Random Sampling for Bandlimited Signals on Product Graphs

Rohan Varma
Dept. of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, U.S.A
rohanv@andrew.cmu.edu

Jelena Kovačević
Tandon School of Engineering
New York University
New York City, U.S.A
jelenak@nyu.edu

***Abstract*—In this work, we construct a structured framework for the efficient random sampling and recovery of bandlimited graph signals that lie on product graphs. Product graphs are a model to construct large complex graphs from smaller simpler building blocks we call *graph atoms*, and are a convenient tool to model rich classes of multi-modal graph-structured data. Our randomized sampling framework prescribes an optimal sampling distribution over the nodes of the product graph constructed by only processing these smaller graph atoms. As a result, the framework achieves significant savings in computational complexity with respect to previous works that do not exploit the inherent structure of product graphs.**

***Index Terms*—sampling, bandlimited, graph signal, product graph, random**

I. Introduction

The task of sampling and recovery is one of the most critical topics in the signal processing community. With the explosive growth of information and communication, signals are being generated at an unprecedented rate from various sources, including social networks, citation networks, biological networks, and physical infrastructure [1]. Unlike time-series signals or images, these signals possess complex, irregular structure, which requires novel processing techniques leading to the emerging field of signal processing on graphs [2]–[4]. Since the structure can be represented by a graph, we call these signals as *graph signals*.

Many examples of real-world graph-structured data are multi-modal in nature and importantly have an inherent structure. Product graphs are a graph model that composes graphs from smaller building blocks we call *graph atoms* and represent a concise way to model such data [4], [5]. For example, product graph *composition* using a product operator is a natural way to model time-varying signals on a sensor network as shown in Figure 1(b). The graph signal formed by the measurements of all the sensors at all the time steps is supported by the graph that is the product of the sensor network graph and the time series graph. The k^{th} measurement of the n^{th} sensor is indexed by the n^{th} node of the k^{th} copy of the sensor network graph. In [5], a generative model that can effectively model the structure of many large real-world networks was presented by recursively applying the Kronecker product on a base graph that can be estimated efficiently. Consequently, constructing a framework for the efficient sampling and recovery on such product graphs is an important step for tasks such as graph signal recovery, compression, and semi-supervised learning on large-scale and multi-modal graphs.

Multiple types of graph products exist, that is, we can enforce connections across modes in different ways [6]. In the case of the Cartesian product as in Figure 1(b), the measurement of the n^{th} sensor at the k^{th} time step is related to not only to its neighboring sensors at the k^{th} time step but also to its measurements at the $(k-1)^{th}$ and $(k+1)^{th}$ time steps respectively. Previous works have however studied sampling strategies on the entire graph in question which can often be computationally expensive for large graphs.

This work is supported in part by NSF under grants CCF-1563918

(a) (b)

Fig. 1: (a) Under the Kronecker product, $(u_1, u_2) \sim (v_1, v_2)$ in the product graph if $u_1 \sim v_1$ and $u_2 \sim v_2$. (b) Under the Cartesian product, $(u_1, u_2) \sim (v_1, v_2)$ in the product graph if $u_1 = v_1$ and $u_2 \sim v_2$ or $u_1 \sim v_1$ and $u_2 = v_2$

The assumption that graph signals vary slowly or are smooth over the graph is a natural one to make. Many real world graph signals like sensor network data and biological network data are smooth, or exhibit bandlimited behavior, or have known limited support with respect to the graph Fourier transform. For example, in the context of semi-supervised classification on graphs, each vertex represents one data point to which a label is associated and a graph can be formed by connecting vertices with weights corresponding to the affinity or distance between the data points in some feature space. It is then natural to assume that the *label signal* has slow variation or is *smooth* on the graph and consequently approximately bandlimited. Since labeled instances are rare or expensive to collect, devising efficient yet frugal sampling algorithms on large complex graphs is of significant interest.

