Downsampling of Signals on Graphs: An Algebraic Perspective

by

NILESHKUMAR VAISHNAV
201121007

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

December, 2018

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Doctor of Philosophy at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Nileshkumar Vaishnav

Certificate

This is to certify that the thesis work entitled DOWNSAMPLING OF SIGNALS ON GRAPHS: AN ALGEBRAIC PERSPECTIVE has been carried out by NILESHKUMAR VAISHNAV for the degree of Doctor of Philosophy at *Dhirubhai Ambani Institute of Information and Communication Technology* under my supervision.

Dr. Aditya Tatu
Thesis Supervisor

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Dr. Aditya Tatu for the continuous support of my study and research. I could not have imagined having a better mentor for my research.

I also take this opportunity to thank Prof. V. P. Sinha, who has been a strong influence in the early years of my work and who has been a guiding force throughout the period of my PhD. I also thank all my colleagues at the institute that helped me with their knowledge and support.

Last but not the least, I would like to thank my wife and my parents for supporting me emotionally and spiritually throughout the period of PhD. I dedicate the thesis to my daughter, who has been a source of inspiration and given the most memorable period in my life.

Contents

Abstract

Real-world data such as weather data, seismic activity data, sensor networks data and social network data can be represented and processed conveniently using a mathematical structure called *Graph*. Graphs are a collection of vertices and edges. The relational structure between the vertices can be represented in form of a matrix called the adjacency matrix. A *Graph Signal* is a signal supported on a given graph. The framework of processing of signals on graphs is called *Graph Signal Processing* (GSP). Various signal processing concepts (e.g. Fourier Transform, filtering, translation, downsampling) need to be defined in the context of graphs. A common approach is to define a Fourier Transform for a graph (called Graph Fourier Transform - GFT), and use it to define other signal processing concepts. There are two popular approaches to define GFT for a graph: 1) using Graph Laplacian 2) using adjacency matrix. In the first method, GFT is interpreted as expansion of a given signal in the eigenvectors of the Graph Laplacian. The second method, i.e., using the adjacency matrix, results in an algebraic framework for graph signals and shift invariant filters.

In the study of Graph Signal Processing, we often encounter signals which are *smooth* in nature. Such signals, which have low variations, can be represented efficiently using samples on fewer number of vertices. The process of selecting such vertices is called *graph downsampling*. As graphs do not exhibit a natural ordering of data, selection of vertices is not trivial.

In this thesis, we analyze a class of graphs called Bipartite Graphs from downsampling perspective and then provide a GFT based approach to downsample signals on arbitrary graphs. For bandlimited signals on a graph, a test is provided to identify whether signal reconstruction is possible from the given downsampled

signal. Moreover, if the signal is not bandlimited, we provide a quality measure for comparing different downsampling schemes. Using this quality measure, we propose a greedy downsampling algorithm. The proposed method is applicable to directed graphs, undirected graphs, and graphs with negative edge-weights. We provide several experiments demonstrating our downsampling scheme, and compare our quality measure with other existing measures (e.g. cut-index). We also provide a method to assign adjacency matrix to the downsampled vertices using an analogy from the bipartite graphs.

We also examine the concepts of homomorphism and isomorphism between two graphs from signal processing point of view, and refer to them as GSP-isomorphism and GSP-homomorphism, respectively. Collectively, we refer to these concepts as Structure Preserving Maps. The fact that linear combination of signals and linear transforms on signals are meaningful operations has implications on the GSP-isomorphism and GSP-homomorphism, which diverges from the topological interpretations of the same concepts (i.e. graph-isomorphism and graph-homomorphism). When Structure Preserving Maps exist between two graphs, signals and filters can be mapped between them while preserving spectral properties. We examine conditions on adjacency matrices for such maps to exist. We also show that isospectral graphs form a special case of GSP-isomorphism and that GSP-isomorphism and GSP-homomorphism is intrinsic to resampling and downsampling process.

List of Tables

List of Figures

CHAPTER 1

Introduction

A significant amount of research in the field of Digital Signal Processing (DSP) is focused on processing regularly sampled signals such as speech, audio, image and video. We refer to this body of work by the phrase *Classical Signal Processing* (CSP). Key concepts involved in CSP framework include Discrete Fourier Transform (DFT), filtering, downsampling, multirate signal processing, multiresolution analysis, and others [25].

Despite the wide scope of the CSP framework, there exist real-world data, which can not be represented suitably in the CSP framework. Examples of such data include weather data, seismic activity data, sensor networks data, social network data, transportation data etc. Graph provides an appropriate model to represent such data. Given the large scope of applications [36], analysis and processing of signals on graph is important. For signals on graphs, there is no natural ordering of the samples, rather the inter-relations between them are important. Defining concepts such as shift, Fourier transform, convolution, downsampling and wavelets is not trivial and diverges from similar concepts for classical signal processing.

A graph is a collection of vertices with a given relation structure between the vertices. Graphs are represented using a set of vertices (also called nodes) and edges. The edges of a graph provide the relation between vertices and it can be represented by a matrix called the *graph adjacency matrix*. For a graph with N nodes, the adjacency matrix is an $N \times N$ matrix with entries representing the edges of the graph. To understand adjacency matrix structures, we consider some examples of social networks. Consider a social network where every individual is

represented by a vertex and existence of an edge between to vertices indicates that the two individuals are friends (e.g. Facebook). In this graph, every existing edge has identical weight. Such a graph is called an unweighted graph. Thus, for an unweighted graph, the adjacency matrix has binary entries (1 representing presence of an edge, 0 representing absence). If the aforementioned friendship social network also has levels of friendship (e.g. acquaintance, colleague, close friends, etc...), we require the edge to have multiple values, which can be represented as a real number called edge-weight. If we require to model connections where two individuals are negatively connected (e.g. two people having different ideological leanings), negative edge-weights may also be required. Whether weighted or unweighted, the friendship social network graph described is still undirected, as the edge is defined by a property that is identical in either direction, i.e., person A being friends with B is same as person B being friends with A. For an undirected graph, the adjacency matrix is symmetric. In some other networks, the relation may be directed. For example, a social network where the defining relationship is 'follow' (e.g. Twitter). If person A is following person B, it does not necessarily mean that person B is also following person A. Thus giving rise to directed graphs. For further details on graphs, refer [5].

Traditionally, spectral properties of graph signals are derived using *graph Laplacian*, which is defined for undirected graphs. The study of eigenvalues and eigenvectors of graph Laplacian is called *Spectral Graph Theory* [9]. The eigenvectors of graph Laplacian are useful in examining some topological properties of a graph. The eigenvector associated with the second smallest eigenvalue is called the Fiedler Vector, and it is used in partitioning of graphs [26]. An approach presented in [32] indicates that the spectral analysis of graph signals can also be carried out effectively using the graph adjacency matrix. This approach allows us to work with signals on directed graphs, which is not possible with Graph Laplacian based approach. The adjacency matrix based processing on graphs result in an algebraic structure, which connects the graph signal processing theory with Algebraic Signal Processing theory [29]. CSP can be viewed as a special case of Graph Signal Processing with an appropriate choice of adjacency matrix [29] [27] [32].

In this chapter, we review algebraic signal processing and graph signal processing frameworks. We discuss the problem of downsampling of signals on graphs and review some existing methods of graph downsampling. We also discuss structure preserving maps on graphs. Then, we summarize the contributions of the thesis.

1.1 Background

In this section, we review the Graph Signal Processing framework. We define signals and filters on graph, and discuss popular approaches for defining the Graph Fourier Transform (GFT) and provide a brief introduction to algebraic signal processing on graphs.

As noted earlier, a graph is the collection of vertices and edges. A graph G is denoted as $(\mathcal{V}, A)$, where $\mathcal{V}$ is the set of vertices $\{v_1, ..., v_N\}$ (we assume the order as mentioned) and A is the graph adjacency matrix which provides the relation structure between the set of vertices. While the order of the vertices is necessary for representation, it has no effect on the inter-relations between the vertices[1]. For adjacency matrix A, each element $a_{i,j}$ is the weight connecting vertex v_j to vertex v_i. The degree d_i of a vertex v_i is defined as the sum of all the edge-weights connecting to v_i. For undirected graphs, *Graph Laplacian* is defined as $L = D - A$, where D is a diagonal matrix with ith diagonal entry being the degree of vertex v_i. The *Normalized Graph Laplacian* is defined as $L_n = D^{-1/2}LD^{-1/2}$. A *graph signal* on a graph G is defined as the vector $\bar{s} = [s_1, s_2, \cdots, s_N]^T$, where s_i is a scalar value on vertex v_i. Thus a signal $\bar{s}$ can be viewed as an element in $\mathbb{C}^N$. Figure 1.1 shows a sample graph along with a signal on the graph. With the given definitions of graph signal and graph adjacency matrix, we now discuss two popular approaches used to define the Graph Fourier Transform.

Graph Laplacian Based GFT: In CSP, the Fourier Transform of a signal can be viewed as expansion of the signal into the eigenvectors of Laplacian. Analogously, Graph Laplacian can be used to define the Graph Fourier Transform for a given

[1] If we change the vertex order using a permutation matrix P, then the adjacency matrix changes to PAP^T.

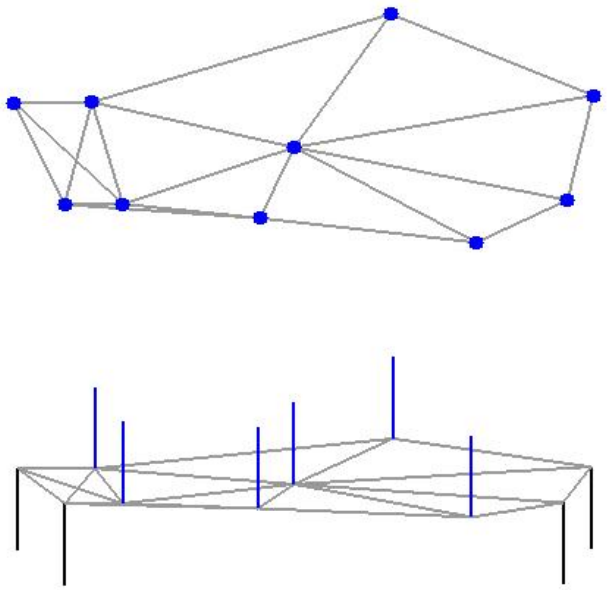

Figure 1.1: (Top) A sample graph with 10 vertices. (Bottom) A signal on the given graph

undirected graph. Thus, expanding a graph signal in terms of eigenvectors of graph Laplacian can provide a way to determine the GFT. Since Graph Laplacian is a symmetric matrix, it is always diagonalizable, and thus eigenvectors span the entire signal space. Specifically, given Graph Laplacian L and its diagonalization $V\Sigma V^T$, where Σ is a diagonal matrix and V is an orthogonal matrix, V^T is the designated *Graph Fourier Transform based on Laplacian*, denoted by GFT_L. Columns of matrix V are the eigenvectors of L, which are the basis for GFT representation of a graph signal. Alternatively, Normalized Graph Laplacian can also be used in place of Graph Laplacian in order to obtain GFT, which is denoted as GFT_N. In the case of GFT_L and GFT_N, the ascending frequency order corresponds to the ascending order of the respective eigenvalues (which are all real). Note that both these approaches of defining GFT are applicable only in case of undirected graphs.

Graph Adjacency Matrix Based GFT: As the Graph Laplacian based approach is limited to undirected graphs, adjacency matrix based approach is also used in graph signal processing. In CSP, the Fourier Transform can be viewed as expansion of a signal in terms of the eigenvectors of the shift operator. Thus, for adja-

cency matrix based approach, the adjacency matrix is axiomatically designated as the shift operator for a given graph. Unlike Graph Laplacian, the adjacency matrix is not always diagonalizable, thus the eigenvectors may not span the entire signal space. Following [32], we use the Jordan Normal Form (JNF) and *generalized eigenvectors* in order to define the GFT. Let $A = VJV^{-1}$ be the Jordan Canonical Decomposition, with J being the Jordan Canonical Form for the adjacency matrix. In this case, V^{-1} is designated as the *Adjacency matrix based Graph Fourier Transform*, denoted as GFT_A.

Once we obtain the Graph Fourier Transform in form of a matrix F, the GFT of a graph signal $\bar{s}$ can be computed using $F\bar{s}$. In this thesis, unless specified, we assume Adjacency Matrix based GFT. We now elaborate the adjacency matrix based approach and the algebraic structure related to the same.

1.1.1 Algebraic Signal Processing Theory

Traditionally, linear signal processing (i.e. processing of signals using linear transforms) has been studied and explored in the light of vector spaces. In this traditional model, the signals are modeled as vectors and linear systems (also referred to as filters) are modeled as linear operators operating on these vectors. This interpretation provides tools to design, implement and analyze various signal processing systems. A recently proposed algebraic interpretation of linear signal processing called Algebraic Signal Processing Theory [27, 29] (abbreviated as ASP) explores algebraic structures [17] present in the interplay of signals and linear systems such as algebra, module and isomorphism between them. This abstraction provides us with generalized notions of z-Transform, Fourier transform, spectrum, frequency response, convolution etc.

We focus on Linear and Shift-Invariant systems, which are referred to as LSI systems. The concept of shift-invariance requires a shift operator to be defined on the signal space. With a given shift operator, any filter that commutes with the shift is called a shift-invariant filter. The set of LSI filters allow for addition, multiplication and scaling of filters. Thus, it exhibits the structure of algebra. In addition, the algebra of filters can be generated using the shift operator. The alge-

bra of LSI filters is denoted as $\mathcal{A}$. Similarly, the space of signals is a vector space. Moreover, the signals can be operated upon by the LSI filters. Thus, the space of signals can be modeled as $\mathcal{A}$-module of the algebra of filters[2]. It is denoted as $\mathcal{M}$. A one-to-one and onto mapping Φ is the generalized z-transform which maps the signal vector space to the module. The triplet $(\mathcal{A}, \mathcal{M}, \Phi)$ is called the algebraic signal model.

Each signal model has its own notion of Fourier transform. For example, in ASP framework, both DCT and DFT are Fourier Transforms which correspond to two different shift operators. In traditional discrete-time signal processing, signals are represented as vectors and systems as matrices. Linearity and shift-invariance makes the matrix representation of the system to be circulant. The DFT, in case of finite discrete-time signals, is simply a change of basis. The DFT changes the basis in such a way that the circulant matrix (representing the system) is diagonalized. This diagonalization of the matrices, converts convolution (or circulant matrix multiplication) to point-wise multiplication (multiplication between diagonal matrices). In the ASP framework, the Chinese Remainder Theorem is used to obtain the Fourier Transform. Example 1 explains the algebraic signal processing model for a periodic discrete signal.

Given an algebra $\mathcal{A}$ and corresponding $\mathcal{A}$-module $\mathcal{M}$, if $\mathcal{M}' \subset \mathcal{M}$ is such that $\mathcal{M}'$ is also an $\mathcal{A}$-module, then $\mathcal{M}'$ is called an $\mathcal{A}$-submodule. Note that the empty set and the set $\mathcal{M}$ are always submodules, hence a proper submodule is a submodule besides the two. If a submodule does not have any *proper* submodule, then it is called an irreducible submodule. Every one dimensional submodule is an irreducible submodule. The irreducible submodules of a given module provide a way to define the Fourier Transform in the algebraic framework. It is possible to represent the signal module $\mathcal{M}$ as a direct sum of irreducible submodules $\mathcal{M}_\omega$. This decomposition is called the Fourier Transform. In [31] [30] [42] [28], the algebraic signal processing models are used to obtain Cooley-Tukey [10] type

[2]It must be emphasized here that every vector space is also a module. However, the space of signals is a vector-space with respect to the field of scalars. At the same time, it is an $\mathcal{A}$-module with respect to the algebra of the LSI filters.

algorithms for various spectral transforms.

Example 1. Consider a periodic sequence $\bar{s}$ with period $N = 6$, i.e., $\bar{s} \in \mathbb{C}^6$. Let the shift operator on the signal space be defined as a unit periodic shift operator (denoted as $\mathcal{D}$). Thus, given $\bar{s} = [s_0, s_1, \cdots, s_5]^T$, the unit-shift on $\bar{s}$ produces $\mathcal{D}\bar{s} = [s_5, s_0, s_1, \cdots, s_4]^T$. In matrix form, the shift operator is given by

$$\mathcal{D} = \begin{bmatrix} & & & & & 1 \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \end{bmatrix}$$

It can be seen that any arbitrary signal $\bar{s} = [s_0, \cdots, s_5]^T$ can be written as $\bar{s} = \Sigma_{i=0}^{5} s_i \mathcal{D}^i \bar{\delta}$, where $\mathcal{D}^i$ is $i-$fold shift and $\bar{\delta}$ is the usual discrete impulse sequence. Similarly, any LSI filter can also be written as a polynomial in $\mathcal{D}$, i.e., filter $H = h(\mathcal{D})$. The shift operator $\mathcal{D}$ is generator of the algebra of all the LSI filters. Using map $\mathcal{D} \mapsto x$, signal $\bar{s}$ and LSI filter H can be mapped to polynomials $s(x)$ and $h(x)$ respectively. Filtering can be performed as $h(x)s(x) mod\ m_{\mathcal{D}}(x)$, where $m_{\mathcal{D}}(x) = x^6 - 1$ is the minimal polynomial of $\mathcal{D}$. In this example, $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/m_{\mathcal{D}}(x)$.

It is possible to view the signal model presented here as a graph by designating the shift operator as the adjacency matrix. The graph representation thus obtained is shown in Figure 1.2. Each vertex is weighted by a polynomial, which is used to convert a given signal into a polynomial. It should be noted here that the matrix that diagonalizes $\mathcal{D}$, is the DFT matrix. Thus the graph presented in Figure 1.2 has the DFT as its GFT. This example explains how to map signals and LSI filters onto polynomials. This principle is extended for arbitrary graphs in the next subsection.

1.1.2 Algebraic Graph Signal Processing

Let us review some important results related to algebraic signal processing on graphs, which will be useful in later chapters. The results are summarized for a

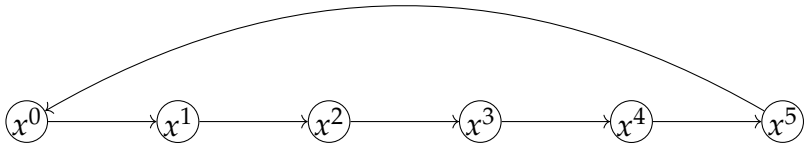

Figure 1.2: 1-D Uniform Directed Graph And Weighting Polynomials

graph $G = (\mathcal{V}, A)$ with $|\mathcal{V}| = N$. Detailed discussion on the results along with proofs can be found in [32, 34]. This approach rests on the foundations of algebraic signal processing theory (ASP) described in [27, 29]. The following proposition (refer Proposition 12.4.1 in [18] for a proof) is useful in deriving the algebraic structure of graph filters.

Proposition 1 *Given $H, A \in \mathbb{C}^{N \times N}$ such that $HA = AH$, then H can be represented as a polynomial in A provided that the characteristic and minimal polynomials of the matrix A are identical. We can write $H = p(A)$, where $p(x)$ is a polynomial of degree at most $N - 1$.*

We summarize the important results (along with assumptions) required for the thesis, below.

1. The adjacency matrix A is designated as the shift operator. The characteristic polynomial of A is denoted by $p_A(x)$ while the minimal polynomial is denoted by $m_A(x)$. Here, we will assume that $m_A(x) = p_A(x)$ [3].

2. A matrix $H \in \mathbb{C}^{N \times N}$ represents a linear transform (or filter) that operates on a given graph signal. If a filter H is shift-invariant, then $HA = AH$. Using Proposition 1, all Linear and Shift Invariant (LSI) filters can be expressed as polynomials in adjacency matrix A assuming $m_A = p_A$. Let $\mathcal{A} = \mathbb{C}[x]/p_A(x)$, the set of polynomials in x with multiplication defined as modulo-$p_A(x)$. Let $\mathcal{F}$ be the space of filters. As stated earlier, given a shift-invariant filter H, it can be represented as a polynomial in A (as $m_A(x) = p_A(x)$), i.e. $H = h(A)$, where h is a polynomial. Now, using the map $A \mapsto x$, we get $H = h(A) \mapsto h(x)$, and thus, $\mathcal{F} \cong \mathcal{A}$ (read as, $\mathcal{F}$ is isomorphic to $\mathcal{A}$). This isomorphism is called the *graph z transform*.

[3] The case where $m_A(x) \neq p_A(x)$ can be handled as described in [32]

3. As noted earlier, given the Jordan Normal Form of adjacency matrix A as VJV^{-1}, the matrix V^{-1} is the GFT for the given graph. The spectrum of a signal $\bar{s}$ is computed as $V^{-1}\bar{s}$.

4. The signal space S is isomorphic to an $\mathcal{A} - module\ \mathcal{M}$ given by $\mathcal{M} = \mathbb{C}[x]/p_A(x) = \{s(x)|s(x) = \sum_{n=0}^{N-1} s_n b_n(x)\}$ where $\bar{s} = (s_0, ..., s_{N-1}) \mapsto s(x)$. Here $b_n(x)$ are polynomial basis which can be computed using adjacency matrix. Thus every vertex of the graph is weighted by a polynomial which allows a signal to be represented in form of a polynomial. If the eigenvalues of A are distinct, then for a signal $\bar{s} \mapsto s(x)$, the spectrum is given by $(s(\lambda_0), ..., s(\lambda_{N-1}))$, where λ_is are eigenvalues of A.

5. If signal $\bar{s} \mapsto s(x)$ is filtered by $H \mapsto h(x)$, then filtered signal $\tilde{s} = H\bar{s} \mapsto \tilde{s}(x)$ is given by, $\tilde{s}(x) = h(x)s(x)\ mod\ p_A(x)$.

In order to obtain the frequency ordering, we use Total Variation (TV), defined in [34] as follows. For a given graph G,

$$TV_G(\bar{s}) = \|\bar{s} - A_{norm}\bar{s}\|_1,$$

where $A_{norm} = \frac{1}{|\lambda_{max}|}A$ is the normalized shift operator and $|\lambda_{max}|$ is the maximum of the absolute eigenvalues of matrix A. This definition is derived using analogy from classical signal processing where Total Variation of a signal is defined as the absolute sum of the first order difference of a given signal, i.e., $TV(\bar{s}) = \|s - \mathcal{D}\bar{s}\|_1$, where $\mathcal{D}$ is the shift operator for CSP. Under the assumption that $\lambda_{max} = \|A\|_2$, normalization of A avoids the attenuation/amplification of the signal while shifting. This assumption is true in case of undirected graphs. If a signal has a higher total variation, it is considered to have larger high frequency content than a signal with lower total variation. For an eigenvector $\bar{e}$ of matrix A with eigenvalue λ, the total variation is given by $|1 - \frac{\lambda}{|\lambda_{max}|}|\|\bar{e}\|_1$. Thus eigenvectors of A can be ranked using the corresponding eigenvalues. This provides a frequency order on the GFT. A convention is to order the eigenvectors in ascending order of frequency.

We now introduce the central problem considered in the thesis, that of downsampling graph signals. Following this, we discuss relations between graphs from a signal processing point of view.

