sub-dictionaries forming $\mathbf{H}_{1:S}(\mathbf{S})$ to be FIR filters and control the eigenvalues of each filter to be at most δ (third contraint) and that the total sum of eigenvalues to be bounded between $(\delta - \delta_1)$ and $(\delta + \delta_2)$ for some scalars $\delta, \delta_1, \delta_2$.

Notice that while other non-graph-based dictionaries can be used to sparsely represent graph signals, the approach in (2) is attractive because: (i) it is built by leveraging the coupling between the signal and the underlying graph; (ii) it forces a local representation of radius K from a node therefore captures local details; and (iii) it often allows sparser representations for graph signals [14]. Therefore, in the sequel, we will focus on sampling graph signals with sparse representations as per (2). However, the approach here presented can be readily extended to any general sparse signal representation by considering a different dictionary.

III. Problem formulation

We consider the scenario that a graph signal $\mathbf{x}$ follows a sparse representation w.r.t. the PGD $\mathbf{\Psi} \in \mathbb{R}^{N\times L}$

$$\mathbf{x} = \mathbf{\Psi}\mathbf{s}_0 \tag{3}$$

where $\mathbf{s}_0$ is an $L \times 1$ vector with sparsity $S_0 \ll L$. Suppose the signal $\mathbf{x}$ is corrupted by an additive Gaussian noise, i.e., $\mathbf{y} = \mathbf{x} + \mathbf{n} = \mathbf{\Psi}\mathbf{s}_0 + \mathbf{n}$ where $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_n)$ with the covariance matrix $\mathbf{\Sigma}_n = \text{diag}(\sigma_1^2, ..., \sigma_N^2)$. Our goal is to design a sampling strategy that subsamples $\mathbf{y}$ from a subset of nodes $\mathcal{S} \subseteq \mathcal{V}$ and to recover the original signal $\mathbf{x}$ using (3) with minimum distortion.

To formalize sampling, let us consider a binary selection matrix $\mathbf{C}_\mathcal{S}$ drawn from the combinatorial set

$$\mathcal{C}_{\mathcal{S},N} = \{\mathbf{C}_\mathcal{S} \in \{0,1\}^{|\mathcal{S}|\times N} : \mathbf{C}_\mathcal{S}\mathbf{1}_N = \mathbf{1}_{|\mathcal{S}|}, \mathbf{C}_\mathcal{S}^\top\mathbf{1}_{|\mathcal{S}|} \preceq \mathbf{1}_{|\mathcal{S}|}\}. \tag{4}$$

By construction, matrix $\mathbf{C}_\mathcal{S}$ satisfies $\mathbf{C}_\mathcal{S}\mathbf{C}_\mathcal{S}^\top = \mathbf{I}_{|\mathcal{S}|}$, $\mathbf{C}_\mathcal{S}^\top\mathbf{C}_\mathcal{S} = \text{diag}(\mathbf{c})$ and $\mathbf{c} \in \{0,1\}^N$ such that $c_i = 1$ if $v_i \in \mathcal{S}$.[1] Thus, the sampled noisy signal over the set $\mathcal{S}$ can be expressed as

$$\begin{aligned}
\mathbf{y}_\mathcal{S} &= \mathbf{C}_\mathcal{S}(\mathbf{x} + \mathbf{n}), \\
&= \mathbf{C}_\mathcal{S}\mathbf{\Psi}\mathbf{s}_0 + \mathbf{n}_\mathcal{S},
\end{aligned} \tag{5}$$

where $\mathbf{n}_\mathcal{S}$ represents the noise at the nodes in $\mathcal{S}$. Sampling and reconstruction now depend on designing $\mathbf{C}_\mathcal{S}$ and estimating $\mathbf{s}_0$ from observations $\mathbf{y}_\mathcal{S}$. However, differently from graph signal sampling with bandlimited representation, model (5) is underdetermined because the system matrix $\mathbf{C}_\mathcal{S}\mathbf{\Psi}$ is wide instead of tall. Hence, conventional sampling techniques cannot be applied anymore.

