International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:11, No:3, 2017

Efficient Filtering of Graph Based Data Using Graph Partitioning

Nileshkumar Vaishnav, Aditya Tatu

***Abstract*—**An algebraic framework for processing graph signals axiomatically designates the graph adjacency matrix as the shift operator. In this setup, we often encounter a problem wherein we know the filtered output and the filter coefficients, and need to find out the input graph signal. Solution to this problem using direct approach requires $\mathcal{O}(N^3)$ operations, where N is the number of vertices in graph. In this paper, we adapt the *spectral graph partitioning* method for partitioning of graphs and use it to reduce the computational cost of the filtering problem. We use the example of denoising of the temperature data to illustrate the efficacy of the approach.

***Keywords*—**Graph signal processing, graph partitioning, inverse filtering on graphs, algebraic signal processing.

I. Introduction

PROCESSING of data such as weather data, seismic activity data, sensor networks data, social network data, transportation data requires the data to be represented in form of signals on graphs. Given the large scope of applications, analysis and processing of signals on graph is important and it has become an emerging field of research in discrete signal processing domain [1]. Signals on graph contrast itself from the signals taken from uniform sampling schemes (e.g. speech/audio signals sampled at uniform sample-rate, image with pixels placed in uniform Cartesian grid,) in that the signals on graph usually come from a nonuniform grid. In graph signals, there is no natural ordering of the signal values, rather the inter-relations between vertices are important, which are captured in the graph adjacency matrix. Defining concepts such as shift, Fourier transform and convolution for such signals is not trivial and diverges greatly from similar concepts defined for uniform signals.

Formally, a graph is a collection of vertices with a given relation structure between the vertices. A graph G is denoted as $(\mathcal{V}, A)$, where $\mathcal{V}$ is the set of vertices $\{v_1, ..., v_N\}$ and A is the graph adjacency matrix which provides the relation structure between the set of vertices. For matrix A, each element $a_{i,j}$ is the weight connecting vertex v_j to vertex v_i. For an unweighted graph, the adjacency matrix A has binary entries. For an undirected graph, A is symmetric. Traditionally, spectral properties of graphs with symmetric adjacency matrices are derived using *graph Laplacian*. Graph Laplacian is defined as $L = D - A$, where D is a diagonal matrix with $d_{i,i}$ being the sum of edge-weights connecting vertex v_i. A recent approach [2] indicates that the spectral analysis of graph signals can also be carried out effectively using the graph adjacency matrix. This allows us to work with signals on directed graphs, which is not possible with Graph Laplaician based approach. It is pertinent to note here that the early research on graph and graph Laplacian focused on deriving topological properties of graphs such as connectedness and the number of connected components. The present focus of research is on how the properties of graph affects the graph signals and how to effectively process the same.

In graph Laplacian based signal processing, the study of eigenvalues and eigenvectors of graph Laplacian is called *Spectral Graph Theory* [3]. The graph Fourier transform is defined by the eigenvector matrix of graph-Laplacian. This is derived using analogy from 1-D signals, where the Fourier transform is an expansion of a signal in terms of eigenvectors of 1-D Laplacian. Hammond [4] defines wavelet transform for graph signals using the aforementioned graph Fourier transform. Convolution, filtering, translation and modulation of graph signals are defined using the graph Fourier transform. A limitation of the graph-Laplacian based approach is that it works only if the graph is undirected.

In another line of work, algebraic signal processing theory for LSI systems is developed by Puschel [5], [6]. Sandryhaila extends the work of ASP for graph based signal processing and introduced the algebraic model for graph signals and filters in [2], [7]–[9]. In this model, both the graph filters and graph signals are mapped to polynomials while the adjacency matrix plays the role of the shift operator. Using this model, concepts of graph Fourier transform, convolution, filtering, modulation and translation are defined on polynomial mappings of the graph signals and filters. An advantage of graph signal processing based on adjacency matrix is that the graph need not be undirected.

In this paper, we explain the inverse filtering problem in the framework of algebraic graph signal processing. We also discuss some existing methods to solve the problem, their advantages and limitations. We propose that graph partitioning is a viable solution to remove the limitations posed by other methods. We adapt the spectral graph partitioning method to achieve partitioning of directed graphs and apply the same for denoising of temperature data.