1.2 Downsampling of Signal on Graphs

With the graph signal processing framework defined, we proceed to the problem of downsampling of signals on graphs, which forms a key part of the thesis. Often, we encounter signals on graph which are *smooth* in nature. Due to lower amount of total variation, such signals exhibit low-pass behavior in spectral domain. When a graph signal does not contain frequency content above a certain cut-off frequency, it is called a *bandlimited signal*. If the graph signal is bandlimited, it can be reconstructed from fewer samples in vertex-domain. The process of finding the collection of vertices which can reconstruct the original signal is called *graph downsampling* (other phrases referring to similar process include *graph coarsening* [12, 19] and *site percolation* [14]). Graph downsampling can be used for compression and as a building block for multiresolution analysis for signals on graph [15].

Downsampling of an arbitrary graph with respect to bandlimited signals diverges from CSP and its downsampling process. In CSP, there is an ordered set of vertices, and the downsampling process amounts to selecting every alternate vertex from the given set of vertices. From a spectral perspective, the selection of every alternate vertex results in folding of spectrum exactly by a factor of two. If the signal is bandlimited with upper-half of frequency content absent, then spectral folding does not introduce any aliasing. Thus, any signal which has no spectral content on the upper-half of the frequency spectrum can be recovered from the downsampled vertices without any error. Thus the spectral view of the signal coincides with the topological view in case of classical signal processing. Downsampling on graph, however, differs from the traditional view in both the domains (the vertex domain and the spectral domain). This is because of the fact that a graph does not provide any topology in which the vertices are ordered (except in

special cases), hence *selecting every alternate vertex* is not a meaningful operation. Moreover, the spectrum of a graph (i.e. eigenvalues of Laplacian/adjacency matrix) does not necessarily show symmetry, indicating that the spectral-folding phenomena is not the same as that in classical signal processing. Another challenge in downsampling on graphs is how to determine the inter-relations among the reduced set of vertices. The determination of new adjacency relation in the reduced graph is essential to obtain a multi-resolution decomposition on graph [11,24,43].

In this thesis, we analyze the downsampling process on bipartite graph using algebraic signal processing theory on graphs. In Chapter 3, we obtain a sampling scheme for an arbitrary graph which takes into account the spectral properties of the graph in order to downsample signals. The approach presented can be applied to both directed as well as undirected graphs. In case the signal is not bandlimited, we provide a measure that allows to choose a scheme with minimum reconstruction error. We provide a greedy algorithm to solve the optimization problem and analyze the methods to improve the computational complexity of the algorithm. Using similar approach, we provide a method to design a frequency band sensitive approach for downsampling of a graph signal.

1.2.1 Related Work

One key problem in graph downsampling is determining a sample-set, i.e., the set of vertices from which a bandlimited signal can be recovered without any error. If the signal is bandlimited in spectral domain, then it can be downsampled without loss of data. However, downsampling is often used for non-bandlimited signals which have negligible (but non-zero) high frequency content. The error introduced due to non-zero high frequency content gives rise to aliasing. It should be emphasized here that the sample-set (of a given cardinality) for a given bandwidth is not unique, and an algorithm may converge to one of those sample-sets. However, different sample-sets have different sensitivity to aliasing in case of signals which are not bandlimited, which indicates that even among sample-sets, the quality of signal-reconstruction differs. The methods to downsample graphs can be divided into two broad categories: Topological and Spectral.

Topological approaches rely on neighborhood properties of vertices in order to downsample the graph. One major class of graphs is called *bipartite graphs*, which provide a natural way to downsample. An analysis of downsampling k-regular bipartite graphs is provided in [23]. However, not all graphs exhibit bipartite structure, so to apply the downsampling to arbitrary graphs, a method proposed in [22] locally approximates the bipartite structure using a graph-colouring technique. On the other hand, Nguyen and Do [24] rely on Maximum Spanning Tree(MST) of a graph in order to downsample[4]. A major limitation with topological approaches is that although the signal is assumed to be bandlimited in spectral domain, the actual process of finding the downsampling scheme does not take into account spectral properties of the graph in a direct way. Moreover, these approaches cannot be applied to downsample a directed graph, or a graph with negative edge-weights.

In contrast to topological approaches, the spectral approaches rely on the spectral properties of the graph in order to downsample the same. The eigenvector corresponding to highest frequency is used to obtain a downsampling scheme in [4]. Based on polarity of eigenvector values, two equivalent sets of downsampled vertices are obtained. Another method to determine the sample-set of an undirected graph is provided by Anis *et al.* [2], in which a greedy approach is used to add a vertex in every iteration to the sample-set, which provides the highest increase in bandwidth, until the cut-off threshold is reached. The method presented in [8] obtains a stable frame for reconstruction in order to minimize the error in samples.

Another issue in graph downsampling is measuring the quality of the affected partition on graph. Cut-index [24] is one of the objective measure used to determine quality of the graph downsampling scheme, and is defined as the ratio of sum of edgeweights of edges to be deleted in order to disconnect two selected partitions, and the total edgeweights in the graph. A downsampling scheme with higher cut-index is considered to have better quality (and hence better signal reconstruction properties). One major issue with cut-index is that a single cut pro-

[4]It should be noted here that every tree is a bipartite graph.

vides us two downsampling options (i.e. both partitions are considered equally good), selecting one of the two is an arbitrary choice.

1.3 Structure Preserving Maps on Graphs

Graph topology and relations between two graphs have been studied in detail in past, e.g. graph-isomorphism and graph-homomorphism are well-researched concepts in graph theory [5, 16, 21]. We focus primarily on the signal processing implications of such relations. We define homomorphisms and isomorphisms between graphs from signal processing point of view, and refer to them as *GSP-isomorphisms* and *GSP-homomorphisms*. GSP-isomorphism and GSP-homomorphism are collectively referred to as *Structure Preserving Maps*. We know that graph-isomorphism defines equivalence between graphs upto re-ordering of the vertices. We define GSP-isomorphism, which aims to extend the concept. One example of GSP-isomorphism is the graph relation resulting from diagonalization of the adjacency matrix (assuming the matrix to be diagonalizable). We may associate a graph to the diagonalized spectral domain, where there are no interconnections of vertices; however, each vertex has a self-loop. From signal processing perspective, the filtering operation can be carried out in either of the two representations with the same frequency response.

1.4 Contribution

In this thesis, we explore the problem of downsampling on graphs. We also analyze the structure preserving maps between graphs from signal processing point of view. The contributions of the research work are summarized below.

- Bipartite graphs are closely related to downsampling process. We analyze the downsampling on bipartite graphs from algebraic perspective and establish that downsampling on bipartite graphs exhibit several features (such as optimality of downsampling, basis polynomial structures and aliasing) which are also true for downsampling on classical signal processing.

- We provide a downsampling measure named SVD based Downsampling Quality Measure (SDQM), to measure the quality of a given downsampling scheme. Based on SDQM, we propose an algorithm that downsamples an arbitrary graph. We use the proposed algorithm to analyze the downsampling of graphs which are related to Discrete Cosine Transform. We optimize the proposed algorithm for computational complexity.

- We analyze the downsampling method further with respect to various bands in high frequency. We propose an approach to tune the algorithm in order to obtain a downsampling scheme with desired frequency sensitivity.

- We analyze structure preserving maps between two graphs (called GSP-isomorphism and GSP-homomorphism). We analyze how GSP-isomorphism and GSP-homomorphism are related to resampling and downsampling process, respectively.

1.5 Organization of Thesis

We begin by analyzing downsampling of bipartite graphs in the context of algebraic graph signal processing in Chapter 2. In Chapter 3, we provide a method (with a greedy algorithm) to downsample an arbitrary graph. In Chapter 4, we explore ways to reduce the complexity of the greedy algorithm and provide a method to downsample graphs with different sensitivity towards different frequency bands. We discuss GSP-isomorphism and GSP-homomorphism in Chapter 5, and we conclude the thesis in Chapter 6.

CHAPTER 2

Downsampling of Bipartite Graphs: Algebraic Analysis

The bipartite graphs are often used in the context of graph downsampling, notably in [23], [2], [22]. In [23], downsampling of a special class of bipartite graphs (called k-regular bipartite graphs) is analyzed from spectral perspective, which is extended in [22] to arbitrary graphs. Thus, it is pertaining to analyze the downsampling of bipartite graphs. We analyze the downsampling of bipartite graphs from algebraic perspective. We analyze the spectral properties of bipartite graph using adjacency matrix based GFT (as opposed to graph Laplacian based GFT used in prior work).

In this chapter, we show that the selection of a bipartite partition is optimal downsampling using different measures. We also show that some of the popular algorithms for graph downsampling converge to the selection of bipartite partition. We analyze the aliasing properties of downsampling of bipartite graphs. We begin by defining key terms which are used frequently in the chapter. We also demonstrate how the downsampling on bipartite graphs shares many key features to downsampling in classical signal processing.

Definition Given a graph $G = (\mathcal{V}, A)$, the process of selecting a proper subset $\mathcal{W} \subset \mathcal{V}$ is called *graph downsampling*. $\mathcal{W}$ is called the *downsampled vertex-set* or *reduced vertex-set*. The signal values defined on the set $\mathcal{W}$ is called the *downsampled signal*.

Definition For a graph $G = (\mathcal{V}, A)$, if there exist two sets of vertices $\mathcal{V}_k$ and $\mathcal{V}_p$ with conditions $\mathcal{V}_k \cup \mathcal{V}_p = \mathcal{V}$ and $\mathcal{V}_k \cap \mathcal{V}_p = \varnothing$, such that there are no edges

between any two vertices in either $\mathcal{V}_k$ or $\mathcal{V}_p$, then graph G is called *bipartite graph*. The sets $\mathcal{V}_k$ and $\mathcal{V}_p$ are called partitions of bipartite graph G. We denote a bipartite graph as $(\mathcal{V}_k, \mathcal{V}_p, A)$.

Definition A bipartite graph with the degree of each vertex being k, is called a k-regular bipartite graph.

In the next section, we focus on downsampling by a factor of two and analyze the spectral properties of the same in context of adjacency matrix based GFT.

2.1 Analysis of Downsampling of Bipartite Graphs

Intuitively, downsampling of a bipartite graph is achieved by selecting signal values on a bipartite partition. Let us first obtain the GFT of a bipartite graph in order to analyze the spectral and aliasing properties of such a downsampling process. We focus on downsampling by a factor two for undirected bipartite graphs, with same number of vertices in partitions $\mathcal{V}_k$ and $\mathcal{V}_p$. If the vertices are ordered according to partitions $\mathcal{V}_k$ and $\mathcal{V}_p$, then the structure of the adjacency matrix A is as follows

$$A = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix} \tag{2.1}$$

where B is an $n \times n$ matrix, while A is an $N \times N$ matrix, with $N = 2n$.

Let the SVD (Singular Value Decomposition) of matrix B be given by $B = U\Sigma V^*$, where U and V are orthogonal matrices and Σ is a diagonal matrix containing singular values in ascending order. Following [9], the matrix A can be diagonalized as

$$A = W\Lambda W^* \tag{2.2}$$

where $W = \frac{1}{\sqrt{2}} \begin{bmatrix} U & U \\ V & -V \end{bmatrix}$ and $\Lambda = \begin{bmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{bmatrix}$. With the frequency order-

ing mentioned in [34], it can be seen that the block corresponding to $-\Sigma$ contains the upper-half of frequencies.

In Subsection 2.1.1, we establish optimality of bipartite graph downsampling using various optimization parameters as well as using different existing algorithms. This discussion throws light on reconstruction errors, aliasing and adjacency relations for downsampled vertices. We also explain how the algebraic properties of bipartite-downsampling resembles their classical signal processing counterparts. We demonstrate how some important properties related to downsampling in classical signal processing carry forward to bipartite graphs.

2.1.1 Optimality of Bipartite Partition Selection with respect to Downsampling

Graph downsampling can also be seen as an optimization problem. One parameter to optimize is maximization of normalized cut-index. As there is no known polynomial-time complexity algorithm to solve the optimization problem, SVD based approach [24] relies on an approximate solution, which uses polarity of highest frequency eigenvector for downsampling. Another parameter to maximize is stability of reconstruction frame [8], wherein a stable reconstruction frame is desired as it is less susceptible to noise during reconstruction process.[1]. We define downsampling quality measures *normalized cut-index* and *reconstruction frame-stability* and show that selecting one of the bipartite partition optimizes the same in case of bipartite graphs.

Definition For graph $G = (\mathcal{V}, A)$, let sets $\mathcal{V}_k$ and $\mathcal{V}_p$ form a partition on set $\mathcal{V}$. Let the total edge-weight for graph be w_e while the sum of weights of edges connecting vertices in $\mathcal{V}_k$ to $\mathcal{V}_p$ be w, then the ratio $\frac{w}{w_e}$ is called *normalized cut-index* for partition $(\mathcal{V}_k, \mathcal{V}_p)$. Ideal value of this measure is 1.

Definition For graph $G = (\mathcal{V}, A)$, let sets $\mathcal{V}_k$ and $\mathcal{V}_p$ form a partition on set $\mathcal{V}$. Let V^{-1} be the GFT for the graph. Let V_L denote the eigenvector matrix for preserved

[1] In Chapter 3, we introduce a metric named *SDQM* for measuring the quality of downsampling scheme.

low frequencies. If we denote selection of vertices for partition $\mathcal{V}_k$ by Ψ_k, then the condition number for submatrix $\Psi_k(V_L)$, is defined as *reconstruction frame-stability* (RFS) for partition $\mathcal{V}_k$. Ideal value of this measure is 1.

The normalized cut index measures the amount of edges to be removed in order to affect a partition on graph. Intuitively, maximization of normalized cut index is akin to rejecting strongly connected neighbors to a given vertex for the selection of partition.[2] The second parameter defined above, RFS, measures the stability of reconstruction frame. Maximization of RFS is necessary because an unstable reconstruction frame results is susceptible to noise resulting in unfaithful reconstruction. The following proposition establishes that selecting bipartite partition (for downsampling of a bipartite graph) optimizes normalized cut-index and RFS parameters.

Proposition 2 *For a bipartite graph $(\mathcal{V}_k, \mathcal{V}_p, A)$, selecting a bipartite partition (i.e. either $\mathcal{V}_k$ or $\mathcal{V}_p$) maximizes the normalized cut-index and reconstruction frame-stability.*

Proof When a bipartite partition is selected, the graph is divided into partitions $\mathcal{V}_k$ and $\mathcal{V}_p$. The only edges that exist in the graph are the edges that connect vertices in $\mathcal{V}_k$ to vertices in $\mathcal{V}_p$. Hence, normalized cut-index of this partition is 1, which is the maximum possible value.

RFS is computed as condition number of submatrix $\Psi_k(V)$. The rows of matrix V correspond to vertices, while the columns correspond to frequency components. Thus, a submatrix of V is a matrix that selects vertices and frequency components. As already noted, the matrix V for bipartite graph is of form $\frac{1}{\sqrt{2}} \begin{bmatrix} U & U \\ V & -V \end{bmatrix}$. A selection on bipartite partition on lower half of spectrum gives us a submatrix U or V (ignoring the scalar $\frac{1}{\sqrt{2}}$). As both these matrices are orthogonal, the condition number is optimal. ∎

For arbitrary graphs, there are suboptimal algorithms for optimizing these quality measures. We select MST-based downsampling [24], downsampling using highest frequency eigenvector and greedy vertex selection using RFS [6, 8] as

[2] On the other extreme, we can minimize the normalized cut index in order to achieve segmentation on a given graph. [26]

representative algorithms and prove that the given algorithm converge to selection of bipartite partition when applied to a bipartite graph. The proofs can also be found in [40].

Proposition 3 *For a bipartite graph $(\mathcal{V}_k, \mathcal{V}_p, A)$, following algorithms for downsampling graphs converge to selection of a bipartite partition. For algorithms to be applicable, we assume that the edge-weights are non-negative.*
(1) MST based downsampling (2) Downsampling based on polarity of eigenvector corresponding to highest frequency (3) Greedy vertex selection using Reconstruction Frame Stability

Proof (1) MST based downsampling approach approximates the given graph by its MST, and then divides the vertices into two partitions with even and odd distance from root node. For a given bipartite graph, the distance between any two nodes in two different bipartite partitions is odd, and the distance between any two nodes within the same bipartite partition is even. Thus, this approach yields selection of bipartite partition as downsampled graph.
(2) For a graph with non-negative edge-weighs, the matrix B also has all non-negative entries. If σ_m is the largest singular value of B, then the eigenvector of A corresponding to eigenvalue $-\sigma_m$ is the highest frequency eigenvector, which is given by $\begin{bmatrix} \bar{u} \\ -\bar{v} \end{bmatrix}$. Here $\bar{u}$ and $\bar{v}$ are left and right singular vectors of matrix B. We know that matrix $\sigma\bar{u}\bar{v}^T$ is the optimal rank-1 approximation for matrix B, hence, respective entries of $\bar{u}$ and $\bar{v}$ have same signs. Similarly, all entries on singular vector $\bar{u}$ also have same signs, as a change in sign would result in a non-optimal rank-1 solution. Hence, all entries of singular vectors $\bar{u}$ and $\bar{v}$ share identical signs. Using this fact, it can be shown that the highest frequency eigenvector based approach provides the bipartite partition as downsampled vertex-set for given bipartite graph.
(3) The greedy algorithm that attempts to maximize RFS starts with an arbitrary node and selects the rows of matrix $\begin{bmatrix} U \\ V \end{bmatrix}$ that maximizes minimum singular value of the selected submatrix. Thus, such an algorithm will select orthogonal

columns as long as they are available. Here, As both U and V are orthogonal matrices, orthogonal columns are available until the downsampling process is complete. Hence, the greedy approach converges to selection of a bipartite partition. ∎

2.1.2 Spectral Folding for Downsampling of Bipartite Graphs

Now, let us explore how the spectrum behaves when we downsample a signal on a bipartite partition. The objective is to examine the aliasing behavior on bipartite graphs. Let the graph be $(\mathcal{V}_k, \mathcal{V}_p, A)$ and let the graph signal be denoted as $\bar{s}$. Let the downsampled version of $\bar{s}$ on vertex-sets $\mathcal{V}_k$ and $\mathcal{V}_p$ be $\bar{s}_k$ and $\bar{s}_p$ respectively. In the Fourier domain, assume that the GFT of signal $\bar{s}$ is $\bar{b}$. We denote the GFT matrix as F. If $\bar{b} = \begin{bmatrix} \bar{b}_L \\ \bar{b}_H \end{bmatrix}$, then the following equations describe the spectral relations between downsampled signals and submatrices of matrix F.

$$F\bar{s} = \bar{b} \Leftrightarrow \frac{1}{\sqrt{2}} \begin{bmatrix} U^* & V^* \\ U^* & -V^* \end{bmatrix} \begin{bmatrix} \bar{s}_k \\ \bar{s}_p \end{bmatrix} = \begin{bmatrix} \bar{b}_L \\ \bar{b}_H \end{bmatrix} \tag{2.3}$$

$$\Rightarrow U^*\bar{s}_k + V^*\bar{s}_p = \sqrt{2}\bar{b}_L \tag{2.4}$$

$$U^*\bar{s}_k - V^*\bar{s}_p = \sqrt{2}\bar{b}_H \tag{2.5}$$

$$\Rightarrow \sqrt{2}U^*\bar{s}_k = \bar{b}_L + \bar{b}_H \tag{2.6}$$

The Graph Fourier Transform for the signal $\bar{s}_k$ is denoted by F_k and it is characterized by relation $F_k\bar{s}_k = \bar{b}_L$. From Equation 2.3, we obtain $F_k = \sqrt{2}U^*$ for bandlimited signals. Let us now look at the effect of this Graph Fourier Transform on spectrum of $\bar{s}_k$, in order to examine the aliasing behavior on bipartite graphs.

The above equation suggests that the GFT of $\bar{s}_k$ is given by $\bar{b}_L + \bar{b}_H$, which

shows the aliasing effect in the frequency domain. Here, the frequencies related to eigenvalues σ and $-\sigma$ for the original graph get merged and the respective Fourier coefficients are added together to obtain the GFT on downsampled graph. From above equations, we can make following two conclusions. 1) If $\bar{b}_H = 0$, then there is no aliasing. 2) If $\bar{b}_H = \bar{b}_L$, then using Equation (2.3) and (2.6), we get $\bar{s}_p = 0$. This is similar to classical signal processing case where zero-insertion results in replication of spectrum.

In classical signal processing, the signal reconstruction from downsampled vertices to original set of vertices is obtained by zero-insertion followed by low-pass filtering. In case of bipartite graph, zero-insertion involves setting $\bar{s}_p = 0$ and then low pass filtering the signal $\bar{s'} = \begin{bmatrix} \bar{s}_k \\ 0 \end{bmatrix}$ with an ideal low pass filter. It can be shown that the filtered signal is given by $\begin{bmatrix} \bar{s}_k \\ VU^* \bar{s}_k \end{bmatrix}$, which is desired reconstruction. Thus, the reconstruction procedure for bipartite graph is identical to that for the classical signal processing.[3]

2.2 Bipartite Graphs: Algebraic Model for Downsampled Graph

In this section, we look at the algebraic model of the downsampling on bipartite graphs. As we know, the algebraic model for graphs can be described using the adjacency matrix. So, we need to provide a way to determine the adjacency matrix for downsampled graph. Let the bipartite graph $G = (\mathcal{V}_k, \mathcal{V}_p, A)$ be downsampled to a graph $G_k = (\mathcal{V}_k, A_k)$. Here $\mathcal{V}_k$ is the set of vertices which are preserved in order to form the new graph with half the number of vertices. We define $A_k := BB^T$. This axiomatic choice of A_k is justified by the results provided in Theorems 2.1-2.4.

In the algebraic model, the graph adjacency matrix is designated as the shift

[3] A similar analysis of spectral folding phenomena can be found in [22], where the normalized graph Laplacian is used in order to derive the spectral properties. The normalized Laplacian has eigenvalues within interval $[0, 2]$. After downsampling, a frequency $f \in [1, 2]$ folds onto $(2 - f) \in [0, 1]$.

operator. Therefore, A_k is the new shift operator and also the generator of the algebra of filters. The signal space for G_k is the set of vectors on ordered vertices $\mathcal{V}_k$. We discuss the Graph Fourier Transform, filtering and the homomorphism that exist between the algebra of filters on graphs G and G_k.