In [7], [8], a sampling theory for bandlimited signals was presented that can be considered as an extension of Nyquist sampling for regular domains to irregular domains. Further, we have extended this sampling theory to product graphs in [9] by showing how to efficiently sample and perfectly recover bandlimited signals on product graphs. Particularly, it was shown that we do not need to process the whole product graph $\mathbf{A}$ or compute its spectral decomposition which is of complexity $O(N^3)$ and is often computationally prohibitive for large graphs. While the sampling theory characterizes sampling sets that enable perfect recovery for bandlimited signals, it does not prescribe easily implementable, robust sampling strategies. Randomized sampling strategies [10], [11], characterized by a probability distribution over the nodes, present a more flexible framework to sample nodes on a graph in the presence of noise that is also easily implementable. In this work, we extend these randomized sampling strategies to product graphs by exploiting the structure of product graph. Particularly, as in the case of the sampling theory for product graphs [9], we only need to process the graph atoms the product graph is composed of.

II. Product Graphs and Graph Signal Processing

A. Graphs

We consider a graph $G = (\mathcal{V}, \mathbf{A})$, where $\mathcal{V} = \{v_0, \dots, v_{N-1}\}$ is the set of nodes and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the graph shift, or a weighted

adjacency matrix. $\mathbf{A}$ Represents the connections of the graph G, which can be either directed or undirected. The edge weight $w(n \to m) = \mathbf{A}_{n,m}$ between nodes v_n and v_m is a quantitative expression of the underlying relation between the n^{th} and the m^{th} node, such as a similarity, a dependency, or a communication pattern. If there exists a non-zero edge weight between v_n and v_m, we write $v_n \sim v_m$. Once the node order is fixed, the graph signal is written as a vector

$$\mathbf{x} \;=\; \left[x_0, x_1, \ldots, x_{N-1}\right]^T \in \mathbb{R}^N.$$

Product graphs are graphs whose adjacency matrices are composed using the *product* (represented by the square symbol $\Box$) of the adjacency matrices of smaller *graph atoms*. Consider two graphs $G_1 = (\mathcal{V}_1, \mathbf{A}_1)$ and $G_2 = (\mathcal{V}_2, \mathbf{A}_2)$. The graph product of G_1 and G_2 is the graph $G = G_1 \Box G_2 = (\mathcal{V}, \mathbf{A}_1 \Box \mathbf{A}_2)$ where $|\mathcal{V}| = |\mathcal{V}_1| \cdot |\mathcal{V}_2|$. The set of nodes $\mathcal{V}$ is the Cartesian product of the sets $\mathcal{V}_1$ and $\mathcal{V}_2$. That is, a node (u_1, u_2) is created for every $u_1 \in \mathcal{V}_1$ and $u_2 \in \mathcal{V}_2$.

Typically, we use one of the Kronecker graph product ($\otimes$, Figure 1(a)), the Cartesian graph product ($\oplus$, Figure 1(b)) or the strong graph product ($\boxtimes$) which is a superimposition of both the Kronecker and Cartesian product to compose product graphs. Since the product is associative, one can extend the above formulation to define product graphs constructed from multiple graph atoms.

Digital images reside on rectangular lattices that are Cartesian products of line graphs for rows and columns. A social network with multiple communities can also be represented by the Kronecker graph product of the graph that represents a community structure and the graph that captures the interaction between neighbors. In the context of recommender engines where we have user ratings for different entities at different times, we can view this as a signal lying on the Kronecker product of three graphs, the graph relating the different users, the graph relating the different entities, and the time graph. In the following exposition, for clarity and brevity, we only consider the Kronecker product. However, the results and theorems either hold or can easily be extended to both Cartesian and strong products. We also only consider undirected graphs where the graph Fourier transform is defined with respect to the symmetric graph shift matrix $\mathbf{A}$, but these results can also be extended for when the graph Fourier transform is defined with respect to the graph Laplacian and when the graph is directed.

Single Graph: The spectral decomposition of $\mathbf{A}$ is $\mathbf{A} = \mathbf{V}\,\Lambda\,\mathbf{U}$ where the eigenvectors of $\mathbf{A}$ form the columns of matrix $\mathbf{V}$. We note that $\mathbf{U} = \mathbf{V}^T$ since the graph G is undirected and $\mathbf{A}$ is symmetric. $\Lambda \in \mathbb{R}^{N \times N}$ is the diagonal matrix of corresponding eigenvalues $\lambda_0 \geq \ldots, \geq \lambda_{N-1}$ of $\mathbf{A}$ ordered in descending order unless specified otherwise. These eigenvalues represent frequencies on the graph [12]. The *graph Fourier transform* is $\widehat{\mathbf{x}} = \mathbf{U}\,\mathbf{x}$ where the vector $\widehat{\mathbf{x}}$ represents the signal's expansion in the eigenvector basis of the graph shift and describes the frequency content of the graph signal $\mathbf{x}$. The *inverse graph Fourier transform* is $\mathbf{x} = \mathbf{V}\,\widehat{\mathbf{x}}$ and reconstructs the graph signal from its frequency content by aggregating graph frequency components weighted by the coefficients of the signal's graph Fourier transform.