II. Algebraic Model for Graphs

This section presents some important results related to algebraic approach to graphs and defines a signal model for the same. Detailed discussion on the results along with proofs can be found in [2], [9].

Nileshkumar Vaishnav is with the Information and Communication Technology Department, DA-IICT, Gandhinagar, India (e-mail: 201121007@daiict.ac.in).

Let $G = (\mathcal{V}, A)$ be the given graph, where $\mathcal{V}$ is the ordered set of vertices and A is the related adjacency matrix. Let the vertices be denoted as $v_i, i = 1, ..., N$. Let the value at vertex v_i be s_i. Then the graph signal $\bar{s}$ is given by,

$$\bar{s} = (s_1, s_2, ..., s_N)^T \in \mathbb{C}^N$$

A matrix $H \in \mathbb{C}^{N \times N}$ represents a linear transform (or filter) that operates on a given graph signal. The adjacency matrix A is designated as the shift operator. A graph filter H is called *shift-invariant* iff H commutes with A. Proposition 12.4.1 in [10] provides an important result related to commutative matrices. The result is reproduced here as Proposition 1.

Proposition 1. Given $N \times N$ matrices H and A 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 $N - 1$.

A. Algebraic Model

We assume that the matrix A has a set of N distinct eigenvalues $\lambda_0, ...\lambda_{N-1}$; which implies that its characteristic and minimal polynomials are identical.[1] The algebraic model for signal processing is denoted by $(\mathcal{A}, \mathcal{M}, \Phi)$, where $\mathcal{A}$ is the algebra of filters, $\mathcal{M}$ is $\mathcal{A}$-module of signals and Φ is an isomorphic transform which generalizes the $z - transform$.

Fourier Transform: Given the Jordan Normal Form of A to be VJV^{-1}, the matrix V^{-1} defines graph Fourier transform. The spectrum of the signal $\bar{s}$ is computed as $V^{-1}\bar{s}$. Since the eigenvalues of A are assumed to be distinct, the spectrum can also be given by $(s(\lambda_0), ..., s(\lambda_{N-1}))$.

Filters: Let $\mathcal{A} = \mathbb{C}[x]/p_A(x)$, where $\mathbb{C}[x]/p_A(x)$ is the set of polynomials in x with multiplication defined as modulo-$p_A(x)$ and $p_A(x)$ is the characteristic polynomial of A. Let $\mathcal{F}$ be the space of filters. Using Proposition 1, a shift-invariant filter H can be represented as a polynomial in A, i.e. $H = h(A)$, where $h(x)$ is a polynomial of degree at most $N - 1$. If $A \mapsto x$, then $H = h(A) \mapsto h(x)$. In general, a filter with L taps $(h_0, ...h_{L-1})$ can be written as

$$H = h_0 A^0 + h_1 A^1 + ... + h_{L-1} A^{L-1}$$

Signals: 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 $s = (s_0, ..., s_{N-1}) \mapsto s(x)$. The isomorphism that maps the signal space S to the module $\mathcal{M}$ is called the *graph z-transform*. With proper ordering of eigenvalues and eigenvectors, following equation holds true; which can be used to compute the coefficients of $s(x)$.

$$(s(\lambda_i))_{i=0}^{N-1} = V^{-1}\bar{s}$$

Filtering: 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,

$$\tilde{s}(x) = h(x)s(x) \ mod \ p_A(x)$$

[1]The algebraic model described in the paper is applicable even if A has repeated eigenvalues. The assumption is made solely to simplify the discussion of key concepts.

Frequency Ordering: Frequency ordering is defined using the *total variation* (TV) of graph signals, denoted by TV_G, and defined as $TV_G(s) = \|s - A_{norm}s\|_1$, 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. Normalization of A avoids the attenuation/amplification of the signal while shifting. Larger value of $TV_G(s)$ indicates presence of higher frequency content in the signal. Frequency ordering provides the framework to define concepts such as low and high frequencies and as a consequence, filters with desired response can be designed. For details on filter design, refer [8].