For such an ill-posed problem, a unique solution for $\mathbf{s}_0$ can be obtained through regularization. To exploit the sparsity of $\mathbf{s}_0$, it is a natural choice to adopt the sparse recovery paradigm [17], which regularizes the ill-posed problem by seeking the sparsest estimate of $\mathbf{s}_0$ that fits the observations. Then, our goal for jointly designing $\mathbf{C}_\mathcal{S}$ and estimating $\mathbf{s}_0$ can be framed as solving the optimization problem

$$\begin{aligned}
\underset{\mathbf{C}_\mathcal{S},\mathbf{s}}{\text{argmin}} \quad & \|\mathbf{s}\|_0 \\
\text{subject to} \quad & \|\mathbf{y}_\mathcal{S} - \mathbf{C}_\mathcal{S}\mathbf{\Psi}\mathbf{s}\|_2 \leq \epsilon, \\
& \mathbf{C}_\mathcal{S} \in \mathcal{C}_{\mathcal{S},N}.
\end{aligned} \tag{$\text{P}_{J,0}^\epsilon$}$$

which seeks for a vector $\mathbf{s}$ with the minimum number of non-zero entries, while tolerating the reconstructed signal $\mathbf{C}_\mathcal{S}\mathbf{\Psi}\mathbf{s}$ on the sampled nodes to deviate from $\mathbf{y}_\mathcal{S}$ by at most ϵ, and constraining $\mathbf{C}_\mathcal{S}$ to be a proper sampling matrix drawn from (4). If it were not for the sampling matrix $\mathbf{C}_\mathcal{S}$, problem ($\text{P}_{J,0}^\epsilon$) could be solved with conventional pursuit algorithms for $\mathbf{s}$ [17]. However, matrix $\mathbf{C}_\mathcal{S}$ adds a major difficulty to the problem because of its combinatorial nature, making problem ($\text{P}_{J,0}^\epsilon$) NP-hard. Our goal next is to circumvent this challenge via an alternating minimization to efficiently estimate the sparse vector $\mathbf{s}$ and optimize the sampling matrix $\mathbf{C}_\mathcal{S}$.

[1] Notice that $\mathcal{S}$, $\mathbf{C}_\mathcal{S}$ and $\mathbf{c}$ are equivalent representations of a certain node selection – knowing one of them, the other two are uniquely defined.

Algorithm 1 ADS algorithm for problem $(\mathrm{P}^{\epsilon}_{J,0})$

Objective: Minimize $(\mathrm{P}^{\epsilon}_{J,0})$ via alternate minimization.
Initialize $\mathcal{S}_0$ randomly, set iteration count $i = 0$
while $|\mathcal{S}_i| \leq |\mathcal{S}|$ **do**
  Build $\tilde{\mathbf{\Psi}}_{\mathcal{S}_i}$ by removing the all-zero columns of $\mathbf{C}_{\mathcal{S}_i}\mathbf{\Psi}$ and normalizing the columns.
  Update $\mathbf{s}$ by solving $(\mathrm{P}^{\epsilon}_{\mathcal{S}_i,0})$ through the OMP algorithm.
  Build $\mathbf{\Psi}^{(i)}_{\text{nnz}}$ with columns of $\mathbf{\Psi}$ corresponding to the non-zero elements in $\hat{\mathbf{s}}^{(i)}$.
  Update $\mathbf{C}_{\mathcal{S}_{i+1}}$ via the greedy algorithm.
  $i \leftarrow i + 1$
end while
Use $\mathbf{y}_{\mathcal{S}}$ to estimate the original signal $\mathbf{x}$ through (8).