Theorem 2.1 *Consider a bipartite graph $G = (\mathcal{V}_k, \mathcal{V}_p, A)$ where $A = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix}$. Let $G_k = (\mathcal{V}_k, A_k)$ be the downsampled graph where $A_k = BB^T$. Let the Singular Value Decomposition of B be given by $U\Sigma V^*$. Then,*

1. *The GFT for graph G_k is $\sqrt{2}U^*$*
2. *If P is the selection matrix that maps a signal on graph G to a signal on graph G_k (i.e. $P\bar{s} = \bar{s}_k$), then $A_k = PA^2P^T$*

Proof 1) As $A_k = BB^T$, we have $A_k = U\Sigma^2U^*$ as diagonalization of A_k. This shows that $\sqrt{2}U^*$ is the GFT for a graph which has A_k as adjacency matrix. The scalar of $\sqrt{2}$ is included for consistency.
2) Given that $P\bar{s} = \bar{s}_k$, also $A^2 = \begin{bmatrix} BB^T & 0 \\ 0 & B^TB \end{bmatrix}$. It can be seen that $PA^2P^T = BB^T$. Hence $A_k = PA^2P^T$. ∎

Theorem 2.2 *Continuing notations from Theorem 2.1, if the set of filters is restricted to filters with only even powers of A, then filtering operation $h(A^2)\bar{s}$ on bipartite graph is homomorphic to filtering operation $h(A_k)\bar{s}_k$ on the downsampled graph.*

Proof As linearity of filters is given, proving $PA^2\bar{s} = A_kP\bar{s} = A_k\bar{s}_k$ is sufficient to show that $h(A_k)P\bar{s} = Ph(A^2)\bar{s}$. Here, $PA^2\bar{s} = P\begin{bmatrix} BB^T & 0 \\ 0 & B^TB \end{bmatrix}\begin{bmatrix} \bar{s}_k \\ \bar{s}_p \end{bmatrix} = P\begin{bmatrix} BB^T\bar{s}_k \\ B^TB\bar{s}_p \end{bmatrix} = BB^T\bar{s}_k = A_kP\bar{s}$. This homomorphism property is also explained via a commutative diagram in Figure 2.1. ∎

This property suggests that every LSI filter with even powers on original graph can be mapped to downsampled graph, thus giving us mapping $A^2 \mapsto A_k$. This

$$\begin{array}{ccc} (h(A^2), \bar{s}) & \xrightarrow{filtering} & h(A^2)\bar{s} \\ {\scriptstyle P(downsample)}\downarrow & & \downarrow{\scriptstyle P(downsample)} \\ (h(A_k), \bar{s}_k) & \xrightarrow{filtering} & h(A_k)\bar{s}_k \end{array}$$

Figure 2.1: The commutative diagram explaining the homomorphism that exist between the bipartite graph and the downsampled graph. Note that $\bar{s} \in \mathbb{C}^N$, P denotes downsampling operator. $h(A_k)P\bar{s} = Ph(A^2)\bar{s}$

is analogous to classical DSP, where $z^{-2} \mapsto z^{-1}$ in the $z - transform$ filter representation, while downsampling by a factor of two.

Now, let us find out the structure of polynomial basis which act as weights on each vertex. The details of findings are provided as Theorems 2.3 and 2.4.

Theorem 2.3 *Given a bipartite graph $(\mathcal{V}_k, \mathcal{V}_p, A)$, there exists a choice of polynomial basis corresponding to each vertex such that the polynomials for one bipartite partition are all even, while the same for the other bipartite partition are all odd.*

Proof In the algebraic model of graphs, the graph signal is mapped to a polynomial by means of polynomial basis $(b_0(x), \cdots, b_{N-1}(x))$ such that $s(x) = \Sigma_{i=0}^{N-1} s_i b_i(x) = b(x)^T \bar{s}$, where $b(x) = [b_0(x), \cdots, b_{N-1}(x)]^T$. Each polynomial $b_i(x)$ can be seen as a weight on the corresponding vertex. Let a basis polynomial be expressed as $b_i(x) = \Sigma_{j=0}^{N-1} b_{ij} x^j$. Construct the basis coefficient matrix $B_c = [b_{ij}]$ and matrix $\Lambda = [\lambda_j^{i-1}]$, where $0 \leq i, j \leq N-1$. From [32], these matrices are related by following equation

$$B_c \Lambda = \begin{bmatrix} U & U \\ V & -V \end{bmatrix}$$

Under the assumption that matrix Λ is invertible, the coefficients b_{ij} for $0 \leq i \leq N/2 - 1$ and even values of j, are all zeroes. Similarly, the coefficients b_{ij} for $(N/2) \leq i \leq N-1$ and odd values of j, are all zeros. Hence, the basis corresponding to one bipartite partition are all even polynomials while the same for the other bipartite partition are all odd polynomials.

In the case where there are repeated eigenvalues, the system of equations is underdetermined, with at least one solution where the basis can be chosen to have the even and odd polynomial property for separate bipartite partitions. ∎

Theorem 2.4 *In the algebraic model, the polynomial corresponding to downsampled signal can be obtained using even powered coefficients of the polynomial representing the original signal.*

Proof In the standard polynomial basis, the polynomial $s(x)$ is given by $a_0x^0 + \cdots + a_{N-1}x^{N-1}$. Let $\hat{s}$ be Fourier Transform for graph signal $\bar{s}$. Then,

$$\begin{bmatrix} \bar{s}_k \\ \bar{s}_p \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} U & U \\ V & -V \end{bmatrix} \hat{s}, \quad \textit{where } \hat{s} = \Lambda^T \begin{bmatrix} a_0 \\ \vdots \\ a_{N-1} \end{bmatrix}$$

As already noted, the eigenvalues of A have following relation. $\lambda_i = -\lambda_{i+N/2}, i = 0, \cdots, N/2-1$. Let $\Lambda_1 = [\lambda_j^{2i}], \Lambda_2 = [\lambda_{j+N/2}^{2i+1}], 0 \leq i, j \leq (N/2)-1$ and $\bar{a}_e = [a_0, a_2, \cdots, a_{N-2}]^T, \bar{a}_o = [a_1, a_3, \cdots, a_{N-1}]^T$. Using these notations, it can be shown that

$$\bar{s}_k = \sqrt{2}U\Lambda_1^T\bar{a}_e \tag{2.7}$$

On the other hand, the GFT for downsampled graph is given by $\sqrt{2}U^*$. If the signal on downsampled graph $\bar{s}_k$ is mapped to polynomial $s_k(x) = b_0x^0 + \cdots + b_{N/2-1}x^{N/2-1}$, then

$$\bar{s}_k = \frac{1}{\sqrt{2}}U\Lambda_1^T\bar{b} \tag{2.8}$$

where $\bar{b} = \begin{bmatrix} b_0 & \cdots & b_{N/2-1} \end{bmatrix}^T$. From (2.7) and (2.8), we can conclude that $\bar{b} = 2\bar{a}_e$. The scalar-multiplier of 2 can be removed by appropriate scaling of GFT of original as well as downsampled graph. ∎

Figure 2.2 gives an example of how polynomials are either even or odd depending upon bipartite partition.

In this chapter, we explored the intertwined nature of downsampling process and bipartite graphs. We analyzed the algebra, spectral and aliasing behavior and the module structure in form of even-odd polynomial assignments depending upon bipartite partition. In the next chapter, we focus on downsampling of arbitrary graphs.

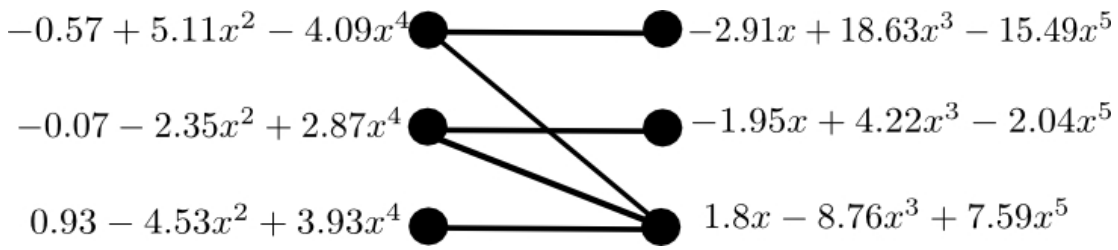

Figure 2.2: Example of polynomial basis for a bipartite graph, notice the even/odd nature of polynomials.

CHAPTER 3

Graph Downsampling by Minimizing the Aliasing Error

In Chapter 2, we proved the optimality of downsampling bipartite graphs by selecting a bipartite partition. However, not all graphs exhibit bipartite structure. In this chapter, we briefly discuss existing approaches to downsample arbitrary graphs and propose an approach based on GFT to accomplish the same. Property of bandlimitedness is important with respect to downsampling of a signal. It can be defined in terms of bandwidth in context of graph Laplacian based GFT, as used in [22]. However, if graphs are not undirected, such definition is restrictive due to the presence of complex eigenvalues. Thus, we use alternative approach of defining bandlimited signal where the number of spectral coefficients is used as a measure of bandwidth. The concepts of bandlimitedness and sample-set are useful in understanding the downsampling of arbitrary graphs, which are defined as follows.

Definition For a signal $\bar{s}$ on a given graph with GFT $\bar{b}$, if $\bar{b}(i) = 0, \forall i \geq n_0, n_0 \in \mathbb{N}$, then the signal is called *bandlimited* with bandwidth n_0.

Definition For a given graph, the set of vertices from which a given bandlimited signal can be reproduced uniquely without any error is called a *sample-set*.

To downsample arbitrary graphs, a method proposed in [22] locally approximates the bipartite structure using a graph-coloring technique. On the other hand, Nguyen and Do [24] rely on Maximum Spanning Tree(MST) of a graph in order

to downsample[1] the same. Both the approaches rely on approximating the given graph by a bipartite graph. These approaches do not consider the spectral aspect of the graphs directly, as they rely on topology of the given graph in order to obtain a sample-set. A major limitation with topological approaches is that although the signal is assumed to be bandlimited in spectral domain, the actual process of finding the downsampling scheme does not take into account spectral properties of the graph in a direct way. Moreover, these approaches can not be applied to downsample a directed graph, or a graph with negative edge-weights.

Spectral approaches to graph downsampling rely on GFT properties in order to obtain the downsampled sample-set. In [4], the eigenvector corresponding to the highest frequency is used to obtain a downsampling scheme. Based on polarity of eigenvector values, two equivalent sets of downsampled vertices are obtained. Another method to determine the sample-set of an undirected graph is provided by Anis *et al.* [2], in which a greedy approach is used to add a vertex in every iteration to the sample-set, which provides the highest increase in bandwidth, until the cut-off threshold is reached. Due to the greedy nature of the algorithm, the sample-set so obtained is not necessarily *optimal*[2]. As an example, every alternate sample is not necessarily selected during downsampling of a standard 1-D uniform grid. It should be emphasized here that the sample-set (of a given cardinality) for a given bandwidth is not unique, and the algorithm indeed converges to one of those sample-sets. However, different sample-sets have different sensitivity to aliasing in case of signals which are not bandlimited, which indicates that even among sample-sets, the quality of signal-reconstruction differs. The objective in [2] is to find the least number of samples (and corresponding sample-set) for a signal with given bandwidth. On the other hand, our purpose is to provide a way to select the *best possible* $N/2$ vertices for a graph with N vertices.

Another issue in graph downsampling is measuring the quality of the affected partition on graph. Cut-index [24] is one of the objective measure used to determine quality of the graph downsampling scheme (defined in Chapter 2). A downsampling scheme with higher cut-index is considered to have better quality

[1]It should be noted here that every tree is a bipartite graph.
[2]The meaning of optimality will be provided in later sections

(and hence considered to have better signal reconstruction properties). One major issue with cut-index is that a single cut provides us two downsampling options (i.e. both partitions are considered equally good), selecting one of the two is an arbitrary choice. Another objective measure used to determine quality of graph downsampling scheme is stability of the reconstruction frame [8] (also defined in Chapter 2). The higher the stability of the frame, the better the quality of downsampling scheme.

For a graph $G = (\mathcal{V}, A)$ with N nodes, let the GFT matrix be denoted by F, where $F \in \mathbb{C}^{N \times N}$. N is assumed to be even as we focus on downsampling the graph by two. If a signal on this graph is bandlimited with all the energy contained in the lower half of the frequency spectrum, then the GFT of the signal is of form $[b_1, b_2, ..., b_{N/2}, 0, ...0]^T$. The spectrum can be expressed as $[\bar{b}_L^T, \bar{b}_H^T]^T$, where $\bar{b}_L = [b_1, b_2, ..., b_{N/2}]^T$ and $\bar{b}_H = [0, ..., 0]^T$. Let $\mathcal{V}_p$ be the set of nodes to be purged and $\mathcal{V}_k$ be the set of nodes to be kept, both containing $N/2$ nodes. For a given graph signal $\bar{s}$, let $\bar{s}_k$ and $\bar{s}_p$ be the signal values taken from nodes in the sets $\mathcal{V}_k$ and $\mathcal{V}_p$ respectively. As both the sets are selections from $\mathcal{V}$, we can write, $P_p\bar{s} = \bar{s}_p, P_k\bar{s} = \bar{s}_k$ where P_p and P_k are selection matrices. If we fix the order of nodes in $\mathcal{V}_k$ and in $\mathcal{V}_p$, then P_p and P_k are unique. We can also write,

$$P\bar{s} = \begin{bmatrix} \bar{s}_k \\ \bar{s}_p \end{bmatrix} \tag{3.1}$$

where P is an invertible permutation matrix, with inverse P^T.

Similarly, we can also define selection matrices P_L and P_H such that $P_L\bar{b} = \bar{b}_L, P_H\bar{b} = \bar{b}_H$.

Since $F\bar{s} = \bar{b}$, $FP^TP\bar{s} = \bar{b}$.

$$\therefore F_P \begin{bmatrix} \bar{s}_k \\ \bar{s}_p \end{bmatrix} = \begin{bmatrix} \bar{b}_L \\ \bar{b}_H \end{bmatrix} \tag{3.2}$$

where $F_P = FP^T$. If we write F_P as

$$\begin{bmatrix} F_1 & F_2 \\ F_3 & F_4 \end{bmatrix},$$

then we get

$$\begin{bmatrix} F_1 & F_2 \\ F_3 & F_4 \end{bmatrix} \begin{bmatrix} \bar{s}_k \\ \bar{s}_p \end{bmatrix} = \begin{bmatrix} \bar{b}_L \\ \bar{b}_H \end{bmatrix} \tag{3.3}$$

where F_1, F_2, F_3 and F_4 are $\frac{N}{2} \times \frac{N}{2}$ matrices. Given $\bar{b}_H = 0$, $\bar{s}_p$ can be uniquely determined from $\bar{s}_k$ if and only if the submatrix F_4 is invertible. Note that $F_4 = P_H F P_p^T$. Moreover, using Schur Complement [44] on the above equation, we obtain a matrix F_{kL} such that

$$F_{kL}\bar{s}_k = \bar{b}_L \Leftrightarrow F_{kL} = F_1 - F_2 F_4^{-1} F_3 \tag{3.4}$$

The matrix F_{kL} can be understood as the GFT on the downsampled graph. The signal on purged nodes, denoted as $\bar{s}_p$ can be recovered from $\bar{s}_k$, using the following reconstruction rule obtained from Equation (3.3):

$$\bar{s}_p = -F_4^{-1} F_3 \bar{s}_k. \tag{3.5}$$

Thus, the procedure described above, allows us to find a condition for perfect reconstruction and at the same time, provides us with the GFT on the downsampled graph. There can be multiple sample-sets of same cardinality for a given graph. A similar analysis for condition for perfect reconstruction of bandlimited signals is provided in [6].

3.1 Minimization of Reconstruction Error for Nonbandlimited Signals

The discussion so far indicates that if the matrix $F_4 = P_H F P_p^T$ is invertible, then any bandlimited signal can be reconstructed without any error from nodes con-

tained in $\mathcal{V}_k$. This raises a question: *Are all possible node-selections with corresponding invertible F_4, equivalent?* As far as bandlimited signals are concerned, all sample-sets are equivalent. However, the property of bandlimitedness is highly restrictive. In the analysis till now, we have assumed a perfectly bandlimited signal, i.e., $\|\bar{b}_H\| = 0$. However, in real-world scenarios, we often encounter situations where $0 < \|\bar{b}_H\| = \epsilon << \|\bar{b}_L\|$. We refer to such signals as *lowpass* signals. In this section, we will analyze this scenario which will help in obtaining an optimal downsampling scheme from the signal reconstruction point of view. From Equation (3.3)

$$F_3\bar{s}_k + F_4\bar{s}_p = \bar{b}_H \Rightarrow \bar{s}_p = -F_4^{-1}F_3\bar{s}_k + F_4^{-1}\bar{b}_H \tag{3.6}$$

The reconstruction error e_r is

$$e_r = F_4^{-1}\bar{b}_H \Rightarrow \|e_r\| \leq \frac{\epsilon}{\sigma_{min}(F_4)} \tag{3.7}$$

Here, $\sigma_{min}(F_4)$ denotes the minimum singular value of F_4 and characterizes the sensitivity of the reconstruction error (from signal values on $\mathcal{V}_k$) to high frequency content. For a given partition $(\mathcal{V}_p, \mathcal{V}_k)$ of a given graph, $\sigma_{min}(F_4)$ is referred to as the *SVD based Downsampling Quality Measure* (abbreviated as SDQM), whereas the matrix F_4 is referred to as the *descriptor submatrix* of the sample-set. It should be observed here that if $SDQM = 0$, then F_4 is not invertible and the signal cannot be reconstructed. Maximizing SDQM reduces the upper-bound on error. As far as bandlimited signals are concerned, all downsampling schemes with $SDQM \neq 0$ are equivalent. However, when the signal is not bandlimited, they exhibit different amount of sensitivity towards the high frequency content of the signal. Thus, the goal of downsampling should be to find a sample-set that maximizes $SDQM$.

With this analysis, the problem of downsampling can be stated as the following optimization problem,

$$P_{opt} = \underset{P_p \in \{0,1\}^{N/2 \times N}}{\operatorname{argmax}} \{\sigma_{min}(P_H F P_p^T)\} \tag{3.8}$$

Set of selected nodes	$SDQM$
$\{1,3,5\},\{2,4,6\}$	0.7071
$\{1,2,3\},\{2,3,4\},\{3,4,5\},...,\{6,1,2\}$	0.1691
Rest of the combinations (12 in total)	0.3568

Table 3.1: SDQM for all possible downsampling schemes for graph in Figure 3.1. The *consecutive selection* (e.g. {1,2,3}) shows least $SDQM$, while *every alternate node selection* (e.g. {1,3,5}) has the largest $SDQM$.

In the above optimization, P_H is known (selection of high frequency components), F is the GFT of graph G and P_p is to be found, which provides the selection of the nodes to be purged. As we regard $SDQM$ as a quality measure for a given downsampling scheme, we explain the effect of this measure by an example on uniform 1-D grid (also called DFT grid [29]). Figure 3.1 shows the well-known downsampling on the grid and the resultant smaller grid for $N = 6$. The optimal solutions based on $SDQM$ criteria are $\{1,3,5\}$ and $\{2,4,6\}$. Table 3.1 shows various selected nodes combinations and corresponding $SDQM$.

The method presented in [8] attempts to obtain a stable reconstruction frame in order to downsample the signal. This approach works on the inverse GFT matrix, selects the columns (i.e. eigenvectors of adjacency matrix) which are to be preserved and them selects the vertices (i.e. rows) such that the reconstruction frame is stable. This approach can be expressed as following optimization problem using the notations used in this document.

$$P_{opt} = \underset{P_k \in \{0,1\}^{N/2 \times N}}{\operatorname{argmax}} \{\sigma_{min}(P_k F^{-1} P_L^T)\}$$

It should be noted that our formulation provides matrix P_p, which gives the set of nodes to be purged, while the matrix P_k in the above formulation provides the set of nodes to be kept. The formulations emphasize the differences in both the approaches. The difference are more noticeable in downsampling of directed graphs as shown in Section 3.4.3. It can be shown that the two formulations are identical when the GFT matrix F is orthogonal. More discussion on the difference between the two approaches is provided in Section 3.6.

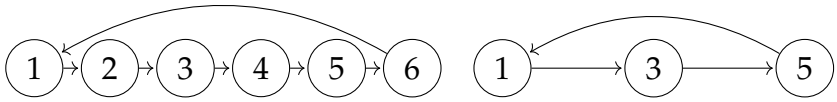

Figure 3.1: (left) A six-node directed circulant graph, (right) Corresponding downsampled graph.

3.2 A Greedy Algorithm for Downsampling Based on SDQM

In the relation $F_4 = P_H F P_p^T$, the matrix P_H is $N/2 \times N$ rectangular matrix. Hence, $P_H F$ is also an $N/2 \times N$ rectangular matrix. Let $F_H = P_H F$, then the desired optimization turns into a column selection problem from F_H such that the resultant matrix has maximum smallest singular value. A similar problem is discussed in [39], where the parameter to minimize is condition number of the selected columns from F_H. The number of combinations to select the columns in matrix F_H are $\binom{N}{N/2}$. An exhaustive search would require computing minimum singular values $\binom{N}{N/2}$ times, which is computationally impractical for large values of N.

To the best of our knowledge, there is no known algorithm to solve the given problem in polynomial time complexity. So, we propose a greedy strategy that may yield a suboptimal solution. The proposed greedy algorithm is summarized in Algorithm 1. Let F_4^i denote an $N/2 \times i$ matrix obtained by selecting i columns from F_H. Given F_4^i, F_4^{i+1} is obtained by augmenting F_4^i with a column from F_H that maximizes the smallest singular value F_4^{i+1}. Iterations continue till $i = N/2$. The indices of the columns selected from F_H forms the set $\mathcal{V}_p$, the set of vertices to be purged.

As discussed earlier, the GFT for bipartite graph has two sets of equal-norm orthogonal columns in the upper-half of frequency spectrum. This allows a single orthogonal column to be selected at every iteration of greedy algorithm, eventually converging to the optimal solution for any bipartite graph with equal number of nodes in each partition.

Algorithm 1: Greedy algorithm for maximizing minimum singular value of the selected columns

Input : F_H, N
Output: $\mathcal{V}_k$
Procedure: DownSample
$i \leftarrow 1$
$\mathcal{V}_k \leftarrow \{1, ..., N\}$
$\mathcal{V}_p \leftarrow \{\}$
while $i \leq N/2$ **do**
  $N_d = getNodeToDelete(F_H, \mathcal{V}_k, \mathcal{V}_p)$
  $\mathcal{V}_p \leftarrow \mathcal{V}_p \cup \{N_d\}$
  $\mathcal{V}_k \leftarrow \mathcal{V}_k - \{N_d\}$
  $i \leftarrow i + 1$
return $\mathcal{V}_k$

Input : $F_H, \mathcal{V}_k, \mathcal{V}_p$
Output: $index$
Procedure: getNodeToDelete
Array $minSVD$
forall $i \in \mathcal{V}_k$ **do**
  $F_{iter} \leftarrow columns\ from\ F_H\ given\ by\ \mathcal{V}_p \cup \{i\}$
  $minSVD(i) \leftarrow \sigma_{min}(F_{iter})$
$index = \text{argmax}_i\{minSVD\}$
return $index$

3.3 Analysis of Standard Graphs in Context of SDQM

In Chapter 2, we analyzed downsampling on bipartite graphs in context of various downsampling approaches. In this section, we analyze graphs related to DFT and DCT in context of SDQM. We verify that the SDQM based approach provides the every alternate vertex selection as a downsampling scheme for DFT graphs. However, the same is not true for DCT graphs, which we analyze in detail.

3.3.1 SDQM for DFT Graphs

In this subsection, we use the SDQM based approach to analyze the downsampling of DFT graphs. If W is the N-point DFT matrix, then it is given by,

$$W = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix},$$

where $\omega = e^{-2\pi i/N}$. The matrix W can also be written as $W = \frac{1}{\sqrt{N}}(\omega^{jk})_{j,k=0,\cdots N-1}$. As we focus on downsampling by a factor of two, we assume that N is even. Here, there are two possibilities, N is divisible by 4 or $N+2$ is divisible by 4. For maintaining simplicity, we analyze the case where N is divisible by 4. The other case can be analyzed using a similar approach.