Product Graphs: We consider a product graph $G = (\mathcal{V}, \mathbf{A}), |\mathcal{V}| = N$, that is constructed from J graph atoms $G_1, \cdots G_j, \cdots G_J$, where $G_j = (\mathcal{V}_j, \mathbf{A}^j), |\mathcal{V}_j| = N_j$, using the Kronecker product where $\prod_{j=1}^{J} N_j = N$. We note that each node i on the product graph corresponds to a tuple of J nodes $(i_{(1)}, \cdots, i_{(j)}, \cdots i_{(J)})$ over the graph atoms where $i_{(j)}$ is a node on G_j. We can write the resulting graph shift matrix of the product graph as

$$\mathbf{A} = \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)} \otimes \cdots \otimes \mathbf{A}^{(J)} = \otimes_{j=1}^{J} \mathbf{A}^{(j)} \tag{1}$$

We can then write the spectral decomposition of the product graph shift $\mathbf{A}$ as $\mathbf{A} = \mathbf{V}\,\Lambda\,\mathbf{U}$ where

$$\mathbf{V} = \mathbf{V}^{(1)} \otimes \mathbf{V}^{(2)} \otimes \cdots \otimes \mathbf{V}^{(J)} = \otimes_{j=1}^{J} \mathbf{V}^{(j)} \tag{2}$$

$$\Lambda = \Lambda^{(1)} \otimes \Lambda^{(2)} \otimes \cdots \otimes \Lambda^{(J)} = \otimes_{j=1}^{J} \Lambda^{(j)} \tag{3}$$

$$\mathbf{U} = \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \otimes \cdots \otimes \mathbf{U}^{(J)} = \otimes_{j=1}^{J} \mathbf{U}^{(j)} = \mathbf{V}^{-1} \tag{4}$$

For a given graph atom, G_j, the columns of $\mathbf{V}^{(j)}$ and their corresponding frequencies are pairs of the form $(\mathbf{v}^{(j)}_{i_{(j)}}, \lambda^{(j)}_{i_{(j)}})$. Here, $i_{(j)}$ is an index for the nodes in G_j that varies from $(1, 2, \cdots N_j)$ where $N_j = |\mathcal{V}_j|$, the number of nodes in G_j.

As a result, under the Kronecker Product, each of the N basis vectors in $\mathbf{V}$ have the form

$$(\mathbf{v}^{(1)}_{i_{(1)}} \otimes \cdots \otimes \mathbf{v}^{(j)}_{i_{(j)}} \otimes \cdots \otimes \mathbf{v}^{(J)}_{i_{(J)}}, \lambda^{(1)}_{i_{(1)}} \times \cdots \times \lambda^{(j)}_{i_{(j)}} \times \cdots \times \lambda^{(J)}_{i_{(J)}}) \tag{5}$$

across all combinations of the indices $(i_{(1)}, \cdots, i_{(j)}, \cdots, i_{(J)})$. For example, if $\mathbf{V}^{(1)} = [\mathbf{v}^{(1)}_1 \,|\, \mathbf{v}^{(1)}_2]$ and $\mathbf{V}^{(2)} = [\mathbf{v}^{(2)}_1 \,|\, \mathbf{v}^{(2)}_2 \,|\, \mathbf{v}^{(2)}_3]$,

$$\begin{aligned}\mathbf{V}^{(1)} \otimes \mathbf{V}^{(2)} = [&\mathbf{v}^{(1)}_1 \otimes \mathbf{v}^{(2)}_1 \,|\, \mathbf{v}^{(1)}_1 \otimes \mathbf{v}^{(2)}_2 \,|\, \mathbf{v}^{(1)}_1 \otimes \mathbf{v}^{(2)}_3 \,|\, \cdots \\ &\mathbf{v}^{(1)}_2 \otimes \mathbf{v}^{(2)}_1 \,|\, \mathbf{v}^{(1)}_2 \otimes \mathbf{v}^{(2)}_2 \,|\, \mathbf{v}^{(1)}_2 \otimes \mathbf{v}^{(2)}_3]\end{aligned}$$