III. Inverse Filtering Problem

Consider a graph $G = (\mathcal{V}, A)$, where $\mathcal{V}$ is the set of vertices with $|\mathcal{V}| = N$ and A is the adjacency matrix. Assume that we know the output graph signal $\bar{b}$ of a particular filter H and our aim is to find the input signal $\bar{x}$. As we assume that the filter H is LSI (with respect to shift A), we can write $H = h(A)$. Hence, the problem of inverse filtering is stated as below.

$$h(A)\bar{x} = \bar{b}$$

We assume that we know the filter h by taps. Solving the above system of linear equations using a direct approach has a computational cost of $\mathcal{O}(N^3)$. This cost is practically prohibitive for large values of N. However, in many cases the matrix A is sparse and this sparsity can be used to minimize the computation cost.

A. The Cost Function

In many cases, a set of signals are required to be filtered by a set of filters. Let's call such a processing requirement as *batch processing*. In solving the inverse filter for a set of filters in batch processing mode, the computational cost comes from three major tasks. The first task is processing of the adjacency matrix (e.g. diagonalization, solving for eigenvalues, triangulation). The second task involves processing the filter $h(A)$ in a form that is suitable for efficient application of filter. The third and final task involves applying the filter on a particular instance of the graph signal. We denote the costs as $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ respectively.

Consider the example of filtering temperature data based on a geodesic adjacency matrix. Let's assume that the temperature values are corrupted by noise and our aim is to recover the original signals. In such a case, the adjacency matrix is identical for processing all instances of signals. If we need to denoise m number of signals by k number of filters for a given adjacency matrix, then the total cost $\mathcal{C}$ can be expressed as

$$\mathcal{C} = \mathcal{C}_1 + k\mathcal{C}_2 + mk\mathcal{C}_3$$

IV. Present Approaches and Their Computational Costs

In this section, we overview some existing approaches to solve the batch filtering problem. We also list down their advantages and limitations that would explain which of the methods is more suitable for batch processing given the values of k and m.

A. Inverse Filtering in Fourier Domain (JNF)

In this approach, the Jordan Normal Form (JNF) of the adjacency matrix is computed first. Let $A = VJV^{-1}$, then a filter $h(A)$ can then be computed as $h(J)$ in the Fourier domain. Similarly a signal $\bar{b}$ must be first converted into Fourier domain as $V^{-1}\bar{b}$.

$$h(A)\bar{x} = \bar{b}$$

$$\Rightarrow h(J)V^{-1}\bar{x} = V^{-1}\bar{b}$$

The computational costs associated with this approach are $\mathcal{O}(N^2)$ for both $\mathcal{C}_2$ and $\mathcal{C}_3$. However, computing the JNF of any matrix is not a numerically stable operation. If the matrix is diagonalizable, such an approach would still outperform any other approach presented here for non-sparse matrices. However, this method does not exploit the sparsity (if present) of A, rather it produces a sparse matrix J which in turn gives the computational advantage.

B. Inverse Filtering in Schur Form

As computing the JNF of a given adjacency matrix is numerically unstable, the next best option is to triangularise the filter. Such an operation can be achieved by using Schur form of adjacency matrix A, given by $A = UTU^*$, where U is a unitary matrix and T is an upper triangular matrix.

By proceeding in the similar fashion as done in JNF, it can be concluded that the matrix U^* can be used to put any $h(A)$ into a triangular form and then the equation can be solved using the process of *back substitution*, solving an instance of signal filtering in $\mathcal{O}(N^2)$.

C. Converting the Problem into Forward Filtering

In this method, instead of directly solving the inverse filter, the problem is first converted into a forward filtering one [11]. If we can find a filter $g(A)$ such that $h(A)g(A) = I$, then $\bar{x} = g(A)\bar{b}$ can be solved in $\mathcal{O}(KLN)$ for K-sparse matrix A and L order filter $g(A)$. However, this approach requires us to compute filter $g(A)$ by evaluating the inverse filter response at N points. Such an operation requires $\mathcal{O}(N^3)$ computations in the worst case scenario. To avoid this situation, the filter-order L is fixed and then the $g(A)$ is found using least-squares approach. However, such an approach is not guaranteed to result in the desired filter response, while still requiring $\mathcal{O}(L^2N)$ operations.

While the methods described above may be useful for various filtering problems, all the three methods suffer from a severe drawback. They all require the knowledge of eigenvalues of matrix A, either directly or indirectly. We know that solving eigenvalue problem requires solving a polynomial of degree N, which is computationally complex and numerically unstable for large N. If the matrix A is sparse, the present professional softwares provide ways to find a few eigenvalues and associated eigenvectors. However, the number of eigenvalues that are provided are minuscule compared to N. In the next section, we look at how we can resolve the issue by partitioning a graph into two nearly equal sized subgraphs.