IV. Alternating Dictionary-based Sampling

We approach problem $(\mathrm{P}^{\epsilon}_{J,0})$ through alternating optimizations between $\mathbf{s}$ and $\mathbf{C}_{\mathcal{S}}$. To be precise, we start with a sampling set $\mathcal{S}_0 \subset \mathcal{S}$ of cardinality $|\mathcal{S}_0| \ll |\mathcal{S}|$. The initial set $\mathcal{S}_0$ is defined either randomly or by a prior. After the noisy data $\mathbf{y}_{\mathcal{S}_0}$ are sampled over the nodes in $\mathcal{S}_0$, we set $\mathbf{C}_{\mathcal{S}} = \mathbf{C}_{\mathcal{S}_0}$ and solve $(\mathrm{P}^{\epsilon}_{J,0})$ only w.r.t. $\mathbf{s}$ via pursuit algorithms [17]. Then, with the obtained solution $\hat{\mathbf{s}}^{(0)}$, we update the sampling set to $\mathcal{S}_1 \supset \mathcal{S}_0$ via sparse sensing techniques [18]. The procedure then repeats with $\mathcal{S}_1$. This alternating update will produce a series of sampling sets $\mathcal{S}_1 \subset \mathcal{S}_2 \subset \ldots \subset \mathcal{S}$. Without loss of generality, we keep the selection stepsize unitary, i.e., $|\mathcal{S}|_i - |\mathcal{S}|_{i-1} = 1$. At iteration i, we therefore first find the sparsest representation $\hat{\mathbf{s}}^{(i)}$ of the sampled noisy data $\mathbf{y}_{\mathcal{S}_i}$, then sample the nodes to build the sampling set $\mathcal{S}_{i+1}$. The algorithm terminates after the selection budget $|\mathcal{S}|$ is reached. We refer to this scheme as alternating dictionary-based sampling (ADS).

A. Support Estimation

To detail the above procedure, at iteration i, we are given the selection set $\mathcal{S}_i$ and the respective samples $\mathbf{y}_{\mathcal{S}_i}$. First, we construct the corresponding normalized subsampled dictionary $\tilde{\mathbf{\Psi}}_{\mathcal{S}_i}$ by first removing the all-zero columns of $\mathbf{C}_{\mathcal{S}_i}\mathbf{\Psi}$, and then normalizing the remaining ones. The support of $\mathbf{s}^{(i)}$ is identified by solving the optimization problem

$$\begin{aligned} \underset{\mathbf{s}^{(i)}}{\text{argmin}} \quad & \|\mathbf{s}^{(i)}\|_0 \\ \text{subject to} \quad & \|\mathbf{y}_{\mathcal{S}_i} - \tilde{\mathbf{\Psi}}_{\mathcal{S}_i}\mathbf{s}^{(i)}\|_2 \leq \epsilon, \end{aligned} \qquad (\mathrm{P}^{\epsilon}_{\mathcal{S}_i,0})$$

which is a typical sparse recovery problem and can be solved via a range of well-developed algorithms [17]. We will use the OMP algorithm to solve problem $(\mathrm{P}^{\epsilon}_{\mathcal{S}_i,0})$. This is because OMP is one of the simplest and fastest sparse recovery algorithms; and our goal is to demonstrate the effectiveness of the proposed ADS paradigm, for which OMP suffices.

The OMP algorithm is a greedy sparse recovery algorithm, which at every iteration identifies the (normalized) dictionary atom that is the most correlated to the input observation. Then, the contribution of the identified atom is excluded from the input data and the residual is regarded as the next input. Such a process is repeated until the stopping criterion is met. There exist a few different OMP stopping criteria that are commonly used [19]:

1) The algorithm terminates when the number of iterations reaches the desired sparsity.
2) The algorithm terminates when the magnitude of the residual is sufficiently small.
3) The algorithm terminates when the correlation between the non-selected atoms and the residual is smaller than a threshold.

While for criterion 1) the OMP algorithm requires the knowledge of the sparsity as prior information, the other two criteria require tuning the respective thresholds to get desirable sparse recovery performances. Since in this work we do not focus on the explicit implementation of the sparse recovery algorithm, we choose the stopping criterion 1) but any of the others can be used.

B. Node Sampling

Once the estimate $\hat{\mathbf{s}}^{(i)}$ is obtained, the goal next is to leverage its sparsity to update the sampling matrix $\mathbf{C}_{\mathcal{S}_{i+1}}$. At first sight, it may seem we still need to face an underdetermined system. But since we know the support of $\hat{\mathbf{s}}^{(i)}$, we can transform the system into an overdetermined one. Considering $\hat{\mathbf{s}}^{(i)}$ is sufficiently sparse, we can remove the redundant atoms in $\mathbf{\Psi}$ and keep only those $\mathbf{\Psi}^{(i)}_{\text{nnz}}$ that are necessary to represent the estimate, i.e.,

$$\hat{\mathbf{x}}^{(i)} = \mathbf{\Psi}^{(i)}_{\text{nnz}}\hat{\mathbf{s}}^{i}_{\text{nnz}} \qquad (6)$$