For downsampling by selecting every alternate vertex, (ignoring the scalar multiplier) the descriptor submatrix is given by $F_4 = (\omega^{jk})_{j,k}$, where $j = N/4, N/4+1, \cdots, 3N/4-1$ and $k = 1, 3, \cdots N-1$. It can be shown that columns of matrix F_4 are orthogonal to each other. Thus, the SDQM for the corresponding sample-set is optimal.

3.3.2 SDQM for DCT Graphs

In this subsection, we analyze the Discrete Cosine Transform (DCT-type II) and its graph, and use the proposed graph downsampling technique to downsample the same. A detailed study on the graphs for which the DCT is GFT is provided in [29]. The graph for DCT-type II transform is given in Figure 3.2. DCT-type II transform diagonalizes the adjacency matrix of this graph. The graph for DCT is undirected, and has self-loops at the end-nodes. Looking at the structure of graph, intuitively, selecting every alternate node is a good strategy for obtaining the downsampled scheme. However, using the $SDQM$ measure, we find that

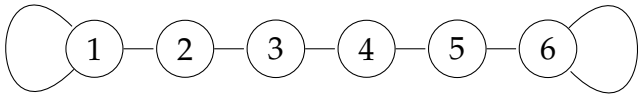

Figure 3.2: Graph For DCT-type II ($|\mathcal{V}| = 6$). Notice the self-loops for end-nodes.

there exists a better quality downsampling scheme.

As mentioned earlier, one can visualize the DCT as a GFT by assigning an appropriate adjacency matrix to the graph. Using details provided in [27], we reproduce the graph related to DCT-type II. We call this form of graph as the *DCT-graph* in short. Figure 3.2 provides a DCT-graph for 6-vertices. The corresponding adjacency matrix is given by,

$$A = \begin{bmatrix} 1 & 1 & & & & \\ 1 & & 1 & & & \\ & 1 & & 1 & & \\ & & 1 & & 1 & \\ & & & 1 & & 1 \\ & & & & 1 & 1 \end{bmatrix} \tag{3.9}$$

Now, we will apply four different approaches of graph downsampling on the DCT graph. The four approaches include bipartite approximation, MST approximation, highest frequency eigenvector based approach and SDQM based approach. The results are provided in form of theorems.

Theorem 3.1 *Given a DCT-graph with even number of nodes to be downsampled by a factor of two, the approach using MST approximation provides every alternate sample selection as the downsampled vertex-set.*

Proof In the MST based approach, we first extract an MST from the given graph. For DCT-graph, the only possible MST is when we remove the two self-loops at end nodes. After extracting MST, we divide the nodes into two sets using even-odd distance from an arbitrary root node. For DCT-graphs, this results in the sample-set obtained by selecting every alternate vertex.

Theorem 3.2 *Given a DCT-graph with even number of nodes to be downsampled by a factor of two, the approach using bipartite approximation provides every alternate sample selection as the downsampled vertex-set.*

Proof In the bipartite approximation approach, we first approximate a given graph by a bipartite graph by removing some edges. In DCT-graphs, the optimal way (i.e. by removing least number of edges) to obtain a bipartite graph is to remove the self-loops at the end nodes. After this, there is unique way in which we can obtain the bipartite partitions. This is identical to the sample-set obtained using every alternate vertex selection.

With respect to Theorems 3.1 and 3.2, it must be reminded that every tree is a bipartite graph. In case of DCT-graphs, the resultant bipartite graph is also a tree. Hence, the procedures yield the same results.

Theorem 3.3 *Given a DCT-graph with even number of nodes, the approach using highest frequency eigenvector provides every alternate sample selection as the downsampled vertex-set.*

Proof In this approach, the polarity of highest frequency eigenvector is used as the feature to determine the downsampled vertex-set. To prove this result, we first extract the highest frequency eigenvector and then show that it has alternate polarity for every alternate sample value. Let N be the number of vertices in the graph (i.e. length of signal). Note that N is even. If signal $\bar{s}$ has DCT $\bar{S}$, then

$$\bar{S}[k] = \sum_{n=0}^{N-1} \bar{s}[n] cos(\pi k(n + 0.5)/N),$$

where $k = 0, \cdots, N-1$. Here, index k corresponds to frequency and index n corresponds to vertex. For highest frequency, substitute $k = N-1$. Let the highest frequency eigenvector be denoted by $\bar{e}_H$, then

$$\bar{e}_H[n] = cos(\pi(N-1)(n + 0.5)/N)$$

Set of selected nodes	$SDQM$
$\{1,4,6\},\{1,3,6\}$	0.5978
$\{1,3,5\},\{2,4,6\}$	0.5120
$\{1,4,5\},\{2,3,6\}$	0.3827
$\{2,3,5\},\{2,4,5\}$	0.3601
Rest of the combinations	< 0.3

Table 3.2: SDQM for all possible downsampling schemes for graph in Figure 3.2.

where $n = 0, \cdots, N-1$. Simplifying above equation,

$$\bar{e}_H[n] = -sin(\pi\alpha)$$

where $\alpha = (n - n/N - 0.5/N)$.

Now, if n is even, then $n = 2m$ and $m = 0, \cdots, (N-2)/2$. Thus, for even n, α can be written as

$$\alpha = 2m - \frac{4m+1}{2N}$$

For $m < (N-1)/2$, it can be shown that $0 < \frac{4m+1}{2N} < 1$. Thus,

$$2m - 1 < \alpha < 2m$$

Hence, we can conclude that $\bar{e}_H[n] > 0$ if n is even. Using a similar approach, we can also show that $\bar{e}_H[n] < 0$ if n is odd. Thus, we can conclude that every alternate sample in the highest frequency eigenvector has alternate polarity. This proves the result.

Theorem 3.4 *Given a DCT-graph with even number of nodes, selecting every alternate vertex does not provide optimal result based on SDQM.*

Proof We prove this result by providing a counter-example. We consider a six-node DCT-graph and list various possible SDQM for different sample-sets. The SDQMs are listed in Table 3.2. It is clear from the table that SDQM from every alternate vertex-set is not the optimal case.

The reason for this counterintuitive result in the SDQM based approach is the

boundary conditions for signal imposed upon the signals while computing the DCT. For DCT, the signal is reflected and extended before computing the transform. This results in different pattern of samples and thus selection of vertices is not the same as the topological one. A similar pattern is observed in 8, 16 and 32 vertex DCT-graphs. We experimentally validate the results in Subsection 3.4.4. The results presented in this subsection (i.e. the observations on downsampling of DCT-graphs) are accepted for publication in [41].

3.4 Experimental Validation

In this section, we apply Algorithm 1 to downsample undirected and directed graphs. The measure of quality of downsampling scheme is given by the reconstruction error, from the downsampled graph to the original graph. In Section 3.4.1, we observe the effect of presence of negative edges on various downsampling schemes. In Sections 3.4.2 and 3.4.3, we downsample undirected and directed graphs respectively, and compare the reconstruction errors with existing downsampling schemes. Downsampling of DCT-graphs is used in JPEG image compression standard for the chrominance components of an image.

3.4.1 Downsampling Random Graphs

In order to test the SDQM based method on a large set of arbitrary graphs, random graphs are chosen. Random graphs allow us to test our algorithm and compare its performance to other approaches on a large number of instances. To generate random graphs, we fix the number of vertices and then randomly draw edge-weights from a predefined probability distribution. As noted in Chapter 1, negative edge-weighs are required to model the cases where the relation between two vertices can be expressed by a negative relation (e.g. negative correlation). On the other hand, if the edge-weights are determined based on distance or probabilities, we may arrive at an adjacency matrix with all entries being non-negative. We conduct the experiment for graphs with non-negative edge-weights and for graphs which have negative as well as positive edge-weights.

	MST	Spectral	Proposed
Nonnegative Edge-weights			
SDQM:	0.0196	0.0178	**0.1449**
Cut Index:	0.5718	**0.6158**	0.5721
Negative And Positive Edge-weights			
SDQM:	0.0162	0.0119	**0.1555**
Cut Index:	0.5710	0.5047	**0.6037**

Table 3.3: SDQM and cut-index for random undirected graphs

In the experiment, we randomly generate graphs with 100 vertices ($|\mathcal{V}| = 100$). For non-negative weights, each entry of adjacency matrix is drawn from a uniform distribution $U(0, 1)$. For adjacency matrix with negative weights, each entry is drawn from Gaussian distribution $\mathcal{N}(0, 1)$. The adjacency matrix thus obtained is made sparse by sparsity ratio in range of $2\% - 30\%$[3]. 1000 instances of such matrices are generated for non-negative and negative-positive each. Table 3.3 summarizes average SDQM and cut-index measures for the trial using MST based approach, spectral approach and proposed approach (i.e. Algorithm 1).

One can observe from the table that presence of negative weights deteriorates performance of both MST based and spectral approach according to SDQM and cut-index measures. In case of spectral method, the difference (between graphs with negative edge weights and those without them) can be explained by the fact that the spectral method attempts to affect a max-cut on the given graph, hence negative edge-weights adversely affects the performance. On the other hand, the proposed approach, while optimizing SDQM also maintains cut-index comparable to MST based approach. One more remarkable feature is that the performance of proposed approach is unaffected by introduction of negative edge-weights. This experiment establishes that the proposed approach maximizes SDQM while maintaining a high cut-index and at the same time, it can also process graphs with negative edge-weights.

[3]Sparsity is selected arbitrarily to simulate sparse matrices

3.4.2 Downsampling Undirected Graphs

The data used in the experiment is temperature data from weather stations, publicly available on [1]. When processing such data, statistical correlation or distance based computations are often used to arrive at the edge-weights, which result in the resultant graph being undirected. We will test the efficacy of proposed algorithm for downsampling such undirected graph. From the database, we consider 196 nodes from which undirected graphs is constructed. Data for year 2014 is considered with data available on all nodes for 365 days. Thus, we have 365 graph signals with number of nodes being 196. We use similarity measure given by statistical correlation. The diagonal entries of the correlation-matrix are all set to 0, and the matrix is normalized with the largest eigenvalue. The matrix is then designated as the adjacency matrix of the graph. This matrix is symmetric, and hence represents an undirected graph. We diagonalize the adjacency matrix in order to obtain the GFT for the given graph.

We obtain the downsampled grids using MST based approach, spectral method and proposed method. For each downsampled grid, we reconstruct the graph signal on purged nodes using the values on kept nodes. The reconstruction accuracy is defined as $20 \log \left(\frac{\|\bar{s}\|}{\|e_r\|} \right)$, where $\| \cdot \|$ denotes 2-norm. Figures 3.3 and 3.4 provide the downsampled grids (both purged and kept nodes) and reconstruction accuracy. The reconstruction accuracies indicate that the proposed algorithm outperforms both the methods. The value of $SDQM$ for spectral downsampling approach and MST based approach are 0.003 and 0.004 respectively, while the same for proposed approach is 0.15. This fact reflects directly in the reconstruction errors.

It should be noted here that maximizing the cut-index amounts to selecting a set of vertices which are strongly related to the purged set of vertices. This is a first order operation, i.e., only the first order effects of vertices on each-other are used to determine strength of relations. Due to strong relation, this allows us to reconstruct a signal on purged set of vertices from the selected set of vertices. On the other hand, the SDQM based approach takes a more direct route of minimizing the reconstruction error and thus able to provide a significant improvement

over other approaches.

3.4.3 Downsampling Directed Graphs

For this experiment, we use the same dataset as used in Section 3.4.2. In some cases [7], to normalize the effect of every vertex, the edge-weights are normalized in a way that the sum of edge-weights coming into every vertex is identical. This process helps prevent the scaling of values for graph signals when we apply graph filter as a higher order polynomial. Such a process results in a directed graph. In order to create a directed graph for the temperature data, we first create an 8-neighborhood distance-based adjacency matrix $\tilde{A}$, whose (i, j) entry is $\tilde{a}_{i,j} = e^{-\frac{dist(i,j)^2}{d_0^2}}$. Here, d_0 is the mean distance over entire grid. Similarly, $dist(i, j)$ is geometric (Euclidean) distance between latitude and longitude of weather stations (nodes) numbered i and j. After this, each row of $\tilde{A}$ is normalized to have unit norm in order to obtain adjacency matrix A. This process makes the adjacency matrix asymmetric, hence the adjacency matrix based approach is used to obtain GFT for this graph. Using this GFT and Algorithm 1, we obtain a downsampling scheme on graph. For this downsampling, the reconstruction error for various levels of high-frequency content is shown in Figure 3.5. We compare the proposed approach to the stable frame approach provided in [8]. The value of $SDQM$ for the obtained partition is 0.13.

3.4.4 Validation of Downsampling of DCT-graphs

In this subsection, we provide two experiments (one using 1-D data and another using 2-D data) in order to validate the results provided in Subsection 3.3.2. For 1-D data, we use temperature data and for 2-D data, we use images.

Validation Of Downsampling Of DCT Graphs Using Temperature Data

In this experiment, we use the temperature data collected across 196 weather stations in US [1]. For a single weather station, data is collected as a vector. In order to obtain vectors of different lengths, we vary the number of days considered

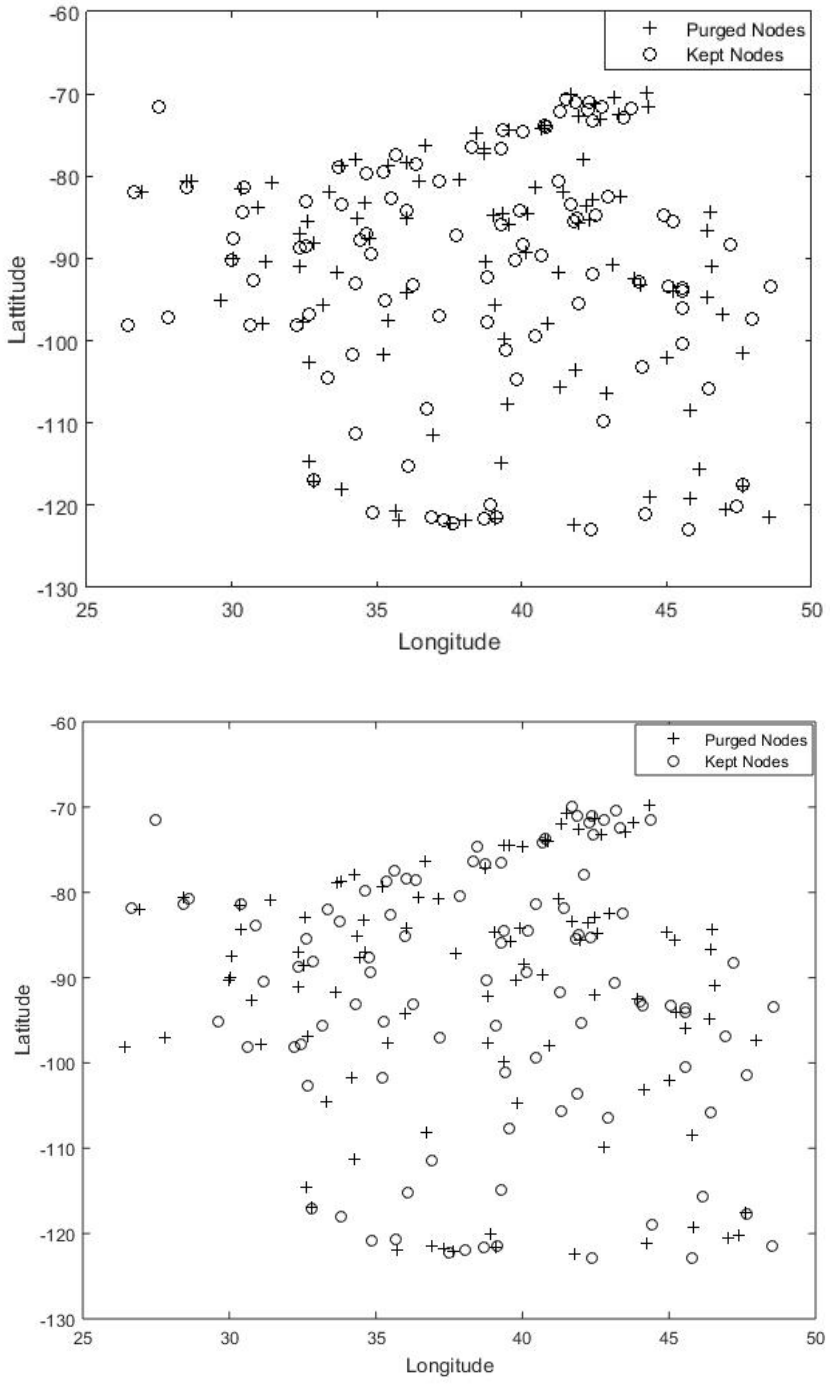

Figure 3.3: Result of downsampling undirected temperature data graph: (Top) MST Based approach (Bottom) SVD Based approach, + denotes purged nodes, ∘ denotes preserved nodes.

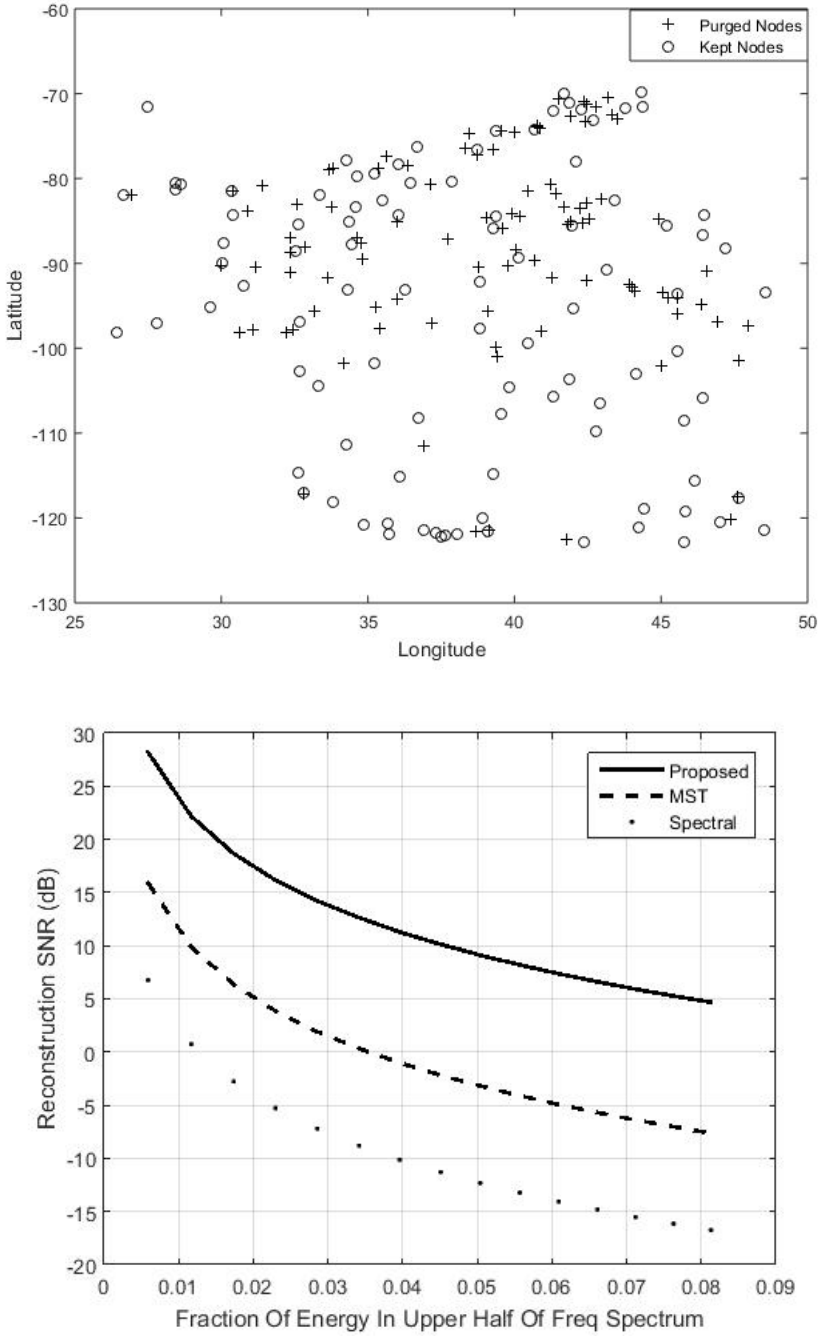

Figure 3.4: (Top) Result of downsampling undirected temperature data graph using proposed method, + denotes purged nodes, ∘ denotes preserved nodes (Bottom) Reconstruction accuracy vs High frequency content in signal

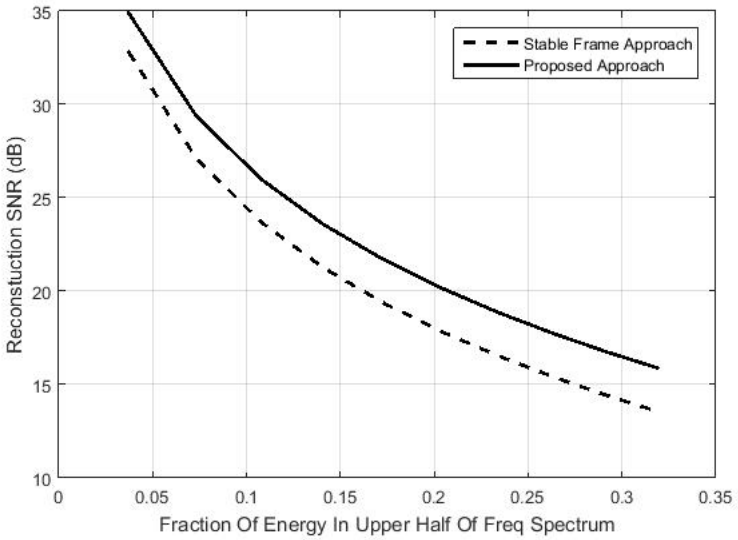

Figure 3.5: Reconstruction accuracy vs High Frequency content in signal (Directed Graph).

from 20 to 200. Thus we obtain 196 vectors of various lengths for testing the DCT downsampling.

To obtain the reconstruction errors for the every alternate vertex method, we downsampled the signal using the same and use $N/2$-point DCT followed by N-point inverse DCT (IDCT) reconstruction applied on zero-padded data. To obtain reconstruction errors for SDQM based downsampling, we downsampled the signal and use $N/2$-point derived GFT followed by N-point IDCT applied on zero-padded data. If the original signal is $\bar{s}$ and reconstructed signal is $\bar{s}_r$, then the percentage error is computed as $100 \times \frac{\|\bar{s}-\bar{s}_r\|}{\|\bar{s}\|}$.

Figure 3.6 shows the error curves for both the approaches. It is apparent from the figure that SDQM based approach performs better in reducing the reconstruction error.