While in this work we only focus on the graph Fourier transform $\mathbf{V}$, this construction can easily be extended to arbitrary bases and frames. When the basis in question corresponds to the graph wavelet transform, we note that this is analogous to separable wavelet construction by tensorization on images and d-dimensional grids. [13]

B. Bandlimited Signals

We can define a class of bandlimited signals on a graph for any arbitrary ordering of the eigenvectors $\mathbf{V}$ denoted by $\mathrm{BL}_K(\mathbf{V})$:

Definition 1. [11] A graph signal $\mathbf{x} \in \mathbb{R}^N$ is *bandlimited* on a graph $\mathbf{A}$ when there exists a $K \in \{0, 1, \cdots, N-1\}$ such that its graph Fourier transform $\widehat{\mathbf{x}}$ satisfies $\widehat{x}_k = 0 \quad \text{for all} \quad k \geq K$.

For a product graph, given some subset of K columns of $\mathbf{V}$ over which the signal is bandlimited, we can accordingly re-order the columns in each of $\mathbf{V}^{(j)}$ such that $\mathbf{V}_{(K)} \subset \mathbf{V}^{(1)}_{(R_1)} \otimes \cdots \mathbf{V}^{(j)}_{(R_j)} \otimes \cdots \mathbf{V}^{(J)}_{(R_J)} = \bigotimes_{j=1}^{J} \mathbf{V}^{(j)}_{(R_j)} = \mathbf{V}_{(S)}$ where $\mathbf{V}^{(j)}_{(R_j)}$ corresponds to the top R_j columns of $\mathbf{V}^{(j)}$ and $S = \prod_{j=1}^{J} R_j$. We note that $K \leq S \leq K^J$. In addition, any signal that is in $\mathrm{BL}_K(\mathbf{V})$ is also in $\mathrm{BL}_S(\mathbf{V})$.

III. Sampling Theorem

Suppose that we want to sample exactly M coefficients in a graph signal $\mathbf{x} \in \mathbb{R}^N$ to produce a sampled signal $\mathbf{x}_{\mathcal{M}} \in \mathbb{R}^M$ ($M < N$). The sampling operator Ψ corresponding to sampling set $\mathcal{M} \subset [n]$ is a linear mapping from $\mathbb{R}^N$ to $\mathbb{R}^M$, defined as

$$\Psi_{i,j} = \begin{cases} 1, & j = \mathcal{M}_i; \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

We then interpolate $\mathbf{x}_{\mathcal{M}}$ with the interpolation operator Φ which is a linear mapping from $\mathbb{R}^M$ to $\mathbb{R}^N$ to get $\mathbf{x}' \in \mathbb{R}^N$.

A. Single Graph

We now show how to sample and perfectly recover bandlimited graph signals on a single graph:

Theorem 1. [7]. Let Ψ be the sampling operator to sample K coefficients in $\mathbf{x} \in \mathrm{BL}_K(\mathbf{U})$ to produce $\mathbf{x}_\mathcal{M} \in \mathbb{R}^K$ and satisfy $\mathrm{rank}(\Psi\,\mathbf{V}_{(K)}) = K$. Let $\mathbf{W}$ be $(\Psi\,\mathbf{V}_{(K)})^\dagger$. Perfect recovery is then achieved by setting $\Phi = \mathbf{V}_{(K)}\,\mathbf{W}$ such that $x = \Phi\mathbf{x}_\mathcal{M}$.

We note that the sample size M should be no smaller than the bandwidth K and at least one set of K linearly-independent rows in $\mathbf{V}_{(K)}$ always exists.

B. Product Graph

As before, we have a bandlimited graph signal $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V})$ that is associated with the product graph $\mathbf{A}$. It is straightforward to sample the product graph using the framework constructed in the previous section and Theorem 1 for a single graph by using the composed graph-shift $\mathbf{A}$ as a whole. Instead, in this section, we look to exploit the structure of the product graph under the Kronecker product composition when we sample the graph. We note here that we are free to order the eigenvectors of $\mathbf{V}$ arbitrarily.