where $\hat{\mathbf{s}}^{i}_{\text{nnz}}$ is the shorter vector containing the non-zero entries of $\hat{\mathbf{s}}^{(i)}$ and $\mathbf{\Psi}^{(i)}_{\text{nnz}}$ has now dimensions $N \times \text{nnz}(\hat{\mathbf{s}}^{(i)})$, with $\text{nnz}(\hat{\mathbf{s}}^{(i)})$ denoting the number of non-zero elements in $\hat{\mathbf{s}}^{(i)}$. Therefore, we can write the measurement model for the sampling set $\mathcal{S}_{i+1}$ as

$$\mathbf{y}_{\mathcal{S}_{i+1}} = \mathbf{C}_{\mathcal{S}_{i+1}}\mathbf{\Psi}^{(i)}_{\text{nnz}}\mathbf{s}^{(i)}_{\text{nnz}} + \mathbf{n}_{\mathcal{S}_{i+1}}, \qquad (7)$$

where now we work with a tall matrix $\mathbf{\Psi}^{(i)}_{\text{nnz}}$. Notice that in (7) we treat $\mathbf{s}^{(i)}_{\text{nnz}}$ as an unknown variable for which we want to design the sampling matrix $\mathbf{C}_{\mathcal{S}_{i+1}}$ that leads to the best estimate for it. In other words, the sparse solution obtained by solving $(\mathrm{P}^{\epsilon}_{\mathcal{S}_i,0})$ with OMP is only used now to obtain the support of the sparse representation and build the tall matrix $\mathbf{\Psi}^{(i)}_{\text{nnz}}$.

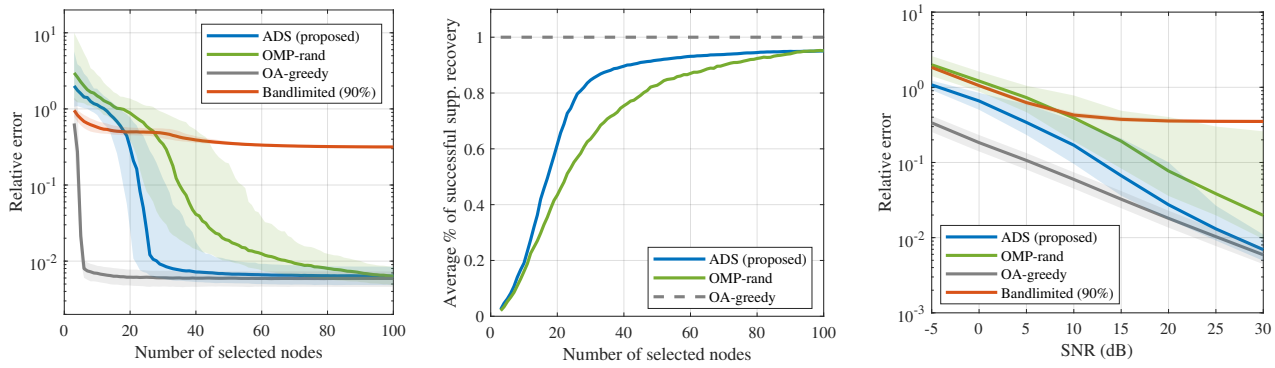
The second step (i.e. node sampling) of the ADS algorithm consists of updating the sampling set by adding one of the residual nodes that yields the best estimate of $\mathbf{s}^{(i)}_{\text{nnz}}$ through model (7). To describe how good the estimation can be achieved for $\mathbf{s}^{(i)}_{\text{nnz}}$ (and hence for the entire graph signal $\mathbf{x}$), we use the best linear unbiased estimator [20],

$$\hat{\mathbf{x}}^{(i+1)}_{\text{B}} = \mathbf{\Psi}^{(i)}_{\text{nnz}}\mathbf{\Theta}^{\dagger}_{i}\mathbf{\Psi}^{(i)\,\text{H}}_{\text{nnz}}\mathbf{C}^{\top}_{\mathcal{S}_{i+1}}\left(\mathbf{C}_{\mathcal{S}_{i+1}}\mathbf{\Sigma}_n\mathbf{C}^{\top}_{\mathcal{S}_{i+1}}\right)^{-1}\mathbf{y}_{\mathcal{S}_{i+1}} \qquad (8)$$