V. Filtering Based on Graph Partitioning

For large number of vertices, efficient implementation of inverse filtering is impractical, therefore we try to partition the given graph into multiple small partitions and then attempt to process the signal into two subgraphs. There is also an algebraic motivation for partitioning a graph into multiple subgraphs. We know that $A = VJV^{-1}$. For simplicity of discussion, assume that J is a diagonal matrix, then $A^l = VJ^lV^{-1}$. The largest permissible power of A is $N-1$. However, the eigenvalues of A do not have identical norms, and hence J^l is dominated by the largest absolute eigenvalue of A as l grows larger. This means that the higher powers of A are unnecessary for filtering purpose as they don't capture the frequency information at whole spectrum of A. This also indicates that we can do the desired filtering with lower powers of A. Partitioning a set into multiple subgraphs automatically reduces the largest allowable power of a filter, and assists lower order filter design. Another advantage of graph partitioning based approach is that the filter H need not be a shift-invariant filter. We will see later that denoising filter contains terms A and A^* and hence it is not necessarily a shift-invariant filter.

Definition 1: Graph Partition. Given a graph $G = (\mathcal{V}, A)$, where $\mathcal{V}$ is the set of vertices, if $\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_P$ are sets of vertices, where $P \leq |\mathcal{V}| = N$, such that $\mathcal{V}_i \cap \mathcal{V}_j = \phi, \forall i \neq j$ and $\bigcup_{i=1}^{P} = \mathcal{V}$, then $\mathcal{V}_1, ...\mathcal{V}_P$ is called partition on graph G.

Our aim is to divide a directed graph into two almost equal size partition subgraphs. A method called *spectral graph partitioning* presented in [12] provides a way to partition an undirected graph into two almost equal sized partition subgraphs. Given an undirected graph, the eigenvector associated with the second eigenvalue (in ascending order of eigenvalues) of the graph Laplacian is computed. This eigenvector has all real entries as graph Laplacian is a positive semi-definite matrix. Let the eigenvector be e_2, then $e_2 > median(e_2)$ divides the graph into two almost equal sized partitions. We can also use a custom constant instead of $median(e_2)$ in order to achieve desirable partition. However, this method requires modifications to be applicable to directed graphs.

Given a directed graph with large number of vertices, first compute $B = AA^T$, where A^T is the transpose of matrix A. The matrix B is symmetric and hence we can apply the spectral graph partitioning on the same. The eigenvector associated with second eigenvalue of graph Laplacian defined using adjacency matrix B provides the desired partitioning for the directed graph under consideration. Other symmetric forms such as $A+A^T$ or A^TA can also be used which result in small variations in output partitions.

A. Experimental Results: Denoising Temperature Data

Temperature data provides a typical example of graph data. We consider each temperature recording sensor as a vertex of the graph and each temperature value from that sensor as the temperature sample. We use the climate data provided by NOAA (National Oceanic And Atmospheric Administration, US), which is publicly available. We consider 197 such sensors hence $N = 197$. Each graph signal is a vector of length N.

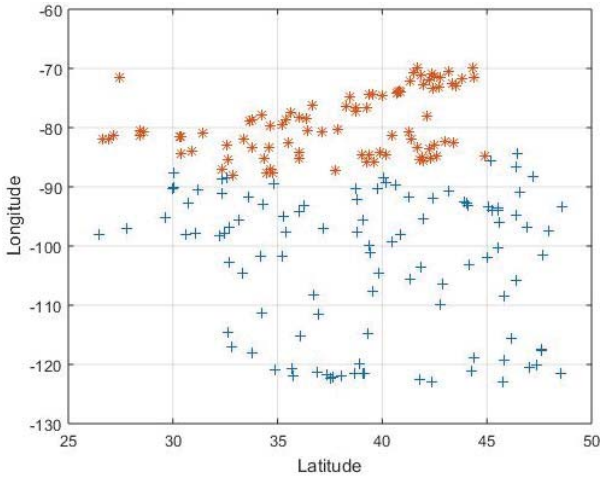

Fig. 1 Partitioning of graph vertices, where '*' and '+' indicate two separate partitions

We consider the US temperature record of a full year (i.e. 365 days) for year 2014. We have kept N as relatively low, so that the unpartitioned filtering can also be computed and be compared with the results of partitioned filtering. We add Gaussian and uniform noise to data and attempt to recover the signal using denoising filter $(I + \alpha(I - A)^*(I - A))^{-1}$, where α is the regularization parameter.[2] The partitioning of the temperature sensor is indicated in Fig. 1.