Theorem 2. [9] Using Theorem 1, for each of the J graph atoms, we can construct appropriate sampling ($\Psi^{(j)}$) and interpolation ($\Phi^{(j)}$) operators corresponding to the subset of columns R_j in $\mathbf{V}^{(j)}$ such that for any $\mathbf{x}^{(j)} \in \mathrm{BL}_{R_j}(\mathbf{V}^{(j)})$, we can sample and perfectly recover such that $\mathbf{x}^{(j)} = \Phi^{(j)}(\Psi^{(j)}\mathbf{x}^{(j)}) = \Phi^{(j)}\mathbf{x}^{(j)}_\mathcal{M}$. In addition, $\mathbf{x}^{(j)}_\mathcal{M}$ is associated with a sampled graph whose graph shift is $\mathbf{A}^{(j)}_\mathcal{M}$.
We then construct the sampling operator Ψ to sample $S = \prod_{j=1}^J R_j$ nodes in the product graph and corresponding interpolation operator Φ such that $\mathbf{x} = \Phi\mathbf{x}_\mathcal{M} = \Phi\Psi\mathbf{x}$ for any bandlimited graph signal $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V})$ as:

$$\Psi = \bigotimes_{j=1}^{J} \Psi^{(j)} \quad \text{and} \quad \Phi = \bigotimes_{j=1}^{J} \Phi^{(j)} \tag{7}$$

We see that we can sample and perfectly recover on the product graph by composing sampling and interpolation operators constructed on the graph atoms. We choose R_j nodes from each of the graphs $G^{(j)}$ and sample $S = \prod_{j=1}^J R_j$ nodes in the product graph such that each sampled node in the product graph correspond to some combination of the sampled nodes on the graph atoms. Hence, we effectively only need to do *choose* $\sum_{j=1}^J R_j$ nodes over the graph atoms. In contrast, in the single graph setting, we need to choose atleast K nodes, where in general $K = O(S)$.

IV. Random Sampling

While the sampling theory discussed above gives conditions on sampling sets that enable perfect recovery for bandlimited signals, it does not prescribe easily implementable robust algorithms to choose these sampling sets. Randomized sampling in this case is particularly favorable especially for large graphs where standard column subset selection or search algorithms may be prohibitive. In this section, we study randomized sampling procedures whereby we sample M nodes proportional to a sampling distribution $\{\pi_i\}$ over the nodes. That is, we sample M nodes without replacement such that in each of the M rounds, the probability of the i-th node being selected is proportional to π_i.

Inspired by the sampling framework discussed in the last section, we want to compose a sampling operator on the product graph from sampling operators we construct on the graph atoms. Consider the following sampling framework: For each of the J graphs, G_j, where $j = \{1, \cdots, J\}$, we define a probability distribution $\{\pi^{(j)}\}$ over its nodes and a corresponding sampling operator $\Psi^{(j)}$ that samples the i-th node in G_j with probability $\pi^{(j)}_i$. As before, we then compose the sampling operators over the graph atoms using the Kronecker product as $\Psi = \bigotimes_{j=1}^J \Psi^{(j)}$ such that the probability of the the i-th node in the product graph G is the product of the probabilities of choosing the corresponding nodes on the graph atoms and $\pi_i = \Pi_{j=1}^J \pi^{(j)}_{i_{(j)}}$. Similarly to the previous section, this allows us to compose a sampling operator Ψ and probability distribution $\{\pi_i\}$ by only processing the graph atoms and constructing sampling operators $\Psi^{(j)}$ and probability distributions $\{\pi^{(j)}\}$ over each graph atom G_j, which is substantially more computationally efficient.