with $\mathbf{\Theta}_i = \mathbf{\Psi}^{(i),\text{H}}_{\text{nnz}}\mathbf{C}^{\top}_{\mathcal{S}_{i+1}}\left(\mathbf{C}_{\mathcal{S}_{i+1}}\mathbf{\Sigma}_n\mathbf{C}^{\top}_{\mathcal{S}_{i+1}}\right)^{-1}\mathbf{C}_{\mathcal{S}_{i+1}}\mathbf{\Psi}^{(i)}_{\text{nnz}}$ and $(\cdot)^{\dagger}$ denoting the Moore–Penrose pesudoinverse. We then quantify the estimation performance through its mean square deviation (MSD)

$$\begin{aligned} \text{MSD} &= \mathbb{E}\left[\|\hat{\mathbf{x}}^{(i+1)}_{\text{B}} - \mathbf{x}\|^2_2\right] = \text{tr}\left(\mathbf{\Theta}^{-1}_i\right) \\ &= \text{tr}\left[\left(\mathbf{\Psi}^{(i)\text{H}}_{\text{nnz}}\text{diag}(\mathbf{c}_{i+1})\mathbf{\Sigma}^{-1}_n\mathbf{\Psi}^{(i)}_{\text{nnz}}\right)^{-1}\right], \end{aligned} \qquad (9)$$

where the last equality holds since $\mathbf{\Sigma}_n$ is diagonal.



(a) Comparison of selection methods (b) Average support recovery rate (c) Impact of SNR

Figure 1. Performance comparison of different sampling methods: (a) relative error of different selection methods vs. the number of observations; (b) average percentage of the correctly recovered support by OMP for the results in (a); (c) relative error for different SNRs. For (a) and (c), the solid lines indicate the medians, whereas the shaded areas mark the inter-quanrtile ranges of 25%-75% percentiles.

Thus, we can now update the sampling set $\mathcal{S}_{i+1}$ by minimizing the MSD w.r.t. $\mathbf{c}_{i+1}$, i.e., solving

$$\begin{aligned} \underset{\mathbf{c}_{i+1}}{\text{argmin}} \quad & \text{tr}\left[\left(\boldsymbol{\Psi}_{\text{nnz}}^{(i)\text{H}}\text{diag}(\mathbf{c}_{i+1})\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\Psi}_{\text{nnz}}^{(i)}\right)^{-1}\right] \\ \text{subject to} \quad & \mathbf{c}_{i+1} \in \{0,1\}^{N\times 1},\ \|\mathbf{c}_{i+1}\|_0 = |\mathcal{S}_{i+1}| \\ & \|\mathbf{c}_{i+1} - \mathbf{c}_i\|_0 = 1,\ \mathcal{S}_i \subset \mathcal{S}_{i+1} \end{aligned} \tag{10}$$

where the last constraint indicates that we increase the sampling set by one sample. Therefore, $\mathcal{S}_{i+1}$ should contain $\mathcal{S}_i$. In the above steps, the noise coveriance $\boldsymbol{\Sigma}_n$ is a known, which can be easily estimated from pilot samples.

Problems of the form in (10) are standard within the sparse sensing framework. Since we have the set inclusion constraint $\mathcal{S}_i \subset \mathcal{S}_{i+1}$ and need to increase the size of the sampling set, approaching such via greedy methods is a natural choice [1]. As a result, we can directly apply the greedy heuristic to problem (10) by adding to the sampling set the residual node $n \in \bar{\mathcal{S}}_i = \mathcal{V} \setminus \mathcal{S}_i$ that minimizes the MSD. Alternatively, other criteria from experimental design that exhibit amenable properties for greedy selection (e.g., submodularity [21]), such as the (pseudo) log-determinant criterion

$$f_i(n) = -\log\det\left[\boldsymbol{\Theta}_i(\mathcal{S}_i \cup \{n\}) + \xi\mathbf{I}\right] \tag{11}$$

can be used, where the term $\xi\mathbf{I}$ with $\xi \ll 1$ avoids the rank deficiency of $\boldsymbol{\Theta}_i$. We considered the log-determinant critetion in the numeral experiments.