Validation Of Downsampling Of DCT Graphs Using Image Data

Downsampling of DCT-graphs is used in the JPEG compression standard for color images. In the JPEG compression standard, a color image is first converted into YUV components (Y is luminance, and U, V are chrominance components). For efficient implementation, the image is first divided into blocks, usual size being $16 \times$

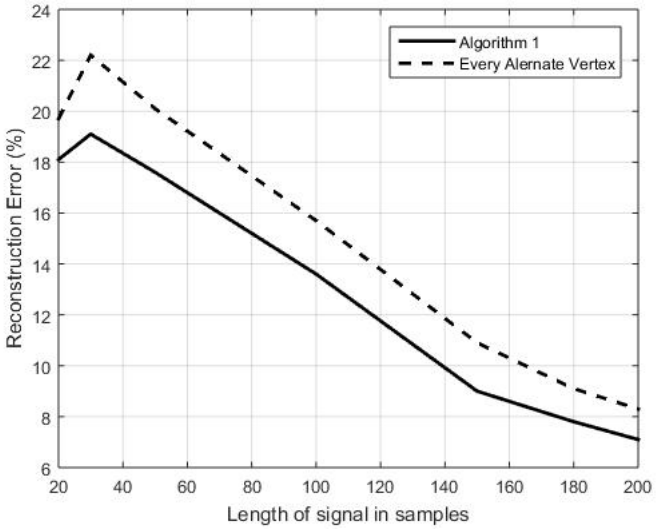

Figure 3.6: Reconstruction errors for different approaches of downsampling the DCT-graph

16. For $\mathcal{V} = 16$, Algorithm 1 converges to the sample-set: $\mathcal{V}_k = \{1, 3, 6, 8, 10, 12, 14, 16\}$. Let us denote the selection of every alternate sample as set $\mathcal{V}_r = \{1, 3, 5, 7, ..., 15\}$, which serves as reference for comparing the results. $SDQM$ for $\mathcal{V}_k$ is 0.4865 while for $\mathcal{V}_r$, it is 0.4323. According to our hypothesis, $\mathcal{V}_k$ should outperform $\mathcal{V}_r$ in signal reconstruction error.

As human eye is less sensitive to chrominance, every 16×16 (non-overlapping) block of U and V components, are first downsampled to 8×8 block and then 2-D DCT is applied on these blocks in order to compress the same[4]. In this experiment, we change the downsampling set from $\mathcal{V}_r$ to $\mathcal{V}_k$ and show how the sample-set $\mathcal{V}_k$ can reproduce original blocks with reduced error. For forward transform, 8-point DCT is used for $\mathcal{V}_r$ and F_{kL} (see Section 3.1) is used for $\mathcal{V}_k$. The reconstruction into 16×16 blocks is performed using 16-point 2-D IDCT on the transformed blocks with appended zeros in both the cases. We select three images namely *Lena*, *Barbara* and *Baboon* images (all of size 192×192), which are shown in Figure 3.7. The blockwise average percentage errors in U and V components are provided for all three images for both the sample-sets in Table 3.4. The error for a single block is computed using 2-norm of the error block, and then the error is averaged over all

[4]Note that a 16×16 pixel block forms a graph that is the cartesian product of the graph given in Figure 3.7 with itself. Hence GFT on the 16×16 node graph is the Kronecker product of GFT for graph in Figure 3.7, with itself. For details refer [33]

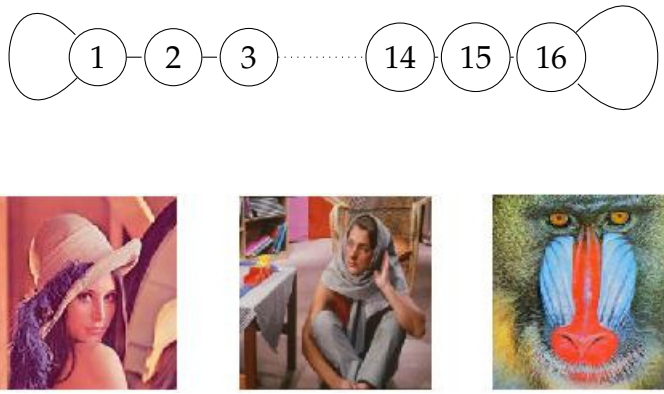

Figure 3.7: (Top) Graph For DCT-type II ($|\mathcal{V}| = 16$). (Bottom) Images (from left to right): Lena, Barbara, Baboon

blocks to obtain blockwise average error. It can be seen that the sample-set derived using the proposed algorithm reproduces the chromatic components with reduced error compared to standard DCT-IDCT method. The difference in SDQM explains the different performance of both schemes.

It should be emphasized here that the purpose of this experiment is not to provide a new method of image-compression. Rather, the purpose is to show how underlying graph structure provide non-intuitive downsampling schemes which are captured well by the proposed quality measure SDQM.

	Baboon		Barbara		Lena	
	$\mathcal{V}_r$	$\mathcal{V}_k$	$\mathcal{V}_r$	$\mathcal{V}_k$	$\mathcal{V}_r$	$\mathcal{V}_k$
U-Component	4.2241	**3.6605**	1.3841	**0.9817**	0.9806	**0.4970**
V-Component	3.6346	**3.0814**	1.2641	**0.8374**	0.6843	**0.4108**

Table 3.4: Blockwise Average Percentage Errors For Downsampled Images

3.4.5 Data Dependent Downsampling: An Example

So far, we have used graphs which are not derived from the signal directly. However, in some applications, the graph for a given signal is derived from the sample values (e.g. graphs used for images in [36]). In this experiment, we observe how the proposed downsampling affects the downsampling of data-dependent

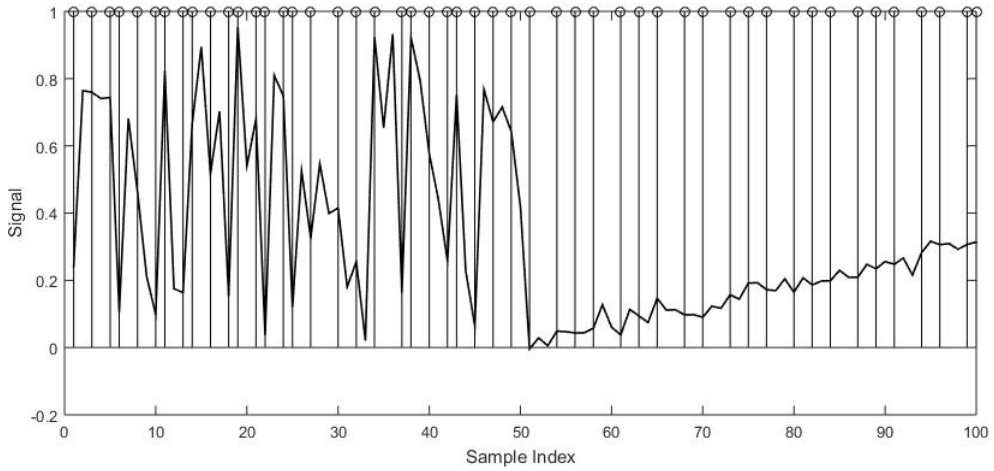

Figure 3.8: Variation in sampling in fast varying vs slow varying portions of signal

graphs. We generate a synthetic signal which has first half samples with high variations and second half with low variations. We then create a 4-neighborhood value based graph for the same signal. i.e., if node i and node j are within 4-neighborhood of eachother, then weight connecting the two nodes is $Ce^{-((s_i-s_j)^2)/\theta}$, where $\theta = 0.2$, while s_i, s_j are signal values on nodes i and j respectively. Value of C is 1 for nearest neighbor and 0.7 for second neighbor.

If we apply the bipartite graph based downsampling approach after finding the MST, then the resultant scheme tends to pick alternate samples from the 1-D grid, in the low variation part. This means, that the downsampling would purge same number of nodes from both the high variation part and the low variation part. On the other hand, the downsampling scheme obtained using the proposed method tends to have higher number of samples in regions with high variations. In Figure 3.8, the first half of the signal contains higher variations, and the downsampling scheme suggests that 60% samples in the downsampled signal would come from first half of the signal.

3.5 Adjacency Matrix for Downsampled Graph

While downsampling of a graph allows us to reduce the number of vertices and offers a simpler graph representation, the inter-relations between the vertices ob-

tained via downsampling process are unknown. The adjacency matrix for the downsampled vertices should also exhibit certain properties present in the original graph. If the original graph is sparse, the downsampled graph should also be sparse. If the original graph is undirected, then the downsampled graph should also be undirected. In this section, we review common approaches to obtain the adjacency matrix for the downsampled graph, discuss their shortcomings and propose a method to do the same from an algebraic perspective using the analogy from bipartite graphs.

Some approaches rely upon Kron-reduction which can be applied on the loopy-Laplacian [11] or the adjacency matrix [43]. The Kron Reduction is a relatively simple technique for eliminating nodes from a network when the signal value at that node is always zero, e.g. grounded nodes in an electrical network always have zero voltage at the node. As the assumption that the sample value on the vertex is always zero is not true for an arbitrary graph, the scope of this approach is limited. Another approach proposed in [8] relies on sampling the eigenvectors on the lower half of the spectrum on the nodes that are selected. It is an algebraic approach, and a major drawback of this approach is that it may not preserve the structure of the original graph in a topological sense (e.g. properties like sparsity and undirectedness).

We propose a simple approach to define the adjacency matrix of downsampled graph in case of downsampling by a factor of two. Specifically, given a graph $G = (\mathcal{V}, A)$, the selection matrix P (for selecting the vertices) and the resulting downsampled graph $G_k = (\mathcal{V}_k, A_k)$, we define

$$A_k = PA^2P^T$$

The above equation is obtained by direct analogy from bipartite graphs (refer Theorem 2.1).

In order to test the effect of filtering using the proposed adjacency matrix, we generated random bipartite graphs with $N = 100$ nodes. We then added edges to the graph making it non-bipartite. We quantified the amount of edges added as a ratio of Frobenius norm of added matrix to the Frobenius norm of resultant

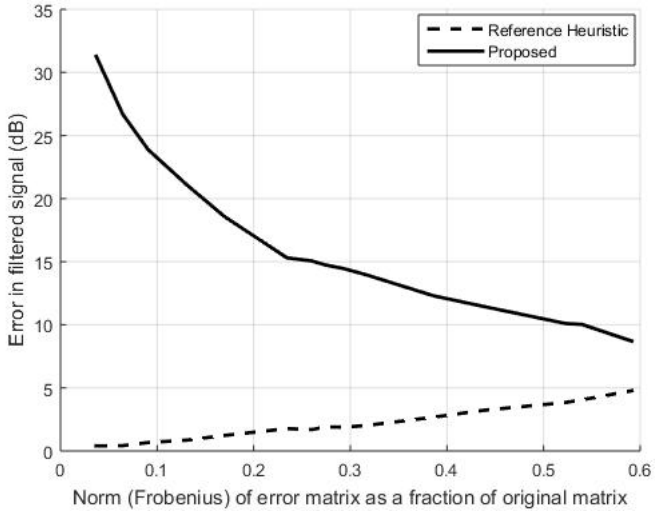

Figure 3.9: Effect of filtering using two different (heuristics method [24] and the the proposed method) adjacency matrices

non-bipartite matrix. We filtered the signal with filter $h(A) = A$.[5] The results are expressed in Figure 3.9. We can see that the proposed method performs with better quality compared to the reference heuristic method [24]. In the reference method, the edges between vertices that were present in the original graph (i.e. graph before downsampling) are preserved in the downsampled graph as well. In addition, new edges are introduced in the downsampled graph based on the second order relations in the original graph.

It is evident that the method proposed for obtaining the adjacency matrix for downsampled graph is limited by the nature of graphs. In order to generalize the concept, there needs to be a deeper analysis on the relationship between the adjacency matrices of two graphs, and the implications of such a relation on the algebra of filters, signal modules and spectral properties. In Chapter 5, we analyze such relationships.

3.6 Discussion

So far, we have assumed downsampling of a graph by an integer factor two. The proposed approach can be adapted to downsampling for any number of samples.

[5] As all the LSI filters can be expressed as polynomials in A, this choice gives us an indication of behavior of any LSI filter.

If a graph with N nodes is to be downsampled into a graph with $m(< N)$ nodes, then we ignore the GFT coefficients corresponding to the $N - m$ highest frequencies. Thus the descriptor submatrix is of dimension $(N - m) \times (N - m)$ and the downsampling scheme can be obtained by optimizing the SDQM.

In subsection 3.1, we mentioned the difference between proposed approach and stable-frame approach. As indicated in [3], stable-frame approach minimizes the aliasing error, which is the same parameter that we optimize. We verify that both the approaches (i.e. proposed approach and stable-frame approach) are identical only if the GFT matrix has orthogonal columns, which is guaranteed for undirected graphs. However, both the methods provide different downsampling schemes in case of directed graphs which have non-orthogonal GFT. Example given below provides such a case where the SDQM based optimization differs from stable-frame approach.

Example 1. Consider a $6-$node directed graph with adjacency matrix

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

The four highest minimum singular values for descriptor submatix are $0.59, 0.58, 0.58$ and 0.57 for selections $\{2,5,6\}, \{1,2,5\}, \{2,3,6\}$ and $\{1,2,3\}$ respectively. On the other hand, the four highest minimum singular values for the selected frame are $0.46, 0.46, 0.36$ and 0.36 for selections $\{1,4,5\}, \{4,5,6\}, \{3,4,6\}$ and $\{1,3,4\}$ respectively. Thus, the optimization using the two parameters would yield different downsampling schemes.

In previous section, we explored how the adjacency matrix for the downsampled set of vertices can be inferred using the analogy with bipartite graph. Here, we discuss how the adjacency matrix can be determined using algebraic properties for an arbitrary graph. We know that $A = VJV^{-1}$. For graphs that are

diagonalizable, J is a diagonal matrix which contains eigenvalues of A as diagonal entries. As given in Equation (3.4), the GFT for the downsampled graph F_{kL} is known. We also know the original graph and the frequencies (and hence eigenvalues) which are preserved in the downsampled graph. Combining this, we can obtain the new adjacency matrix as $V_{kL}J_LV_{kL}^{-1}$, where J_L is a diagonal matrix derived using eigenvalues which are preserved (i.e. the eigenvalues corresponding to the frequencies to be preserved while downsampling) and $V_{kL} = F_{kL}^{-1}$. It can be verified that this approach provides the desired adjacency matrix for downsampling of graphs related to classical signal processing (i.e. graphs whose GFT is the DFT). However, when applied to arbitrary graphs, this approach does not preserve the properties of symmetry and sparseness of the original adjacency matrix. Specifically, $V_{kL}J_LV_{kL}^{-1}$ is not guaranteed to be symmetric (as the matrix V_{kL} is not necessarily orthogonal) or sparse. However, when matrix V_{kL} contains orthogonal columns (e.g. classical signal processing), this approach provides a way to obtain the adjacency matrix for the downsampled set of vertices.

In [24], the adjacency matrix for the downsampled graph is used for obtaining multiresolution. However, we can also obtain a multiresolution on a given graph without assigning the adjacency matrices to the downsampled graphs. Here, we use the fact that the proposed downsampling method relies only on the GFT matrix and not on the adjacency matrix in a direct way. We can also compute the GFT matrix for the downsampled graph using Equation (3.4). Thus, the same algorithm can be used on the new GFT matrix in order to obtain downsampling of the same. Repeating this process, we can obtain a multiresolution on a given graph.

A converse problem for graph signal processing is upsampling of graphs, i.e., adding vertices to a given graph (or merging two graphs) to produce a larger graph. One way to resolve this is to use eigenvalues and eigenvectors of smaller graphs and create a larger graph combining the two. Such a process can be shown to produce appropriate results for DFT-graphs. However, for arbitrary graphs, this problem requires multiple considerations, such as the resultant graph has to have properties like sparsity and directedness from the graphs to be merged. This

requires further research in that direction.

In this chapter, we proposed a downsampling schemes that is applicable to arbitrary graphs. We framed the downsampling problem as an optimization problem and provided a greedy algorithm in order to obtain a suboptimal solution. We validated the proposed approach and compared the same with various existing approaches. We also discussed a few possible ways to obtain the adjacency matrix for downsampled graphs. In the next chapter, we analyze and improve upon the computational complexity of the greedy downsampling algorithm and study the frequency sensitivity of a given sample-set with respect to various frequency bands.

CHAPTER 4

Optimization of Algorithm for SDQM Downsampling and Frequency Selective Downsampling

In the previous chapter, we derived a graph downsampling approach based on minimization of aliasing error of a signal. We formulated the graph downsampling problem as an optimization problem which was solved using a greedy algorithm. In this chapter, we analyze and explore ways to reduce the computational complexity of the proposed greedy algorithm. Additionally, the approach derived in previous chapter has a limitation that it treats the whole higher frequency components as a single band. We extend the same and design downsampling schemes that are sensitive to different frequency bands. We validate our claims using experimental results.

4.1 Reducing the Complexity of SDQM Based Greedy Algorithm

In previous chapter, we derived SDQM based graph downsampling as maximization of $\sigma_{min}(F_4)$, where F_4 is the descriptor submatrix for the given downsampling scheme. The computation of σ_{min} requires $\mathcal{O}(N^3)$ operations. In order to find the optimal solution to this optimization problem, we require to compute σ_{min} for $\binom{N}{N/2}$ (considering the problem of downsampling by a factor of 2) times. This problem cannot be solved using a known polynomial time algorithm. Thus, in

order to obtain an approximate solution, we use a greedy algorithm, which requires computation of σ_{min} for $\mathcal{O}(N^2)$ times. This results in the computational complexity of the proposed greedy algorithm to be $\mathcal{O}(N^5)$. This computational cost is impractical for larger graphs.

In order to reduce computational complexity, we focus on the computation of σ_{min}. Notice that maximization of $\sigma_{min}(F_4)$ is identical to minimization of $\sigma_{max}(F_4^{-1})$. This change in formulation does not provide any savings in the computational cost. This is due to the fact that computing the inverse of a matrix as well as computation of largest singular value of a matrix have same complexity (i.e. $\mathcal{O}(N^3)$) as computing the smallest singular value. However, in context of the greedy approach, we can reduce the computational complexity of computing the inverse matrix by distributing the same along iterations. We can also use some alternative approaches to compute the maximum singular value of a matrix (spectral norm of a matrix) which is useful in reduction of overall computational cost.

4.1.1 Reorganization of Greedy Algorithm for Computational Cost

The present organization of the algorithm is about selection of $N/2$ columns from N given columns. We select a column and then iteratively add columns to the selection such that the minimum singular value of the current iteration is maximized. To this end, we form an $N/2 \times i$ matrix S_i for ith iteration, and compute the minimum singular value of the same and add a column that maximizes the same. The minimum singular value of S_i is the square-root of minimum eigenvalue of $S_i^T S_i$ (which is an $i \times i$ matrix). If the matrix is nonsingular, then minimum singular value can be obtained as $\frac{1}{\sigma_{max}((S_i^T S_i)^{-1})}$. Here, it is clear that we need to compute $(S_i^T S_i)^{-1}$ and σ_{max} of the same in an efficient way in order to reduce the computational complexity.

Now let $j = i + 1$, then S_j is given by $[S_i, c]$, where c is a column to be added to the matrix for next iteration. Thus,

$$S_j^T S_j = \begin{bmatrix} S_i^T S_i & S_i^T c \\ c^T S_i & c^T c \end{bmatrix}$$

If $c^T c$ is non-zero (which is true for every non-zero column) and $S_i^T S_i$ is non-singular, then the matrix is invertible and the inverse can be computed efficiently by following block inversion formula (as given in [20]).

$$(S_j^T S_j)^{-1} = \begin{bmatrix} (S_i^T S_i)^{-1} + (S_i^T S_i)^{-1}(S_i^T c)\Delta(c^T S_i)(S_i^T S_i)^{-1} & -(S_i^T S_i)^{-1}(S_i^T c)\Delta \\ \Delta(c^T S_i)(S_i^T S_i)^{-1} & \Delta \end{bmatrix}, \tag{4.1}$$

where $\Delta = (c^T c - (c^T S_i)(S_i^T S_i)^{-1}(S_i^T c))^{-1}$. If $(S_i^T S_i)^{-1}$ is stored at every iteration, this computation can be accomplished in $\mathcal{O}(N^2)$ operations.[1] Thus, we can compute the inverse of $S_j^T S_j$ efficiently using the inverse of $S_i^T S_i$. The next step is to compute the norm of $(S_j^T S_j)^{-1}$ in an efficient way.

As shown in Equation (3.7), reconstruction error is given by $\|e_r\| \leq \|F_4^{-1}\|_2 \|b_H\|_2$. For any matrix, the spectral norm is equal to or lower than the Frobenius norm. Thus $\|e_r\| \leq \|F_4^{-1}\|_F \|b_H\|$. Since computation of Frobenius norm is $\mathcal{O}(N^2)$, using the same in place of spectral norm (which is $\mathcal{O}(N^3)$ when computed using SVD expansion) results in computational gain. It should be noted here that the Frobenius norm provides a more relaxed upper-bound compared to the one provided by spectral norm. This may result in small degradation of the SDQM.

If it is undesirable to use the Frobenius norm, then we may compute the minimum singular value of S_i as the inverse largest eigenvalue of $(S_i^T S_i)^{-1}$. As discussed above, $(S_i^T S_i)^{-1}$ can be computed efficiently using iterative nature of algorithm. Additionally, we can use the power-method in order to compute the largest eigenvalue of the same. Power method [13] is widely used to compute the dominant eigenvalue of a given matrix (this is especially true if the matrix under consideration is sparse). It is an iterative method of computation where the rate of convergence depends upon the ratio of largest eigenvalue to second largest eigenvalue.[2]

[1] The order of evaluation of various terms in the above inversion formula needs to be examined closely in order to achieve computational gain.

[2] In order to test the efficacy of the power method in computing the spectral norm, we used randomly generated symmetric matrices (each entry of the matrix is drawn independently from a uniform distribution) of sizes varying from 10×10 to 1000×1000 and used the power method

Downsampling Method	SDQM
Algorithm 1	0.15
Using Frobenius Norm (M1)	0.124
Using Power Method (M2)	0.14

Table 4.1: Optimized SDQM based downsampling and corresponding SDQMs

Now, we experimentally validate the methods proposed for computational gains.

4.1.2 Experimental Validation

The data used in the experiment is temperature data from weather stations, publicly available on [1] and also used in our experiments in Chapter 3. From the database, we consider 196 nodes from which directed and undirected graphs are constructed. Data for year 2014 is considered with data available on all nodes for 365 days. Thus, we have 365 graph signals with number of nodes being 196. We construct the adjacency matrix for the graph using correlation matrix. We downsample this graph using Algorithm 1, the Frobenius norm based approach and power method based approach. We note the SDQM values obtained for three different downsampling schemes and observe the reconstruction accuracy.

Table 4.1 lists the SDQM values for the downsampled sample-sets obtained by various methods. Note that the degradation in SDQM is relatively low. To observe the effect of the same on signal reconstruction, refer to Figure 4.1 and Figure 4.2. From Figure 4.1, it is clear that the reconstruction quality of the optimized approach is close to the one obtained using Algorithm 1. As a reference, outcome of MST based downsampling is also provided. More interestingly, the approach using the power method (refer Figure 4.2) tracks so closely to one provided by Algorithm 1 that the two are not distinguishable in the figure. In the figures and Table 4.1, the approach using Frobenius norm and the one using power method are referred to as M1 and M2 respectively.

with iteration count of $log_{10}(N)$. Thus the spectral norm can be computed in $\mathcal{O}(N^2 log(N))$. We found that the mean normalized error in the spectral norm is of the order of 10^{-16}, which is adequate for our purpose.