A. Uniform Random Sampling

We first consider uniform random sampling, which is non-adaptive to the graph structure, such that $\pi^{(j)}_i = \frac{1}{N_j}$. Due to lack of space, we only briefly discuss this and provide limited details. Circulant graphs are a popular and well studied graph model because of their regularity and linear shift invariance property. We can also approximate any arbitrary graph by decomposing it into a linear combination of circulant graphs [14]. A circulant graph is a graph whose adjacency matrix $\mathbf{C}$ is circulant such that it can be represented a polynomial of the cyclic permutation matrix, $\mathbf{A}$, whose corresponding graph Fourier transform is the discrete Fourier transform matrix $\mathbf{F}$ such that $\mathbf{C} = \sum_{i=0}^{L-1} h_i\,\mathbf{A}^i = \mathbf{F}^* \left(\sum_{i=0}^{L-1} h_i \Lambda^i\right) \mathbf{F}$. Since the graph Fourier transform matrix of circulant graphs is $\mathbf{F}$, we can perfectly recover a circulant-graph signals with bandwidth K by randomly sampling any $M \geq K$ signal coefficients. Consider a product graph $\mathbf{C} = \mathbf{C}_1 \otimes \mathbf{C}_2$ composed of circulant graphs $\mathbf{C}_1$ and $\mathbf{C}_2$ diagonalized by $\mathbf{F}_1$ and $\mathbf{F}_2$ respectively. We extend this idea to product graphs by showing that the corresponding graph Fourier transform of $\mathbf{C}$ is $\mathbf{F}_1 \otimes \mathbf{F}_2$, and consequently it is sufficient to uniformly randomly sample nodes on each of the graph atoms and compose the sampling operator over the product graph using the product. A frame $\mathcal{F}$ is a generating system $\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N\}$ of $\mathbb{R}^K$, where $N \geq K$. We represent the frame as an $N \times K$ matrix with rows $\mathbf{f}_k^T$. The frame $\mathcal{F}$ is *maximally robust to erasures* when every $K \times K$ submatrix obtained by deleting $N - K$ rows of $\mathcal{F}$ is invertible [15]. More generally, it then follows that if $\mathbf{V}_{(K)}$ is (approximately) maximally robust to erasures, any uniform random sampling operator that samples at least K signal coefficients guarantees recovery with high probability. We can show that this holds true for a family of graphs that include circulant graphs, graphs with a *regular* structure like d-dimensional grids or ring graphs, and Erdős-Rényi random graphs. Further, for product graphs composed of graph atoms that belong to this family of graphs, it is sufficient to uniformly randomly sample on each of the graph atoms and compose the sampling operator using the product operator.

B. Experimentally-Designed Sampling

Uniform random sampling performs sub-optimally for many real-world irregular graphs and more complex graph models. We now consider experimentally designed sampling, where the sampling distribution is non-uniform and adapted to the graph structure. Particularly, we aim to sample the most informative nodes with respect to the bandlimited class of signals. In [10], a random sampling framework is presented such that only $M = O(K \log(K))$ measurements are sufficient to ensure stable and robust recovery of bandlimited graph signals $\mathrm{BL}_K(\mathbf{V})$ from their samples. It is shown that the graph weighted coherence $\rho_K = \max_i\{\pi_i^{-1/2} \| \mathbf{V}^T_{(K)}\,\delta_i\|_2\}$ governs the sample complexity for stable and robust recovery. It is then easy to show that the optimal sampling distribution $\{\pi_i^*\}$ that minimizes

the graph weighted coherence ρ_K is $\pi_i^* = \| \mathbf{V}_{(K)}^T \delta_i \|_2^2 / K$ which also corresponds to the statistical leverage scores of $\mathbf{V}_{(K)}$ and can be computed efficiently. In this section, we generalize this random sampling framework to product graphs by exploiting the structure of product graphs. Towards this, we first show how we can compute this optimal sampling score for a given node of the product graph from the optimal sampling scores of that node's corresponding nodes over the graph atoms for signals in $\mathrm{BL}_S(\mathbf{V})$.

Lemma 1. Let $\{\pi^{*(j)}\}$ be the optimal sampling distribution corresponding to the j-th graph atom and $\mathbf{V}_{(R_j)}$ such that $\pi_{i_{(j)}}^{*(j)} = \| \mathbf{V}_{(R_j)}^{(j)T} \delta_{i_{(j)}} \|_2^2 / R_j$. It then follows that the optimal sampling score for a node on the product graph is simply the product of the sampling scores of the corresponding nodes in the graph atom such that

$$\pi_i^* = \frac{\| \mathbf{V}_{(S)}^T \delta_i \|_2^2}{S} = \bigotimes_{j=1}^{J} \frac{\| \mathbf{V}_{(R_j)}^{(j)T} \delta_{i_{(j)}} \|_2^2}{R_j} = \bigotimes_{j=1}^{J} \pi_{i_{(j)}}^{*(j)}. \quad (8)$$

Under the randomized sampling framework over the graph atoms described in Lemma 1, we now provide (optimal) sufficient conditions on the minimum number of samples that ensure a stable embedding of graph signals in $\mathrm{BL}_K(\mathbf{V})$ on the product graph.