The node sampling step concludes iteration i of the ADS algorithm. All the steps are summarized in Algorithm 1. We would like to remark that the interplay between the support estimation and node selection does not necessarily have any submodularity guarantee even if a submodular function is used in (10). This is because the support estimated from $(\text{P}^{\epsilon}_{\mathcal{S}_i,0})$ may change between iterations; especially in the earlier ones. However, when the number of samples becomes large enough and the support does not change, using submodular functions in (10) may come with near-optimal guarantees [21]. A deeper analysis of the latter will be done in future work.

V. Numerical Experiments

This section presents numerical experiments to corroborate the proposed approach and compare it with the baseline graph-bandlimited sampling. We considered the scenario in [14] comprising a graph of $N = 100$ randomly placed nodes and a PGD with $S = 4$ subdictionaries of FIR filters with orders $K = 5$. We generate the signal $\mathbf{x}$ through the linear combination of four random atoms of the oracle dictionary with uniformly distributed coefficients in $[0, 1]$; hence $S_0 \leq 4$. We corrupt the data with different SNRs in $[-5\text{dB}, 30\text{dB}]$. The parameter ξ in (11) is set to 10^{-3}. The initial selection set is built at random and contains two nodes out of the 100 available. We run $R = 10^3$ Monte-Carlo simulations, and we measure the reconstruction performance through the relative error, $\text{RE} = \|\mathbf{x}_r - \hat{\mathbf{x}}_r\|_2 / \|\mathbf{x}_r\|_2$, where $\mathbf{x}_r$ and $\hat{\mathbf{x}}_r$ denotes the r-th true signal and reconstructed signal, respectively. We compare the proposed ADS approach with:

1) OA-greedy: Oracle algorithm that knows the $S_0 = 4$ true atoms (instead of estimating through OMP) and adopts a greedy solution to sample the nodes. It allows comparing with the best performance we can achieve.
2) OMP-rand: Estimates the support with OMP but samples nodes uniformly at random. Shows the effectiveness of the greedy sampling de-factorizing the impact of the support.
3) Standard bandlimited-based greedy selection with a bandwidth containing 90% of the energy. The bandlimited frequency supports are calculated on a per signal basis. This case contrasts the ADS with a baseline solution.

Cardinality sampling set. We first analyze the impact of the number of sampled nodes; thus, we consider the true dictionary known and fix the SNR to 30dB. Fig. 1(a) depicts the reconstruction performance of the compared methods for different $|\mathcal{S}|$. In contrast to the bandlimited sampling, the ADS and OMP-rand converge to the OA-greedy lower bound, which confirms that the proposed method can indeed address the concerned sampling that bandlimited approach struggles to handle. The result of ADS has a typical trend - the relative error initially reaches a slow-decaying *plateau*, followed by a rapid decrease after around 30 observations. This plateau can be a result of initial lack of observations, which makes it considerably hard for the OMP algorithm to estimate the sparse support. Once sufficient observations are supplied, more correct supports can be identified. The evidence can also be found in Fig. 1(b), which illustrates the average rate of successful support recovery for the different methods. For the ADS, the recovery rate is initially low and reaches its turning points

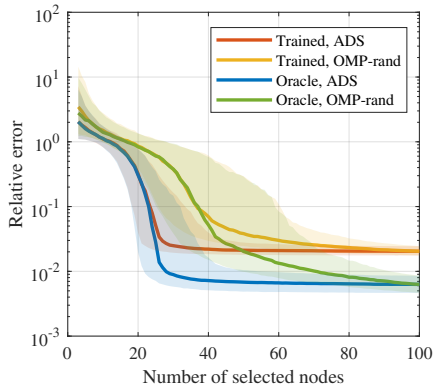


Figure 2. Performance comparison for using a trained and the exact dictionary. Solid lines and shaded areas indicate the median and interquartile ranges.

roughly at around $N = 30$ nodes. The random selection follows a similar pattern, but reaches the turning point slower (at around $N = 45$), suggesting that greedily sampled data not only gives better reconstruction but also benefits the sparse recovery.

SNR. Fig 1(c) compares the reconstruction error for the concerned methods for $|\mathcal{S}| = 50$ observations under different SNRs. We can observe that the SNR has a significant impact on the proposed method. Under low SNRs, the proposed scheme results in considerably high reconstruction errors even with half of the nodes being sampled. This is possibly due to the fact the OMP's performance degrades significantly in highly noisy conditions. For higher SNRs, the proposed method improves rapidly and converges to the oracle result at around 20dB. Furthermore, the proposed method performs uniformly better than random sampling in terms of the median and the spread of the errors, proving the greedy method is an effective selection scheme for the ADS framework.