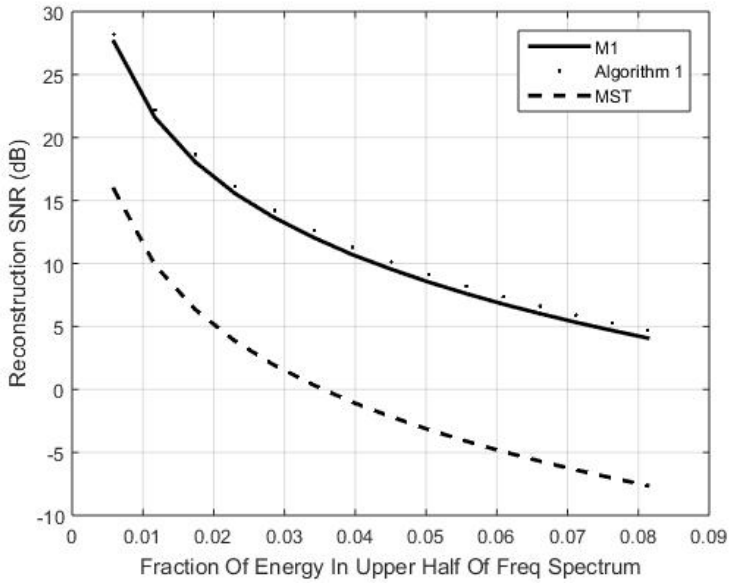

Figure 4.1: Reconstruction quality using algorithm based on Frobenius norm (M1). Reconstruction qualities using Algorithm 1 and MST based approach are provided as reference.

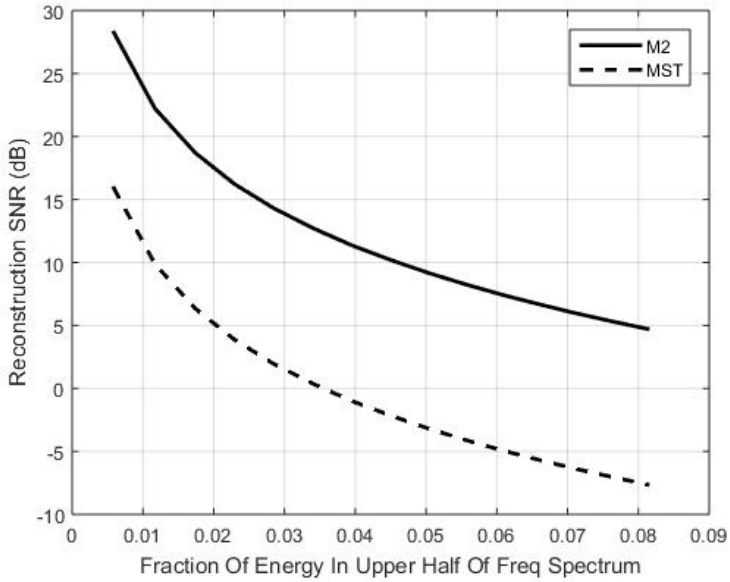

Figure 4.2: Reconstruction quality using algorithm based on Power Method (M2). Reconstruction quality using MST based approach provided as reference.

4.2 Frequency Selective Downsampling Scheme

The optimization parameter SDQM indicates the sensitivity of a sample-set with respect to upper-half of the frequency spectrum (higher SDQM implies lower sensitivity). In this section, we extend this concept to individual frequencies and frequency bands. We begin by defining a frequency sensitivity function and then use the same in order to obtain different sampling schemes with desired sensitivities in different frequency bands. The frequency sensitivity function defined below measures the maximum normalized error in signal reconstruction when the signal contains only the frequencies in the given band.

Definition For a given graph, let a band of frequencies be denoted by β, let V_β be the set of signals spanned by the eigenvectors corresponding to β, then the sensitivity of a sample-set with respect to the band β, is defined as

$$T(\beta) = \sup_{\bar{s} \in V_\beta} \frac{\|\bar{s} - \bar{s_r}\|}{\|\bar{s}\|} \tag{4.2}$$

under a given reconstruction method, where $\bar{s}_r$ is the reconstructed signal using the sample-set.

Theorem 4.1 *For a given graph, and a given sample-set and its descriptor submatrix as F_4,*

1. The frequency sensitivity of any sample-set with respect to the preserved frequencies (e.g. lower half f frequencies for downsampling by a factor of two) is always zero.

2. The sensitivity towards a band of frequencies is given by spectral-norm of the matrix consisting of the corresponding columns of the matrix F_4^{-1}.

Proof 1. If a signal is bandlimited to lower frequencies, it can be reconstructed without any error from a given sample-set. Thus, the frequency sensitivity, as defined earlier, is always zero.

2. The error in reconstruction is given by Equation (3.7) as $\|F_4^{-1}\bar{b}_H\|_2$. Given a signal confined to a frequency band, the contribution to the error is only the non-zero coefficients in $\bar{b}_H$, and hence the corresponding columns of matrix F_4^{-1}. Thus

the normalized error is given by the spectral norm of the submatrix constructed by corresponding columns of F_4^{-1}. ∎

Using Theorem 4.1, we can find the sensitivity to a given frequency band β as,

$$T(\beta) = \sigma_{max}(F_4^{-1}(\beta))$$

where $F_4^{-1}(\beta)$ is the submatrix of F_4^{-1} corresponding to the columns selected by β.

The graph whose GFT is the Discrete Fourier Transform (DFT), the frequency sensitivity of each frequency is uniform for the standard downsampling scheme (i.e. selecting every alternate sample). Also, the frequency sensitivity of each band with identical number of frequencies is identical. We use the results in Theorem 4.1 in order to obtain a greedy approach for designing frequency sensitive downsampling schemes.

4.2.1 Downsampling Algorithm Based on Optimizing Frequency Sensitivity

We now explore the details of an approach that would allow us to design a desired frequency sensitive sampling scheme using Theorem 4.1. Given the descriptor submatrix F_4, we are required to compute F_4^{-1} and then use the desired frequency sensitivity as the optimization parameter. We consider two bands in the higher frequency range, which are required to have desired relative sensitivity.

To simplify the details, let us assume that the graph has N nodes, where N is divisible by 4. Consider the case of downsampling by a factor of two, thus the higher frequencies constitute the upper half of the frequencies (Denoted by β_H). Within the upper-half of frequencies, let us assume that there are two equal sized frequency bands, i.e. the lower portion of the upper-half (denoted by β_{HL} and the higher portion of the upper-half (denoted by β_{HH}). We desire our downsampling scheme to be less sensitive towards β_{HL} and more sensitive towards β_{HH}. This assumption is valid in cases where the signal energy is significantly higher in band

β_{HL} compared to the energy in band β_{HH}. The following optimization strategy can be used in order to obtain the desired sample-set.

$$P_{opt} = \underset{P_p \in \{0,1\}^{N/2 \times N}}{\operatorname{argmin}} \{\alpha T(\beta_{HL}) + (1-\alpha) T(\beta_{HH})\} \tag{4.3}$$

where $\alpha \in (0,1)$ is used to adjust the sensitivity to different bands. For example, setting α closer to 1 would put more emphasis on the band β_{HL}. It should be noted here that setting $\alpha = 1$ may result in an unstable frame for reconstruction. Therefore, α should not be selected to be too close to 0 or 1.

The optimization problem given by Equation (4.3) can not be solved using a polynomial time algorithm. Thus a suboptimal greedy approach is given in Algorithm 2, which can solve the problem in polynomial time. In Algorithm 2, F_H is the GFT submatrix selected on upper-half of frequencies (i.e. rows). The frequency bands β_{HL} and β_{HH} are known and we begin with an initial sample-set $\mathcal{V}_{init}$. As computation of each $T(\beta_{HL})$ and $T(\beta_{HH})$ require computation of F_4^{-1}, the algorithm differs from Algorithm 1 by avoiding rectangular matrices. The Algorithm 2 changes the initial sample-set $\mathcal{V}_{init}$ in order to obtain desired frequency sensitivity for bands. As the algorithm does not intend to optimize the overall SDQM of the sample-set, we must ensure the $\mathcal{V}_{init}$ has acceptable SDQM. To address the issue, the set $\mathcal{V}_{init}$ is selected using Algorithm 1.

Algorithm 2: Greedy algorithm for frequency selective downsampling

$F_H, \beta_{HL}, \beta_{HH}, \mathcal{V}, \mathcal{V}_{init}$
$\mathcal{V}_k \leftarrow \mathcal{V}_{init}$
$\mathcal{V}_p \leftarrow \mathcal{V} - \mathcal{V}_k$
$F_{Hk} \leftarrow$ Columns of F_H corresponding $\mathcal{V}_k$
$F_{Hp} \leftarrow$ Columns of F_H corresponding $\mathcal{V}_p$
FOR $i = 1 \cdots N/2$
  Swap $i-th$ column of F_{Hk} with a column of F_{Hp} that minimizes:
 $\alpha T(\beta_{HL}) + (1-\alpha) T(\beta_{HH})$
  Update $\mathcal{V}_k$ and $\mathcal{V}_p$
  Update F_{Hk} and F_{Hp}
ENDFOR
RETURN $\mathcal{V}_k$

Approach	$SDQM$	$T(\beta_{HL})$	$T(\beta_{HH})$
Uniform	0.1502	5.98	3.52
Weighted	0.1492	5.27	4.52

Table 4.2: SDQM and Frequency Sensitivity for two approaches

4.2.2 Experimental Validation

Now, we use the frequency selective optimization and Algorithm 2 in order to obtain the desired downsampling scheme for the graph created on temperature data. We select 196 weather stations and 365 days data for temperature across US in year 2014 [1]. We construct an undirected graph using statistical correlation. We find out a stable reconstruction frame using the SDQM optimized approach. As this approach does not impose different weightage to different frequency bands, it is referred to as uniform approach. The outcome of this approach is used as reference. We use the proposed approach in order to optimize the selected sample-set with respect to the band β_{HL}. This results in small degradation in SDQM and a relatively larger trade-off in $T(\beta_{HH})$. The sensitivities and SDQMs are summarized in Table 4.2. The sensitivity $T(\beta_{HL})$ reduces from 5.98 to 5.27, thus the sample-set obtained exhibits lower sensitivity to changes in β_{HL}.

In order to test our hypothesis, we modify the temperature data to have very low energy content in band β_{HH}. After this, we alter the content in the band β_{HL} to test the effect of both the sample-sets. As seen in Figure 4.3, even though the SDQM for the weighted approach is marginally lower than the uniform approach, the error in the uniform approach is higher than that in the weighted approach. It can also be seen that the margin of error increases as the energy of content in the band β_{HL} increases.

In this chapter, we analyzed the greedy algorithm for downsampling presented in Chapter 3, in the context of computational cost and proposed two methods to reduce the same. We also analyzed the downsampling schemes with respect to their sensitivity towards different frequency bands and provided an approach to obtain a downsampling scheme with desired frequency sensitivity. In

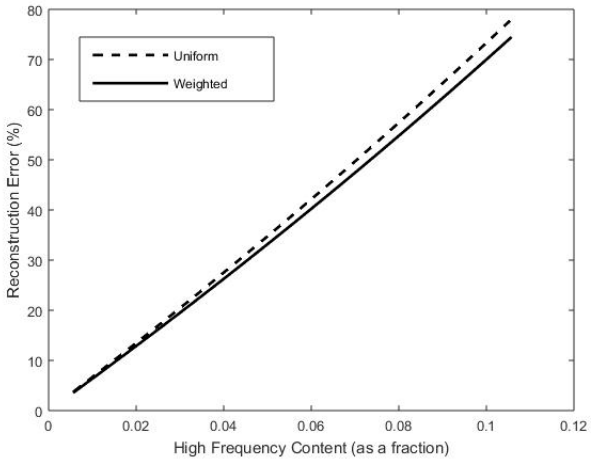

Figure 4.3: The errors for the temperature data by two approaches. *Uniform* approach indicates non frequency selective approach while *Weighted* approach indicates the proposed frequency selective optimization. High frequency content indicates the relative energy in band β_{HL}

the next chapter, we study GSP-isomorphism and GSP-homomorphism and explore their relationships with resampling and downsampling process.

CHAPTER 5

Structure Preserving Maps

The phrase *Structure Preserving Map* (SPM) refers to how one graph can relate to another graph, preserving certain algebraic properties such as algebra of filters and module of signals. Our main objective is to study the signal processing properties of graph theoretical concepts such as homomorphism and isomorphism. We study how the adjacency matrices are related to each other and what constraints are imposed upon the adjacency matrices for the homomorphism and isomorphism to hold. To explain this, we present an example which provides us with insights into SPM between graphs.

Example: As noted earlier, CSP can be interpreted as a special case of GSP via an appropriate assignment of the adjacency matrix [28, 29]. Consider graph provided in Figure 5.1, which represents periodic sequence with period 4. The GFT corresponding to this graph is given by 4-point DFT (Discrete Fourier Transform). Figure 5.2 shows a graph obtained by applying a similarity transform with the following matrix P on the adjacency matrix of graph in Figure 5.1.

$$P = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -i \\ -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & i \end{bmatrix}$$

Consider a signal on graph in Figure 5.1, $\bar{s}_1 = [1, 0, 0, 0]^T$. Let the image of this signal on the graph in Figure 5.2 be $\bar{s}_2$, such that $\bar{s}_1 = P\bar{s}_2$. Let a filter $H \mapsto h(x)$ be $h(x) = x + x^2$. The output of this filtering operation is $h(A_1)\bar{s}_1 = [0, 1, 1, 0]^T$. The

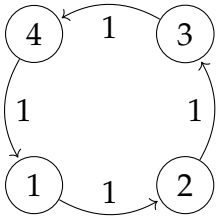

Figure 5.1: 4-node graph representing classical DSP as a graph

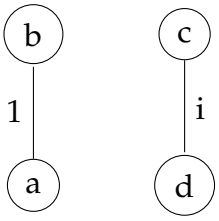

Figure 5.2: A graph obtained by a similarity transform on adjacency matrix in 5.1

output of the same filtering operation in second graph is $h(A_2)\bar{s}_2 = \frac{1}{2}[-1, -1, -1, i]$. It can be seen that $h(A_1)\bar{s}_1 = Ph(A_2)\bar{s}_2$. Thus, the linear map P preserves the filtering operation.

The above example shows us that there exist maps between graphs that preserve the filtering operation. Our aim is to study such maps and the conditions on the graphs which allow for the preservation of filtering operations.

5.1 Structure on Graphs

In this section, we first introduce *graph signal processing framework* (GSPF) and then define structure preserving maps (more specifically, homomorphic and isomorphic maps) between two graph signal processing frameworks (or *structure preserving maps between graphs* in brief).

Let $G = (\mathcal{V}, A)$ be the graph under consideration. Let S be a set of signals on a graph G, and F be a set of linear and shift-invariant systems that operate on the signals. Note that the sets S and F themselves also have structures of their own. The set of signals S is a vector space, while the set of filters F forms an algebra with well defined addition, scalar multiplication and composition operations. As per Algebraic Signal Processing Theory, we know that S can be seen as an $F - module$, and forms an algebraic module. We refer to the structure $(S, F, \cdot)$ as *graph signal*

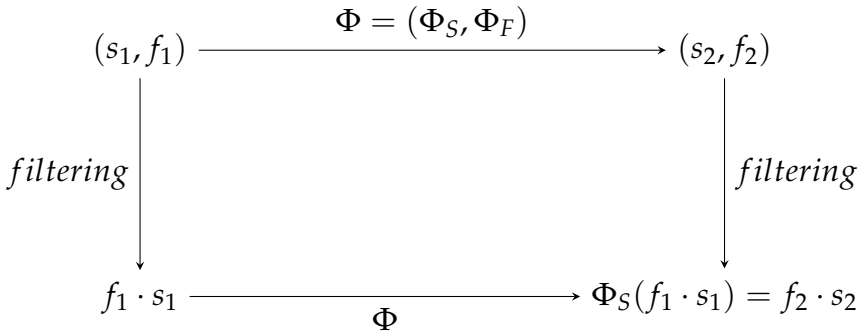

Figure 5.3: Illustration of structure preserving map Φ between two GSPFs. A pair of signal and filter (s_1, f_1) on one GSPF is mapped to a pair (s_2, f_2) (where $s_2 = \Phi_S(s_1)$ and $f_2 = \Phi_F(f_1)$) on the other. The filtering operation produces signals $f_1 \cdot s_1$ and $f_2 \cdot s_2$ respectively.

processing framework for graph G. A fundamental property of an GSPF $(S, F, \cdot)$ is that for any signal $s \in S$ and any filter $f \in F$, $f \cdot s$ is a well defined operation with $f \cdot s \in S$.

Definition Consider two graphs G_1 and G_2 with their GSPFs $(S_1, F_1, \cdot_1)$ and $(S_2, F_2, \cdot_2)$ respectively. If linear maps $\Phi_S : S_1 \to S_2$ and $\Phi_F : F_1 \to F_2$ exist such that

$$\forall f_1 \in F_1, \forall s_1 \in S_1, \Phi_S(f_1 \cdot_1 s_1) = \Phi_F(f_1) \cdot_2 \Phi_S(s_1), \tag{5.1}$$

then $\Phi = (\Phi_S, \Phi_F)$ is called *GSP-homomorphism* and the GSPFs $(S_1, F_1, \cdot_1)$ and $(S_2, F_2, \cdot_2)$ are said to be *homomorphic*.

For maintaining simplicity, we will omit symbols $\cdot_1$ and $\cdot_2$, as the specifics of filtering can be obtained from context of the expression. For example, instead of writing $f_1 \cdot_1 s$, we will write $f_1 \cdot s$.

As Φ_S and Φ_F are linear maps, and considering the fact that f_1 and f_2 are linear transforms, an important relation between Φ_S and Φ_F can be obtained, which is provided in form of Theorem 5.1.

Theorem 5.1 *If $\Phi = (\Phi_S, \Phi_F)$ defines a GSP-homomorphism between two GSPFs (S_1, F_1) and (S_2, F_2), then*

$$\forall f \in F_1, \Phi_F(f)\Phi_S = \Phi_S f \tag{5.2}$$

Proof By definition of homomorphism, $\Phi_S(f \cdot s) = \Phi_F(f) \cdot \Phi_S(s)$. As, Φ_S and f both are linear transforms, they follow associative laws. Hence, we can write $(\Phi_F(f)\Phi_S)s = (\Phi_S f)s$. As this relation has to be true for all signals $s \in S_1$, following result can be obtained which proves the statement.

$$\Phi_F(f)\Phi_S = \Phi_S f \quad \blacksquare \tag{5.3}$$

Theorem 5.2 *If $\Phi = (\Phi_S, \Phi_F)$ defines a GSP-homomorphism between two GSPFs (S_1, F_1) and (S_2, F_2), then*

$$\forall f_1, f_2 \in F_1, \Phi_F(f_1 f_2)\Phi_S = \Phi_F(f_1)\Phi_F(f_2)\Phi_S \tag{5.4}$$

Proof Using Theorem 5.1, $\Phi_F(f_1 f_2)\Phi_S = \Phi_S f_1 f_2$. Also, $\forall s \in S_1, \Phi_S(f_1(f_2 \cdot s)) = \Phi_F(f_1)\Phi_S(f_2 \cdot s) = \Phi_F(f_1)\Phi_F(f_2)\Phi_S(s)$. As this has to be true $\forall s \in S_1$, we get $\Phi_S f_1 f_2 = \Phi_F(f_1)\Phi_F(f_2)\Phi_S$, which proves the desired result. $\blacksquare$

If $\Phi = (\Phi_S, \Phi_F)$ defines a GSP-homomorphism between two GSPFs (S_1, F_1) and (S_2, F_2), and Φ_S is surjective, then $\Phi_F(f) = \Phi_S f \Phi_S^{R-}$, where Φ_S^{R-} is right-inverse of Φ_S. It must be noted here that Φ_S^{R-} is not unique. We now introduce GSP-isomorphism based on GSP-homomorphism already defined.

Definition Let $\Phi = (\Phi_S, \Phi_F)$ be a GSP-homomorphism. If Φ_S is bijective, then the mapping is called GSP-isomorphism.

Theorem 5.3 *If $\Phi = (\Phi_S, \Phi_F)$ defines a GSP-isomorphism, then $\Phi_F(f) = \Phi_S f \Phi_S^{-1}$; where Φ_S^{-1} is the inverse of Φ_S.*

Proof As per definition of GSP-isomorphism, Φ_S is bijective (i.e. surjective and injective). Thus, using Theorem 5.1, we get $\Phi_F(f)\Phi_S = \Phi_S f$. However, as Φ_S is bijective, with inverse Φ_S^{-1}, we obtain $\Phi_F(f) = \Phi_S f \Phi_S^{-1}$. $\blacksquare$

If Φ_S is an invertible linear map, then we can write $f = \Phi_S^{-1}\Phi_F(f)\Phi_S$. Thus Φ_F is also an invertible linear map with $\Phi_F^{-1}(f) = \Phi_S^{-1} f \Phi_S$. Thus, the linearity of Φ_F is not independent of the nature of Φ_S. In addition, if Φ_S is invertible, then $\forall f_1, f_2 \in F, \Phi_F(f_1 f_2) = \Phi_F(f_1)\Phi_F(f_2)$. Theorem 5.4 tells us that GSP-homomorphism is transitive.

Theorem 5.4 *If (S_1, F_1), (S_2, F_2) and (S_3, F_3) are three GSPFs such that there exists a GSP-homomorphism between (S_1, F_1) and (S_2, F_2), and between (S_2, F_2) and (S_3, F_3), then there exists a GSP-homomorphism between (S_1, F_1) and (S_3, F_3).*

Proof Let $\Phi = (\Phi_S, \Phi_F)$ be the GSP-homomorphism between (S_1, F_1) and (S_2, F_2). Also let $\Psi = (\Psi_S, \Psi_F)$ be the GSP-homomorphism between (S_2, F_2) and (S_3, F_3). Now, let $\Gamma_S = \Psi_S \circ \Phi_S$ and $\Gamma_F = \Psi_F \circ \Phi_F$. Using definition of GSP-homomorphism, we obtain

$$\forall f_1 \in F_1, s_1 \in S_1,\ \Gamma_S(f_1 s_1) = \Psi_S \circ \Phi_S(f_1 s_1)$$

$$\Rightarrow \Gamma_S(f_1 s_1) = \Psi_S(\Phi_S(f_1 s_1)) = \Psi_S(\Phi_F(f_1)\Phi_S(s_1)) = \Psi_F \circ \Phi_F(f_1)\Psi_S \circ \Phi_S(s_1)$$

$$\Rightarrow \Gamma_S(f_1 s_1) = \Gamma_F(f_1)\Gamma_S(s_1) \quad \blacksquare \tag{5.5}$$

Through Theorems 5.1-5.4, we have focused on linearity and not explored the shift-invariance aspect of the filters. As discuss earlier, the LSI filters on graphs are polynomials in adjacency matrix. Hence, GSP-homomorphism and GSP-isomorphism put constraints on adjacency matrices of the two graphs under consideration. We examine the same in detail in following sections.