Theorem 3. Let $\Psi^{(j)}$ sample M_j nodes according to the sampling distribution $\pi^{(j)*}$ such that $\Psi = \bigotimes_{j=1}^{J} \Psi^{(j)}$ samples $M = \Pi_{j=1}^{J} M_j$ nodes. Let $\mathbf{D}^{(j)}$ be a diagonal rescaling matrix such that $\mathbf{D}^{(j)}_{i_{(j)},i_{(j)}} = 1/\sqrt{M_j \pi_{i_{(j)}}^{(j)}}$ and $\mathbf{D} = \bigotimes_{j=1}^{J} \mathbf{D}^{(j)}$. For any $\delta, \epsilon \in (0,1)$ if,

$$M \geq \frac{3}{\delta^2} S \log(\frac{2K}{\epsilon}),$$

where $S = \Pi_{j=1}^{J} R_j$, we have that with probability atleast $1-\epsilon$, $\Psi\mathbf{D}$ represents a stable embedding for any $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V})$ such that

$$(1-\delta)\|\mathbf{x}\|_2^2 \leq \|\Psi\mathbf{D}\mathbf{x}\|_2^2 \leq (1+\delta)\|\mathbf{x}\|_2^2 \quad (9)$$

Proof. Full proof omitted due to lack of space. The proof is a consequence of Lemma 1 and is in parts constructed similarly to Theorem 3 in [10]. □

Algorithm 1. We recover the original graph signal by solving the following optimization problem:

$$\mathbf{x}_{\mathrm{SP}}^* = \mathbf{V}_{(K)} \arg\min_{\widehat{\mathbf{x}}_{(K)}} \left\| \Psi^T \Psi \mathbf{D}^2 \Psi^T \Psi \mathbf{y} - \mathbf{V}_{(K)} \widehat{\mathbf{x}}_{(K)} \right\|_2^2 = (\bigotimes_{j=1}^{J} \Phi^{(j)}) \mathbf{y}$$

where

$$\Phi^{(j)} = \mathbf{V}_{R_j}^{(j)} \mathbf{U}_{R_j}^{(j)} \Psi^{(j)^T} \Psi^{(j)} \mathbf{D}^{(j)2} \Psi^{(j)T}$$

Hence, we see that we can compose the interpolation operators by only processing the graph atoms. We can now provide lower and upper bounds on the squared error.

Corollary 1. Assume we compose a sampling operator with sufficient samples as proposed in Lemma 1 with respect to the optimal sampling distribution $\{\pi_i^*\}$ and use Algorithm 1 to recover the original signal. We then have, with probability atleast $1-\epsilon$,

$$\frac{1}{M\sqrt{1+\delta}} \|\Psi\mathbf{D}\epsilon\|_2 \leq \|\mathbf{x}_{\mathrm{SP}}^* - \mathbf{x}\|_2 \leq \frac{2}{M\sqrt{1-\delta}} \|\Psi\mathbf{D}\epsilon\|_2$$

Proof. This is a direct consequence of Theorem 6 in [10] because of the restricted isometry property satisfied in Theorem 3. □

Remark 1. Smooth graph signals are bandlimited under a fixed frequency ordering [11]. We can show that with our framework on product graphs, under the Cartesian product, we only need $O(KlogK)$ samples to sample and recover a smooth signal in $BL_K(\mathbf{V})$ which is optimal.

Remark 2. We have seen that we do not need to process the whole product graph $\mathbf{A}$ or compute its spectral decomposition (GFT basis) to construct random sampling and interpolation operators on the product graph. Instead, we only need to compute the spectral decompositions of its graph atoms $\mathbf{A}^{(j)}$ that are of size $O(poly(N^{\frac{1}{J}}))$.