TABLE I
AVERAGE RMS ERRORS IN DENOISED TEMPERATURE DATA

Noise		Regularization Parameter				
		0.01	0.1	0.5	1	10
Gaussian	U	9.88	8.98	7.06	6.50	12.52
RMS = 10	P	9.87	8.95	7.05	6.58	12.58
Gaussian	U	19.77	18.03	13.89	11.99	14.86
RMS = 20	P	19.77	18.04	13.98	12.10	14.95
Uniform	U	9.89	9.06	7.32	6.91	12.88
RMS = 10	P	9.89	9.07	7.45	7.10	13.00
Uniform	U	19.77	18.00	13.55	11.25	13.03
RMS = 20	P	19.77	17.98	13.54	11.24	12.86

U indicates output for unpartitioned graph, p indicates output for partitioned graph.

TABLE II
STATISTICAL COMPARISON OF RUNTIME (IN SECONDS) FOR FILTERING UNDER UNPARTITIONED AND PARTITIONED GRAPHS FOR 100 TRIALS

	Unpartitioned	Partitioned
Mean	4.05	1.63
Std Deviation	0.3	0.1

From Table I, it can be seen that the denoising based on partitioned filter performs comparatively to the unpartitioned filter. Table II shows the runtimes of both the approaches over a run of 100 trials. From Tables I and II, it can be concluded that there is a significant improvement in runtime with partitioned graphs with negligible change in the output. At the same time, for very large values of N (typically $N > 10000$), partitioning is the only practical way to proceed, as the matrices associated with such large N tend to consume too much memory to be processed without aid of special purpose computers.

[2]The filter is arrived at by minimizing the cost function $\|\bar{x} - \bar{y}\|_2^2 + \alpha\|\bar{x} - A\bar{x}\|_2^2$ with respect to $\bar{x}$. Here, $\bar{y}$ is the noisy signal.

VI. CONCLUSION AND FUTURE WORK

We looked at various approaches to solve the inverse filtering problem for graph signal processing. For processing graph with large number of vertices, we propose a partition based filtering method, the efficacy of which is established using the temperature data denoising. We also demonstrated how symmetric forms such as AA^T can be used to partition a directed graph using spectral graph partitioning method.

REFERENCES

[1] Pascal Frossard Antonio Ortega David I Shuman, Sunil K. Narang and Pierre Vandergheynst, "The emerging field of signal processing on graphs," *IEEE Signal Processing Magazine*, pp. 83–98, May 2013.
[2] Jose M. F. Moura Aliaksei Sandryhaila, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, pp. 1644–1656, 2013.
[3] F. R. K. Chung, *Spectral Graph Theory*, AMS, 1996.
[4] P. Vandergheynst D. K. Hammond and R. Gribonval, "Wavelets on graphs via spectral graph theory," *J. Appl. Comp. Harm. Anal*, vol. 30, no. 2, pp. 129150, 2011.
[5] Markus Püschel and José M. F. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3572–3585, 2008.
[6] Markus Püschel and José M. F. Moura, "Algebraic signal processing theory: 1-D space," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3586–3599, 2008.
[7] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph fourier transform," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 6167-6170*, 2013.
[8] J. M. F. Moura A. Sandryhaila, "Discrete signal processing on graphs: Graph filters," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 6163-6166*, 2013.
[9] J. M. F. Moura A. Sandryhaila, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.
[10] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, Academic Press, 2nd edition, 1985.
[11] Jose M. F. Moura Jelena Kovacevic Siheng Chen, Aliaksei Sandryhaila, "Signal denoising on graphs via graph filtering," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP),*, December 2014.
[12] Kang-Pu Paul Liu Alex Pothen, Horst D. Simon, "Partitioning sparse matrices with eigenvectors of graphs," *Report, NAS Systems Division, NASA Ames Research Center*, 1989.