5.2 GSP-Isomorphism

Consider two graphs $G_1 = (\mathcal{V}_1, A_1)$ and $G_2 = (\mathcal{V}_2, A_2)$ and let $\Phi = (\Phi_S, \Phi_F)$ be a GSP-isomorphism between the two. Let the number of vertices in G_1 and G_2 be N_1 and N_2, respectively. We limit the set of filters in GSPF to be Linear and Shift-Invariant. Let the GSPFs for graphs G_1 and G_2 be (S_1, F_1) and (S_2, F_2), respectively. If $N_1 = N_2$,[1] then GSP-isomorphism turns into graphs being isospectral, which is explained in Section 5.2.1.

[1] If $N_1 \neq N_2$, then the graphs can have GSP-isomorphism only if the dimension of signal spaces S_1 and S_2 are identical. Here, the vectors representing the signals will have dimensions $N_1 \times 1$ and $N_2 \times 1$, however, the signal could belong to a lower dimensional subspace (e.g. bandlimited signals). Limiting of the graph-signals to a subspace also has a limiting effect on the set of filters as the closure needs to be maintained. Thus, the GSP-isomorphism between two graphs can be achieved by limiting the signal-space and the filter-space for graph with more nodes to subspaces of dimensions that comply with the other graph.

5.2.1 Isospectral Graphs

In this section, we explore the properties of *isospectral graphs*[2] (defined below). We show that isospectral graphs have a GSP-isomorphism between them. We also show that isospectral graphs exhibit one to one correspondence between signals and filters of two graphs that preserves the algebraic properties.

Definition Two graphs $G_1 = (\mathcal{V}_1, A_1)$ and $G_2 = (\mathcal{V}_2, A_2)$ are called isospectral iff matrices A_1 and A_2 are similar. They are also called *co-spectral* graphs.

The concept of isospectral graphs can be thought of as an extension to the concept of graph-isomorphism. As Φ_S is a linear map, it can be represented as a matrix P. In graph-isomorphism, the matrix that relates the two adjacency matrices (i.e. matrix P) is a permutation matrix, whereas in isospectral graphs, P is any invertible matrix. The following result shows that isospectral graphs also share a GSP-isomorphism.

Theorem 5.5 *If two graphs are isospectral, a GSP-isomorphism can be established between the two.*

Proof Let graphs $G_1 = (\mathcal{V}_1, A_1)$ and $G_2 = (\mathcal{V}_2, A_2)$ be isospectral. By definition, there exists an invertible matrix P such that $A_1 = P^{-1}A_2P$. If we define $s_1 \in S_1, \Phi_S(s_1) = Ps_1$ and $f_1 \in F_1, \Phi_F(f_1) = Pf_1P^{-1}$, then we can conclude by definition that (Φ_S, Φ_F) constitutes a GSP-isomorphism between G_1 and G_2. ∎

Theorem 5.6 *Given two isospectral graphs $G_1 = (\mathcal{V}_1, A_1)$ and $G_2 = (\mathcal{V}_2, A_2)$ with $A_1 = P^{-1}A_2P$, then*
(i) *If H_1 is the matrix representing a shift-invariant filter in G_1, then there exists H_2 similar to H_1 which is a shift-invariant filter in G_2.*
(ii) *If H_1 be a shift-invariant filter in graph G_1, then filter $H_2 = PH_1P^{-1}$ in graph G_2 has identical frequency response to H_1.*
(iii) *Let H_1 be a shift-invariant filter in graph G_1. If filter H_2 in graph G_2 has identical frequency response to H_1, then $H_2 = PA_1P^{-1}$.*

[2]The isospectral relations between trees are studied in [35]

Proof (i) H_1 is a shift-invariant matrix in G_1. With A_1 being the shift operator for G_1, then $H_1A_1 = A_1H_1$.

Using $A_1 = P^{-1}A_2P$, $H_1P^{-1}A_2P = P^{-1}A_2PH_1$. $\Rightarrow PH_1P^{-1}A_2 = A_2PH_1P^{-1}$

Substitute $H_2 = PH_1P^{-1}$, we get $H_2A_2 = A_2H_2$. Hence, we can conclude that H_2 is shift-invariant in G_2.

(ii) Since the two graphs are similar, their adjacency matrices have identical set of eigenvalues and hence their characteristic polynomials are identical. Let's assume the characteristic polynomials to be $p(x)$.

Also assume that the filter H_1 in graph G_1 is mapped to a polynomial $h_1(x) \in \mathbb{C}[x]/p(x)$. Similarly H_2 in graph G_2 is mapped to a polynomial $h_2(x) \in \mathbb{C}[x]/p(x)$. Thus, $H_1 = h_1(x)|_{x=A_1} = h_1(A_1)$ and $H_2 = h_2(x)|_{x=A_2} = h_2(A_2)$. Now, $H_1 = h_1(A_1) \Rightarrow H_1 = \sum_{i=0}^{N-1} \alpha_i A_1^i$. Using relation $A_1 = P^{-1}A_2P$, $H_1 = \sum_{i=0}^{N-1} \alpha_i (P^{-1}A_2P)^i$ $\Rightarrow H_1 = P^{-1}(\sum_{i=0}^{N-1} \alpha_i A_2^i)P \Rightarrow H_1 = P^{-1}h_1(A_2)P$. Combine the above result with the fact that $H_1 = P^{-1}H_2P$, we get $H_2 = h_1(A_2) \Rightarrow H_2 = h_1(x)|_{x=A_2}$. The above result indicates $h_1(x) = h_2(x)$. Hence, the graph Fourier Transform of both H_1 and H_2 are also identical.

(iii) Result **(iii)** is the converse of **(ii)**, and it can be proven using similar line of arguments. ∎

Theorem 5.7 *Given two isospectral graphs $G_1 = (\mathcal{V}_1, A_1)$ and $G_2 = (\mathcal{V}_2, A_2)$ with $A_1 = P^{-1}A_2P$, then*

***(i)** If V_1^{-1} and V_2^{-1} are the matrices representing graph Fourier transforms for graph G_1 and G_2 respectively, then the matrix P is given by $P = V_2V_1^{-1}$*

***(ii)** The GFT of a signal on G_1 and its image on G_2 are identical.*

Proof (i) As V_1^{-1} and V_2^{-1} are the graph Fourier transforms respectively, A_1 and A_2 can be written as, $A_1 = V_1J_1V_1^{-1}$ and $A_2 = V_2J_2V_2^{-1}$, where J_1 and J_2 are Jordan Normal Forms of A_1 and A_2 respectively.

As matrices A_1 and A_2 are similar, their JNFs are identical. Hence, $J_1 = J_2 \Rightarrow V_1^{-1}A_1V_1 = V_2^{-1}A_2V_2 \Rightarrow A_1 = V_1V_2^{-1}A_2V_2V_1^{-1}$. Since $A_1 = P^{-1}A_2P$, the desired result can be achieved as $P = V_2V_1^{-1}$

(ii) Let signal on G_1 be $\bar{s}_1$, and its image on G_2 is given by $\bar{s}_2 = P\bar{s}_1$. As $P = V_2V_1^{-1}$, we obtain $V_1^{-1}\bar{s}_1 = V_2^{-1}\bar{s}_2$, which is the desired result. ∎

Using Theorem 5.6, one can transfer definitions of shift, shift-invariance and linear shift-invariant filters from one graph to a graph which is isospectral to it. There is a one-to-one correspondence between the set of filters and signals on the two isospectral graphs. One of the applications of isospectral graphs is discussed in [38], where a given graph is transformed into a graph with desired eigenvector structure for the purpose of multirate signal processing on graphs. Instead of imposing a structure on the eigenvectors, in the next subsection we use the compatibility of shift operators between two graphs to obtain isospectral graphs. Thus the isospectral graphs are not only algebraic manipulations mirroring similarity transforms, but they also have a physical implication on the shift operator. In the following subsection, we explain how the resampling process leads to isospectral graphs using the example of 1-D nonuniform signals.

5.2.2 Nonuniform Signal Processing: Algebraic Model

We examine the signal processing model for 1-D signals which are non-uniformly sampled. We show that under certain conditions, signals which are non-uniformly sampled can be represented and processed as graph signals with appropriate choice of adjacency matrix. Moreover, if we assume a finite number of continuous time signals as a common basis for various sampling grids, we show that these grids equipped with suitable interrelations between vertices give rise to graphs which share GSP-isomorphism. This can be used for applications like signal reconstruction and filter-design for non-uniform 1-D signals.

Consider a signal $s(t)$ (supported on a continuous domain of values) sampled at two grids one uniform and the other nonuniform, giving two discrete signals $\bar{s}_1$ and $\bar{s}_2$ respectively. Here, we assume that the signal $s(t)$ can be reconstructed without any error from either of the sampled versions. In such a case, the operator P that maps $\bar{s}_1$ to $\bar{s}_2$ is invertible. The shift operator corresponding to the first sampling grid is denoted by A_1 and the shift operator for the second sampling grid is denoted by A_2. Figure 5.4 shows the scenario using a commutative diagram.

As A_1 and A_2 represent shift operators in their respective graphs, the effect of shifting is assumed to be identical to the underlying signal $s(t)$, i.e., the re-

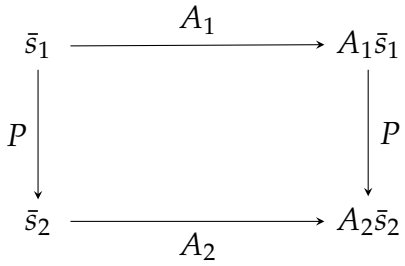

Figure 5.4: Commutative diagram for isospectral graphs. A signal $s(t)$ is sampled on two different grids giving signals $\bar{s}_1$ and $\bar{s}_2$. A_1 and A_2 are the adjacency matrices representing the graphs while P is a linear and invertible operator which allows reconstruction from one grid to another.

constructed shifted version of $s(t)$ is the same whether the reconstruction is done using $A_1\bar{s}_1$ or using $A_2\bar{s}_2$. Our objective here is to find out the adjacency matrix for the non-uniform grid, which will allow us to use GSP to process non-uniform signals. Given A_1 (and its related information such as spectrum and GFT matrix), we would like to find out the adjacency matrix (A_2) for the non-uniform grid. From the commutative diagram in Figure 5.4,

$$PA_1\bar{s}_1 = A_2\bar{s}_2 = A_2P\bar{s}_1 \tag{5.6}$$

As the above condition has to be met for any $\bar{s}_1$,

$$A_2 = PA_1P^{-1} \tag{5.7}$$

This indicates that the matrices A_1 and A_2 are related by a similarity transform. Hence, the respective graphs are isospectral. Let $s(t) \in span\{b_1(t), ..., b_N(t)\}$, where each $b_i(t)$ is a basis vector. Let the samples obtained from uniform grid be denoted by $\bar{s}_1$ and the samples obtained from nonuniform grid be denoted by $\bar{s}_2$, each of which is an N-dimensional vector. Given the N sampling locations $(t_1, ..., t_N)$, we call the matrix $\beta = [b_i(t_j)]_{i,j}$ as the grid basis matrix. We thus get two such grid-basis matrices β_1 and β_2, corresponding to uniform and nonuniform grids respectively. Both β_1 and β_2 are invertible as the grids allow perfect reconstruction from sampled signals. Keeping consistency in notation, denote the uniform DFT matrix as V_1^{-1}. As the underlying continuous signal corresponding

to both the uniform and nonuniform samples is identical, their Fourier transforms should be identical too, i.e., the graph Fourier transform for $\bar{s}_2$ should be identical to the DFT of $\bar{s}_1$. It can be seen that,

$$\forall \bar{s}_1 \in S_1, \bar{s}_2 \in S_2, V_1^{-1}\bar{s}_1 = V_2^{-1}\bar{s}_2 \Leftrightarrow V_1^{-1}\beta_1 = V_2^{-1}\beta_2 \tag{5.8}$$

Using above framework, we derive information required to characterize the graph representation of the non-uniform grid. Primarily, we derive the GFT, spectrum, adjacency matrix and boundary conditions. We will refer to graph representation for non-uniform grid as *non-uniform graph* in short. The GFT on for non-uniform graph is denoted V_2^{-1} and it is given by $V_2^{-1} = V_1^{-1}\beta_1\beta_2^{-1}$. As discussed before, we have all the information needed to compute the GFT matrix V_2^{-1}. Let the Jordan Normal Form of A_2 be given by $A_2 = V_2 J_2 V_2^{-1}$. Here, as A_1 and A_2 are similar, the matrix J_2 is identical to that for A_1. As the eigenvalues of the matrix A_1 are distinct [3], their characteristic and minimal polynomials are identical. This results in J_2 being a diagonal matrix. Thus A_2 can be computed, which is the designated shift operator for the non-uniform graph G_2. Following example explains the usefulness of the isospectral approach of constructing graphs using nonuniform sampling.

Example 1. Consider the signals supported on interval $[0, 4]$ which are sampled on two different grids. The first grid is a uniform grid with sample-points given by $(0, 1, 2, 3, 4)$. To illustrate the effect of isospectral graphs, we select the second grid as a non-uniform grid with sample-points, $(0, 1, 2.1, 3, 4)$. Thus there is a single sample location change from grid-1 to grid-2. The adjacency matrix for the uniform grid is given as follows (for details of adjacency matrix, refer [29]),

$$A_1 = \begin{bmatrix} & & & & 1 \\ 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \end{bmatrix}$$

[3] The eigenvalues of matrix A_1 of size N are the N complex roots of unity.

Without using the principle of the isospectral graphs, and using analogy from A_1, one arrives at following adjacency matrix ($\tilde{A}_2$) for nonuniform sampling.

$$\tilde{A}_2 = \begin{bmatrix} & & & & 1 \\ 1 & & & & \\ & 1.1 & & & \\ & & 0.9 & & \\ & & & 1 & \end{bmatrix}$$

However, if we impose an isospectral graph structure described above, we can arrive at following adjacency matrix (A_2) for nonuniform sampling.

$$A_2 = \begin{bmatrix} & & & & 1 \\ 1 & & & & \\ -0.10 & 0.99 & 0.11 & -0.08 & 0.07 \\ -0.06 & 0.09 & 1.01 & -0.11 & 0.06 \\ & & & 1 & \end{bmatrix}$$

Now, consider a signal $s(t) = 1, \forall t \in [0, 4]$. Thus, the sampled versions of the signal on both the grids will be, $\bar{s}_1 = \bar{s}_2 = [1, 1, 1, 1, 1]^T$. It can be seen that $\tilde{A}_2\bar{s}_2 = [1, 1, 1.1, 0.9, 1]$ while $A_2\bar{s}_2 = [1, 1, 1, 1, 1]$. Thus, the isospectral graph provides the desired shift operation on the given signal.

In order to visualize the effect of shift operator produced using the proposed approach, we create a longer duration nonuniform 1D signal by introducing jitter[4] in sampling. We compare the signal shifted with the proposed shift operator and the shift obtained using the adjacency matrix constructed via a distance based approach (an approach similar to one used in [7]). The original signal, and the two shifted signals are shown in Figure 5.5. This demonstrates the efficacy of the matrix constructed using similarity property.

As the matrix A_1 represents the shift operator for uniform grid, the adjacency matrix is a circulant shift matrix, which has characteristic equation as $x^N - 1 = 0$.

[4]Jitter in sampling is the variations in sampling positions from a predetermined uniform pattern

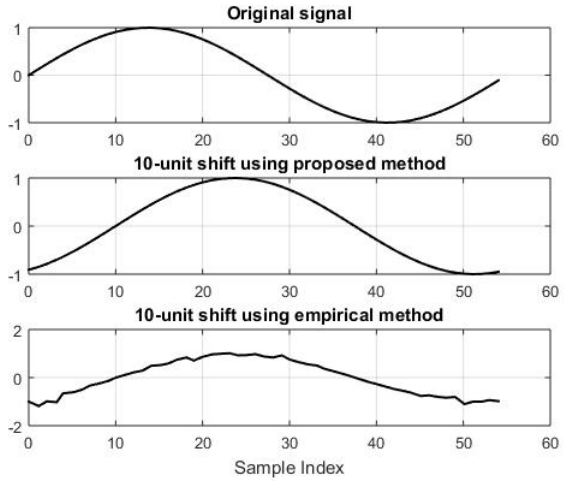

Figure 5.5: An Example of shifting a non-uniform (jittered in this case) signal by means of proposed method of creating adjacency matrix compared to distance based adjacency matrix

As matrix A_2 is similar to A_1, they have identical characteristic equations, which provide us the boundary conditions. For example, shifting a signal N times on either graphs would produce the original signal back.

5.2.3 Filtering using GSP-Isomorphism

Theorem 5.6 provides a way to filter non-uniform data directly, without converting it into uniformly sampled data. One can design filters for non-uniform data using some of the well established methods (see [25]) of uniformly sampled signal filter design. From Theorem 5.6, a filter H_2 on the non-uniform graph is related to a filter on uniform graph H_1 by $H_2 = PH_1P^{-1}$. Consider a signal $\bar{s}_2$ in nonuniform graph which is to be filtered by filter H_2. Using reconstruction formula $\bar{s}_2 = P\bar{s}_1$, $H_2\bar{s}_2 = (P^{-1}H_1P)(P^{-1}\bar{s}_1) = (P^{-1}H_1\bar{s}_1)$. The above result indicates that given a nonuniform signal, if we convert it into a uniform signal, apply the filter in uniform domain and then reconvert into non-uniform signal is the same process that can be achieved by applying a filter on non-uniform signal directly. Following example illustrates the procedure in the context of denoising.

Example 2. In this example, we use low-pass filtering as a means to remove high frequency noise from a nonuniform signal. We generate a nonuniform sig-

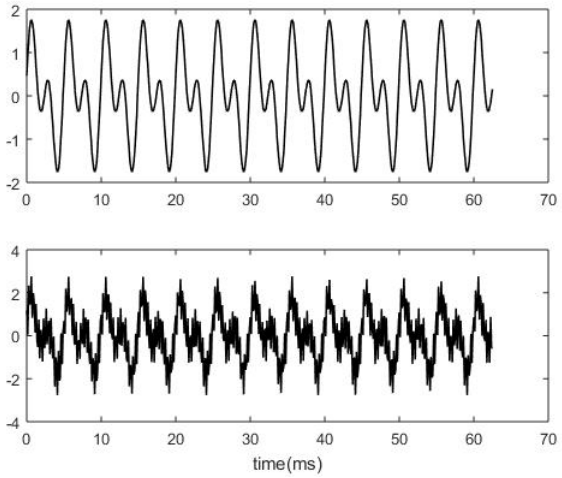

Figure 5.6: See Example 2. (Top) Original Signal, (Bottom) Noisy Signal. Signals are nonuniformly sampled

nal using combination of sine waves and add high frequency noise to the same. The original signal (i.e. without high frequency noise) and the noisy signal are shown in Figure 5.6. The average sample-rate of nonuniform signal is 8 kHz. The denoised signals with two different approaches (i.e. one with the proposed isospectral approach and one with the distance based adjacency matrix as reference) are shown in Figure 5.7. The efficacy of the proposed approach is noticeable in Figure 5.7.

In general, considering the same signal space being sampled on multiple equivalent (i.e. signal can be reconstructed without any error from one grid to another) non-uniform grids with respective adjacency matrices as $A_1, A_2, \cdots A_k$, then these matrices are all similar and all the respective graphs are isospectral. Not only that, a particular signal is mapped to identical polynomials (irrespective of sampling-grid) via the adjacency matrix. This is pictorially represented in Figure 5.8.

GSP-isomorphism is not limited to the Discrete Fourier Transform. The principles established so far can be applied to any graph. We now apply GSP-isomorphism to Discrete Cosine Transform(DCT), which is often used in data compression. We demonstrate the use of GSP-isomorphism in the case of compression of DCT-compressible signals (i.e. signal that can be efficiently compressed using the DCT)

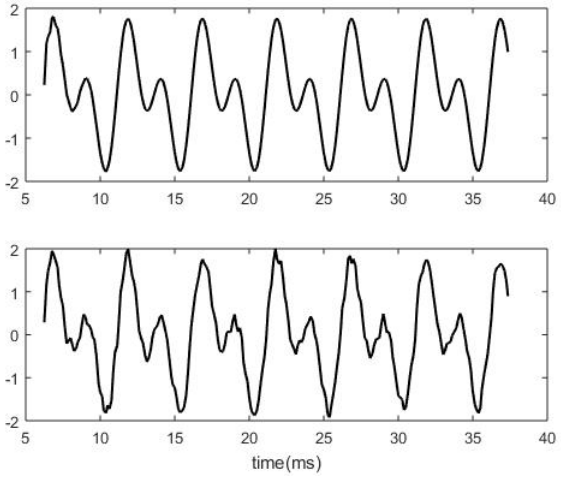

Figure 5.7: See Example 2.Filtered signals (Top) Using proposed adjacency matrix (Bottom) Using reference adjacency matrix.

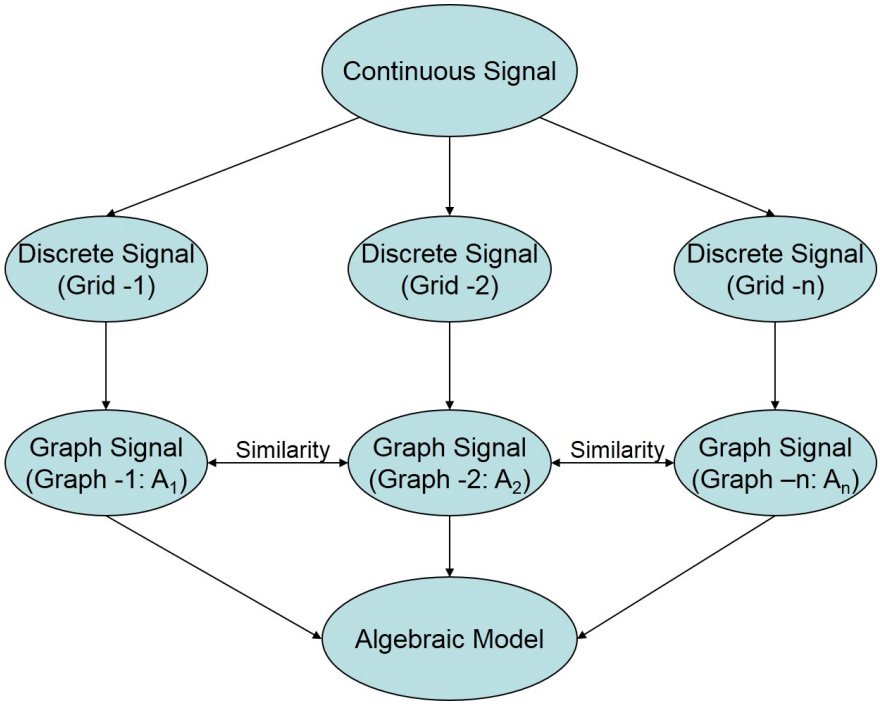

Figure 5.8: Illustration of sampling grids and their algebraic relations via isospectral graphs. The matrices are similar which give rise to identical spectral domains and identical representation of signal in algebraic domain.

in the context of nonuniform sampling, in the following example.