Remark 3. In [11], a class of approximately bandlimited graph signals was defined to be a more general class of graph signals that relaxes the requirement of bandlimitedness, but still promotes smoothness by allowing for a tail after the first K frequency components. It is then shown that sampling with probability distribution $\{\pi^*\}$ and reconstructing according to Algorithm 1 is minimax optimal in terms of both the sample complexity and mean square error rates for the class of approximately bandlimited graph signals.

V. Numerical Experiments

In this section, we test our randomized sampling framework on the product graph $\mathbf{A} = \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)}$ composed over two graph atoms where $\mathbf{A}^{(1)}$ is the Minnesota road graph [16] with $N_1 = 2642$ nodes which we randomly sample according to the optimal sampling distribution illustrated in the heatmap in Figure 2 and the path graph ($N_2 = 8$) which we can uniformly randomly sample. We set $K = 100$ such that $R_1 = 40$ and $R_2 = 3$ and $S = R_1 \times R_2 = 120$ and generate a synthetic bandlimited signal on $\mathbf{A}$ with respect to $\mathbf{V}_{(K)}$. We perform the experiment over varying noise settings by injecting the true signal with white gaussian noise such that the noisy signal we sample from has SNR of 5dB, 10dB or 15dB. We sample M_1 nodes on $\mathbf{A}^{(1)}$ and M_2 nodes on $\mathbf{A}^{(2)}$ with $\Psi^{(1)}$ and $\Psi^{(2)}$ respectively and compose the sampling operator $\Psi = \Psi^{(1)} \otimes \Psi^{(2)}$ to sample $M = M_1 \cdot M_2$ nodes on the product graph $\mathbf{A}$. We recover using the interpolation operators $\Phi^{(1)}$ and $\Phi^{(2)}$ corresponding to each graph atom as described in Algorithm 1. We illustrate the performance of our framework which is consistent with our theoretical analysis in Figure 2 where we plot the reconstruction SNR versus the size of the sample set M averaged over 20 iterations.

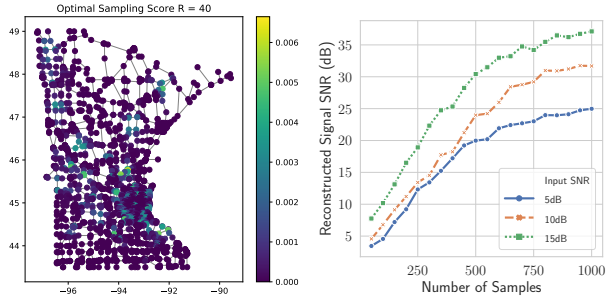

Fig. 2: (a) Optimal sampling scores 8 for $\mathbf{A}^{(1)}$ with $R_1 = 40$ (b) Reconstructed signal SNR vs. number of samples

VI. Conclusion

In this work, we presented a robust randomized sampling and recovery framework for bandlimited signals on product graphs that yields optimal performance in terms of both the reconstruction accuracy and sample complexity. Particularly, we showed that by exploiting the structure of a product graph and designing appropriate sampling and reconstruction operators by only processing the graph atoms that the product graph is composed of, we can achieve significant savings in computational complexity.

REFERENCES

[1] M. Newman, *Networks*. Oxford University Press, 2018.

[2] 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.

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

[4] A. Sandryhaila and J. M. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, 2014.

[5] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 985–1042, 2010.

[6] P. M. Weichsel, "The Kronecker product of graphs," *Proceedings of the American mathematical society*, vol. 13, no. 1, pp. 47–52, 1962.

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

[8] 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.

[9] R. Varma and J. Kovačević, "Sampling Theory for Graph Signals on Product Graphs," *arXiv preprint arXiv:1809.10049*, 2018.

[10] 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.

[11] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Signal recovery on graphs: Fundamental limits of sampling strategies," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 539–554, 2016.

[12] A. Sandryhaila and J. M. Moura, "Discrete Signal Processing on Graphs: Frequency Analysis." *IEEE Trans. Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[13] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.

[14] V. N. Ekambaram, G. C. Fanti, B. Ayazifar, and K. Ramchandran, "Circulant structures and graph signal processing," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*. IEEE, 2013, pp. 834–838.

[15] J. Kovačević and A. Chebira, "An introduction to frames," *Foundations and Trends® in Signal Processing*, vol. 2, no. 1, pp. 1–94, 2008.

[16] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *arXiv preprint arXiv:1408.5781*, 2014.