Trained dictionary. Lastly, we evaluate the proposed method when the true dictionary is unknown. We generate 2600 data samples corrupted by noise with SNR $= 30$ dB, from which 600 are used to train the dictionary while the rest for testing. The training procedure follows [14] but now with noisy data, and results in a relative dictionary representation error of 3.62%. Fig. 2 depicts the reconstruction errors as a function of $|\mathcal{S}|$ for both the true (oracle) and trained dictionaries. When the number of selected nodes is low, the OMP-induced noise dominates. After around 30 observations, both medians drop rapidly and converge to their steady-state errors. The difference between the trained and oracle dictionary in steady-state is comparable to the dictionary representation error, which indicates the main source of error to be dictionary training. These results shows also the reliability of the proposed method with trained dictionary, which does not contribute to significant excessive error beyond those of the trained dictionary. Such error can be further reduced by increasing the training samples.

VI. Conclusions

This work proposed a sampling strategy for graph signals that enjoy sparse dictionary representations. This sparse representation is of interest when the graph signal does not satisfy the smoothness or the bandlimitedness prior, thus cannot facilitate effective sampling. We instead resort to the parametric graph dictionaries to represent the signal and proposed a sampling scheme for this underdetermined system that relies on the interplay between support estimation and subset selection. Starting with a small given sampling set (e.g., random), we solve a trimmed-dictionary representation problem via orthogonal matching pursuit to estimate the support. Then, we use this support to sample extra nodes, which in turn are used to refine the support estimate. This alternating procedure is repeated until the desired number of samples is reached. One of the main limitations we have observed is the impact of the initial sampling set, which can be addressed by not starting at random. This aspect is left for future work due to space limitations.

References

[1] P. Di Lorenzo, S. Barbarossa, and P. Banelli, "Sampling and recovery of graph signals," *arXiv preprint arXiv:1712.09310*, 2017.

[2] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Transactions on Signal Processing*, vol. 63, pp. 6510–6523, Dec. 2015.

[3] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, "Sampling signals on graphs: From theory to applications," 2020.

[4] I. Pesenson, "Sampling in paley-wiener spaces on combinatorial graphs," *Transactions of the American Mathematical Society*, vol. 360, no. 10, pp. 5603–5627, 2008.

[5] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 5445–5449.

[6] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, 2016.

[7] F. Gama, A. G. Marques, G. Mateos, and A. Ribeiro, "Rethinking sketching as sampling: Linear transforms of graph signals," in *2016 50th Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 522–526.

[8] L. F. O. Chamon and A. Ribeiro, "Greedy sampling of graph signals," *IEEE Transactions on Signal Processing*, vol. 66, no. 1, pp. 34–47, 2018.

[9] Y. Bai, F. Wang, G. Cheung, Y. Nakatsukasa, and W. Gao, "Fast graph sampling set selection using gershgorin disc alignment," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2419–2434, 2020.

[10] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3775–3789, 2016.

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

[12] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Applied and Computational Harmonic Analysis*, vol. 44, no. 2, pp. 446 – 475, 2018.

[13] N. Perraudin, B. Ricaud, D. I. Shuman, and P. Vandergheynst, "Global and local uncertainty principles for signals on graphs," *APSIPA Transactions on Signal and Information Processing*, vol. 7, pp. e3, 2018.

[14] D. Thanou, D. I. Shuman, and P. Frossard, "Learning parametric dictionaries for signals on graphs," *IEEE Transactions on Signal Processing*, vol. 62, no. 15, pp. 3849–3862, 2014.

[15] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[16] 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.

[17] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer Publishing Company, Incorporated, 1st edition, 2010.

[18] S. P. Chepuri and G. Leus, "Sparse sensing for statistical inference," *Foundations and Trends® in Signal Processing*, vol. 9, no. 3–4, pp. 233–368, 2016.

[19] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[20] S. M. Kay, *Fundamentals of statistical signal processing*, Prentice Hall PTR, 1993.

[21] E. Tohidi, R. Amiri, M. Coutino, D. Gesbert, G. Leus, and A. Karbasi, "Submodularity in action: From machine learning to signal processing applications," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 120–133, 2020.