Example 3. In this example, we use the formula $V_2^{-1} = V_1^{-1}\beta_1\beta_2^{-1}$, where V_1^{-1} is the standard DCT, in order to construct the nonuniform DCT matrix denoted byV_2^{-1}. Assuming that the signal space is spanned by appropriate cosines (DCT Type-II) of various frequencies, we can find β_1 and β_2, by uniform and nonuniform sampling of the basis, respectively. We create a DCT-compressible (DCT bandlimited) signal and sample the same at nonuniform intervals. We then use the standard DCT and nonuniform DCT in order to compress the signal using various compression ratios, i.e., the number of DCT coefficient used to the total number of DCT coefficients. It can be seen from Figure 5.9 that the derived nonuniform DCT provides better compression compared to the standard DCT.

5.3 GSP-Homomorphism

In this section, we describe the GSP-homomorphism, the conditions for homomorphism, and application of the same in graph downsampling. Consider two graphs $G_A = (\mathcal{V}_A, A)$ and $G_B = (\mathcal{V}_B, B)$ with their GSPFs (S_A, F_A) and (S_B, F_B) respectively. Let $\Phi = (\Phi_S, \Phi_F)$ be a GSP-homomorphism between the two GSPFs. Let the number of vertices in G_A and G_B be n and m respectively, with $n > m$. The linear map Φ_S, which maps signal $\bar{s}_A$ on graph G_A to a signal $\bar{s}_B$ on graph G_B, is represented by matrix M, i.e. $\bar{s}_B = M\bar{s}_A$. It is evident that matrix M is a rectangular $m \times n$ matrix. We assume that the matrix has rank m, so it has a right-inverse, denoted by M^+, such that $MM^+ = I_{m\times m}$. As we focus on LSI filters, the filters are polynomials in A and B for respective graphs. The GSP-homomorphism property is pictorially explained in Figure 5.10.

We need to obtain the conditions on M for the homomorphism property to hold in light of the fact that filters on both graphs are polynomials in adjacency matrices. We also need to establish relation between A and B given $\bar{s}_B = M\bar{s}_A$. Thus, the matrix M plays the role of linear map Φ_S in this case. We need to find the map Φ_F in order to describe the GSP-homomorphism completely. There can be multiple GSP-homomorphisms for given pair of GSPFs, out of which we select

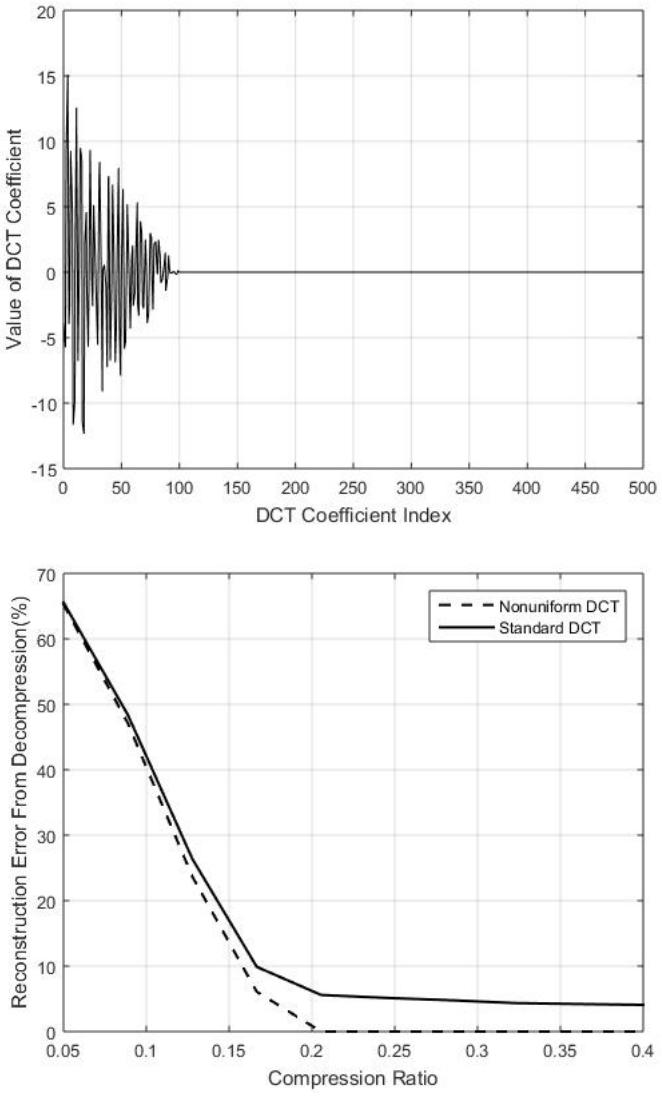

Figure 5.9: Refer Example 3. (Top) The DCT profile of the signal undergoing compression at various scales. (Bottom) Reconstruction Errors using various compression ratios by two different approaches.

$$\begin{array}{ccc} (A, \bar{s}_A) & \xrightarrow{Shift\ A} & A\bar{s}_A \\ {\scriptstyle M}\downarrow & & \downarrow{\scriptstyle M} \\ (B, \bar{s}_B) & \xrightarrow{Shift\ B} & B\bar{s}_B \end{array}$$

Figure 5.10: The commutative diagram for GSP-homomorphism: Shifting

$$\begin{array}{ccc} (h(A), \bar{s}_A) & \xrightarrow{filtering} & h(A)\bar{s}_A \\ {\scriptstyle M}\downarrow & & \downarrow{\scriptstyle M} \\ (h(B), \bar{s}_B) & \xrightarrow{filtering} & h(B)\bar{s}_B \end{array}$$

Figure 5.11: The commutative diagram for GSP-homomorphism: Filtering

one in which the GSP-homomorphism (Φ_F) maps the generator A of F_A to the generator B of F_B. Thus

$$\Phi_F(A) = B$$

Using Theorem 5.1, $\forall f \in F_A, \Phi_F(f)\Phi_S = \Phi_S f$. For $f = A$, we get $MA = BM$. This condition is represented using a commutative diagram in Figure 5.10.

Theorem 5.8 *Consider graphs $G_A = (\mathcal{V}_A, A)$ and $G_B = (\mathcal{V}_B, B)$ with GSPFs (S_A, F_A) and (S_B, F_B), where S_A and S_B are sets of signals on graphs G_A and G_B respectively, and F_A and F_B are set of LSI filters on graphs G_A and G_B respectively. Let $\Phi = (\Phi_S, \Phi_F)$ represent a GSP-homomorphism with $\Phi_F(A) = B$. Then, $\forall i \in \mathbb{N}, MA^i = B^i M$*

Proof In Theorem 5.4, let $f_1 = f_2 = A$ and $\Phi_S = M$, then

$$\Phi_F(A)\Phi_F(A)M = M(A)(A) \Rightarrow MA^2 = B^2M$$

Successively applying this formula proves $MA^i = B^iM$ for an arbitrary $i \in \mathbb{N}$. ∎

Now, let $h(A)$ be a filter in graph G_A, then using Theorem 5.8 and the fact that M is linear, gives us $Mh(A) = h(B)M$. This result is pictorially represented in Figure 5.11.

Theorem 5.8 shows that the GSP-homomorphism puts conditions on matrix M and also matrices A and B. The following theorem (Theorem 5.9) explores a condition that guarantees the existence of a GSP-homomorphism.

Theorem 5.9 *Consider graphs $G_A = (\mathcal{V}_A, A)$ and $G_B = (\mathcal{V}_B, B)$. If eigenvalues of B form a subset of eigenvalues of A, then there exists a GSP-homomorphism between two respective GSPFs.*

Proof We continue with notations used in Theorem 5.8. Here, relation between A and B is provided and we need to show GSP-homomorphism. Let $A = V_A \Lambda_A V_A^T$ and $B = V_B \Lambda_B V_B^T$ be the diagonalization of A and B respectively. As eigenvalues of B form a subset of eigenvalues of A, we can express Λ_A in form of Λ_B with proper ordering as $\Lambda_A = diag(\Lambda_B, \Lambda'_A)$.

Now, assign $M = V_B P V_A^T$, where $P = [I, 0]$, with I being $m \times m$ identity matrix and P being $m \times n$ matrix. For an arbitrary $i \in \mathbb{N}$, $MA^i = V_B P V_A^T V_A \Lambda_A^i V_A^T = V_B [\Lambda_B^i, 0] V_A^T$ and $B^i M = V_B \Lambda_B^i V_B^T V_B P V_A^T = V_B [\Lambda_B^i, 0] V_A^T$. Thus $MA^i = B^i M$. Using Theorem 5.8, we obtain the required result. ∎

Now, we look at instances of GSP-homomorphism.

5.3.1 GSP-homomorphism Between a Graph and its Connected Component

Consider a graph which has two connected components (i.e. the graph is disconnected). With appropriate ordering and without the loss of generality, the graph adjacency matrix can be written as

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}.$$

where A is the graph adjacency matrix, where A_1 and A_2 are adjacency matrices corresponding to the connected components.

Theorem 5.10 *There exists a GSP-homomorphism between a graph and its connected component.*

Proof We provide proof for a graph with exactly two connected components. The proof can then be generalized for any number of connected components.
Here,

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$$

Thus, for an arbitrary polynomial h,

$$h(A) = \begin{bmatrix} h(A_1) & 0 \\ 0 & h(A_2) \end{bmatrix}$$

Consider a selection matrix S that maps a signal on original graph $\bar{s}$ to a signal on one of the connected components $\bar{s}_1$, e.g. component corresponding to A_1. Let $\bar{s}_2$ be the signal corresponding to A_2. Let $h(A) \mapsto h(A_1)$ and $\bar{s} \mapsto \bar{s}_1$. Now, $Sh(A)\bar{s}$

$$= S \begin{bmatrix} h(A_1) & 0 \\ 0 & h(A_2) \end{bmatrix} \bar{s} = [h(A_1) \;\; 0]\bar{s} = h(A_1)\bar{s}_1$$

Hence $h(A)\bar{s} \mapsto h(A_1)\bar{s}_1$, which provides the GSP-homomorphism. ∎

Using Theorem 5.10, we can define a different type of GSP-homomorphism, termed as *Composite GSP-homomorphism*.

Definition Given a graph $G = (\mathcal{V}, A)$, if there exist anther graph $G_1 = (\mathcal{V}_1, A_1)$ and a polynomial r such that $r(A)$, with appropriate ordering of vertices, can be written as $\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$, then G is said to have a composite GSP-homomorphism with G_1.

A prominent example of composite GSP-homomorphism is provided by graphs representing classical DSP grids. Figure 5.12 provides one such instance.

5.3.2 GSP-homomorphism: Invariant Subspaces

Consider an $N-$node graph $G = (\mathcal{V}, A)$ with GSPF (S, F). Let S_1 be an invariant subspace (under the set of shift invariant filters F) of vector space S. In such a

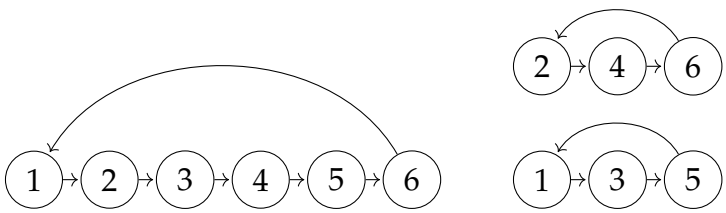

Figure 5.12: Illustration of composite GSP-homomorphism for (Left) 6-node classical DSP graph and (Right) graph obtained by squaring the adjacency matrix.

scenario, there exists an invertible matrix P such that it block diagonalizes matrix A via a similarity transform. Thus PAP^{-1} can be written as

$$PAP^{-1} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$$

Let $\tilde{G}$ and G_1 be the graphs described by adjacency matrices $\tilde{A} = PAP^{-1}$ and A_1, respectively. Using Theorem 5.10, $\tilde{G}$ has a GSP-homomorphism with G_1. Let (S_1, F_1) be the GSPF associated with G_1. Thus $\Phi = (\Phi_S, \Phi_F)$, where Φ_S defines the projection from S to S_1 and Φ_F maps a filter $h(A)$ in GSPF (S, F) to a filter $h(A_1)$ in GSPF (S_1, F_1). Let us denote the graph corresponding to adjacency matrix A_1 as G_1. As the graphs corresponding to A and PAP^{-1} are isospectral by definition, they also share a GSP-isomorphism (which is a special case of GSP-homomorphism). Moreover, the graph corresponding to PAP^{-1} shares a GSP-homomorphism with the graph corresponding to A_1. Thus, using Theorem 5.4, we conclude that the graph G shares a GSP-homomorphism with the graph G_1.

Let us assume that the adjacency matrix A has eigenvalues $\{\lambda_1, \cdots, \lambda_N\}$ with corresponding eigenvectors as $\{e_1, \cdots, e_N\}$. Let $S_R = span\{e_1, \cdots, e_{N_1}\}$, where $N_1 < N, N_1 \in \mathbb{N}$. Let $S_1 = \{\Pi P s \mid \forall s \in S\}$ (where $\Pi = [I_{N_1}\ 0]_{N_1 \times N}$) and F_1 be the set of polynomials in A_1. The map P is such that eigenvalues of A_1 are $\lambda_1, \cdots, \lambda_{N_1}$. From above discussion, we know that there exists a GSP-homomorphism between GSPFs (S_R, F) and (S_1, F_1). However, the map $\Phi_S : S_R \to S_1$ is invertible, thus there exists a GSP-isomorphism between (S_R, F_R) and (S_1, F_1), where $F_R = \{h(A) \mid h(A)e_i = 0, i = N_1 + 1, \cdots N\}$.

5.4 GSP-Isomorhisms and GSP-homomorphisms in Downsampling of Bipartite Graphs

In this section, we analyze the bipartite graphs in the light of GSP-isomorphism and GSP-homomorphism discussed in previous sections. We know that bipartite graphs are closely coupled with the process of downsampling [22, 40]. We take a look at the relations between a bipartite graph and its downsampled version from the perspective of GSP-homomorphism and GSP-isomorphism. To maintain simplicity, we focus on a bipartite graph with identical number of nodes in both the bipartite partitions.

With appropriate ordering, the adjacency matrix for a bipartite graph can be expressed as $A = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix}$. It is trivial to see that $A^2 = \begin{bmatrix} BB^T & 0 \\ 0 & B^TB \end{bmatrix}$. As explained in [37, 40], the designated adjacency matrix for a downsampled graph on the bipartite partitions are BB^T and B^TB. Thus, a bipartite graph exhibits the property of composite GSP-homomorphism with $r(A) = A^2$. The whole downsampling process and the assignment of adjacency matrices can be re-looked in the light of the concepts of GSP-homomorphism and GSP-isomorphism.

We can see that the matrix A^2 represents a graph with at-least two connected components. From Theorem 5.10, this graph has a GSP-homomorphism with its connected components. Thus a bipartite graph exhibits GSP-homomorphism with the downsampled graph on a bipartite partition. At the same time, we can also see the two downsampled graphs with adjacency matrices BB^T and B^TB. It is trivial to see that both are isospectral and hence they share a GSP-isomorphism. This validates the observation that graphs sampled on the same space are isospectral.

CHAPTER 6

Conclusion

The research carried out for the thesis helped us understand the nature of graph signals and it also illuminated the fact that the classical signal processing framework can be visualized as a special case of graph signal processing. Thus, we can take inspiration from classical signal processing in order to analyze signal processing on graphs and we can use graph signal processing framework to obtain some new insights for classical signal processing. Traditionally, discrete time signal processing is studied using sampling of continuous time signals. With graph signal processing framework, the discrete time signal processing can be analyzed without using the continuous time signal processing. This also allows us to accommodate various boundary conditions through different adjacency matrices, which results in different spectral transforms (e.g. Discrete Fourier Transform, Discrete Cosine Transform) as Graph Fourier Transforms. Thus, different spectral transforms can also be analyzed using the graph signal processing framework.

In this thesis, we explored the downsampling of graphs from algebraic perspective. We analyzed the bipartite graphs in the context of downsampling and obtained the algebraic model for the same. We obtained a measure (named SDQM) to quantify the quality of a downsampling scheme and provided a greedy algorithm using the same in order to downsample an arbitrary graph. Using the proposed approach, we analyzed the downsampling for Discrete Cosine Transform. We further analyzed the downsampling process for sensitivity towards different frequency bands and provided algorithm to obtain a downsampling scheme with desired frequency sensitivity. We also explored the structure preserving maps between two graphs and mapped nonuniform signal processing as a special case of

graph signal processing using an appropriate assignment of adjacency matrix.

One major limitation of the proposed downsampling approach is its high computational complexity. One research direction could be to examine the SDQM for special class of graphs, and derive computationally efficient algorithms for downsampling of such graphs. Another research direction could be to use a hybrid approach involving graph segmentation and downsampling together in order to downsample large graphs. An open problem is the assignment of adjacency matrix to the downsampled set of vertices. In the thesis, we have indicated the relation between adjacency matrix of a graph with that of a downsampled graph. This can be explored further to find out adjacency matrix for downsampled graphs based on preservation of algebraic properties. It would be interesting to study the downsampling process for different reconstruction methods.

References

[1] National centers for environmental information NOAA: Temperature data, 2014.

[2] A. Anis, A. Gadde, and A. Ortega. Towards a sampling theorem for signals on arbitrary graphs. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3864–3868, May 2014.

[3] A. Anis, A. Gadde, and A. Ortega. Efficient sampling set selection for bandlimited graph signals using graph spectral proxies. *IEEE Trans. Signal Processing*, 64(14):3775–3789, 2016.

[4] T. Bıyıkoğlu, J. Leydold, and P. F. Stadler. *Laplacian eigenvectors of graphs: Perron-Frobenius and Faber-Krahn type theorems*. Springer Verlag, 2007.

[5] G. Chartrand. *Introductory Graph Theory*. Dover, New York, 1985.

[6] S. Chen, A. Sandryhaila, and J. Kovačević. Sampling theory for graph signals. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3392–3396, April 2015.

[7] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovacevic. Signal denoising on graphs via graph filtering. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, December 2014.

[8] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic. Discrete signal processing on graphs: Sampling theory. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, Vol. 63(NO. 24), December 2015.

[9] F. R. K. Chung. *Spectral Graph Theory*. AMS, 1996.

[10] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19:297Ű301, 1965.

[11] F. Dörfler and F. Bullo. Kron reduction of graphs with applications to electrical networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(1):150–163, Jan. 2013.

[12] D. Gfeller and P. De Los Rios. Spectral coarse graining of complex networks. *Phys. Rev. Lett.*, 99:038701, Jul 2007.

[13] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[14] G. R. Grimmett. *Percolation*. Springer, 1997.

[15] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *J. Appl. Comp. Harm. Anal*, 30(2):129Ű150, 2011.

[16] P. Hell and J. NeŽetril. *Graphs and Homomorphisms, Oxford Lecture Series in Mathematics and Its Applications*. Oxford University Press, 2004.

[17] I.N.Herstein. *Topics In Algebra*. John Wiley & Sons, 2007.

[18] P. Lancaster and M. Tismenetsky. *The Theory of Matrices*. Academic Press, 2nd edition, 1985.

[19] P. Liu, X. Wang, and Y. Gu. Coarsening graph signal with spectral invariance. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1070–1074, May 2014.

[20] T.-T. Lu and S.-H. Shiou. Inverses of 2 x 2 block matrices. *Computers & Mathematics with Applications*, 43(1):119 – 129, 2002.

[21] B. D. McKay and A. Piperno. Practical graph isomorphism, II. *CoRR*, abs/1301.1493, 2013.

[22] S. K. Narang, Ortega, and Antonio. Perfect reconstruction two-channel wavelet filter banks for graph structured data. *IEEE Transactions on Signal Processing*, 60(6):2786–2799, 2012.

[23] S. K. Narang and A. Ortega. Downsampling graphs using spectral theory. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4208–4211, May 2011.

[24] H. Q. Nguyen and M. N. Do. Downsampling of signals on graphs via maximum spanning trees. *IEEE Transactions on Signal Processing*, 63(1):182–191, 2015.

[25] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Pearson Education India, 2006.

[26] A. Pothen, H. D. Simon, and K.-P. P. Liu. Partitioning sparse matrices with eigenvectors of graphs. *Report, NAS Systems Division, NASA Ames Research Center*, 1989.

[27] M. Püschel and J. M. F. Moura. Algebraic signal processing theory: 1-D space. *IEEE Transactions on Signal Processing*, 56(8):3586–3599, 2008.

[28] M. Püschel and J. M. F. Moura. Algebraic signal processing theory: Cooley-Tukey type algorithms for DCTs and DSTs. *IEEE Transactions on Signal Processing*, 56(4):1502–1521, 2008.

[29] M. Püschel and J. M. F. Moura. Algebraic signal processing theory: Foundation and 1-D time. *IEEE Transactions on Signal Processing*, 56(8):3572–3585, 2008.

[30] M. Püschel and M. Rötteler. Cooley-Tukey FFT like algorithm for the discrete triangle transform. In *IEEE Digital Signal Processing Workshop*, pages 158–162, 2004.

[31] A. Sandryhaila, J. Kovacevic, and M. Püschel. Algebraic signal processing theory: Cooley-Tukey type algorithms for polynomial transforms based on induction. *SIAM Journal on Matrix Analysis and Applications*, 32(2):364–384, 2011.

[32] A. Sandryhaila and J. M. F. Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61:1644–1656, 2013.

[33] A. Sandryhaila and J. M. F. Moura. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Processing Magazine*, 31(5):80–90, Sept 2014.

[34] A. Sandryhaila and J. M. F. Moura. Discrete signal processing on graphs: Frequency analysis. *IEEE Transactions on Signal Processing*, 62(12):3042–3054, 2014.

[35] A. Schwenk and F. Harary. *Almost all trees are cospectral, In: New Directions in the Theory of Graphs, pp. 275-307,*. Academic Press, New York, 1973.

[36] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs. *IEEE Signal Processing Magazine*, pages 83–98, May 2013.

[37] O. Teke and P. P. Vaidyanathan. Extending classical multirate signal processing theory to graphs - part I: fundamentals. *IEEE Trans. Signal Processing*, 65(2):409–422, 2017.

[38] O. Teke and P. P. Vaidyanathan. Extending classical multirate signal processing theory to graphs - part II: m-channel filter banks. *IEEE Trans. Signal Processing*, 65(2):423–437, 2017.

[39] J. A. Tropp. Column subset selection, matrix factorization, and eigenvalue optimization. In C. Mathieu, editor, *SODA*, pages 978–986. SIAM, 2009.

[40] N. Vaishnav and A. Tatu. Downsampling on bipartite graphs: An algebraic perspective. In *5th IEEE Global Conference on Signal and Information Processing*, 2017.

[41] N. Vaishnav and A. Tatu. To appear: Analysis of downsampling of DCT-graphs. In *International Conference on Signal Processing and Communications (SPCOM)*, 2018.

[42] Y. Voronenko and M. Püschel. Algebraic signal processing theory: Cooley-Tukey type algorithms for real DFTs. *IEEE Transactions on Signal Processing*, 57(1):205–222, 2009.

[43] C. Zhang, D. Florencio, and P. A. Chou. Graph signal processing - a probabilistic framework. Technical report, Microsoft Research, April 2015.

[44] F. Zhang. *The Schur Complement and Its Applications*. Springer, 2005.