Computer Life
ISSN: 1819-4818 | Vol. 12, No. 1, 2024

Performance Validation of Jointly Optimized Graph Filters

Bin Cui [1,*]

Henan Polytechnic University, Jiaozuo, 454003, China
[*] **Corresponding author**: Bin Cui (Email: 1570874432@qq.com)

Abstract: In this paper, a novel jointly optimized graph filter is investigated, which is based on the vertex sampling method and the sum-of-squares integration method. It takes the frequency response of the ideal graph filter and the error value of the frequency response of the designed graph filter as the optimization objectives of the jointly optimized graph filter. Firstly, the theoretical analysis and validation of the practical performance of this method are carried out in this paper, and theoretically, we can get that the performance of the jointly optimized graph filter is improved compared with that of the non-jointly optimized graph filter. Then, the actual performance of the jointly optimized graph filter is verified using real data sets. Finally, the jointly optimized graph filter design method is compared with two benchmark graph filter design methods. It is concluded that the performance of the jointly optimized graph filter also has a good improvement over the benchmark graph filters.

Keywords: Graph signal processing, graph filters, joint optimization.

1. Introduction

Due to the complex structure of graph signals, such as genetic networks[1][2], social networks[3][4], sensor networks[5][6], etc., are receiving more and more attention from researchers. The main research of graph signal processing is to process and analyze the modeled graph information. Different from traditional signal processing, graph signal processing needs to consider the spatial structure and topological relationship of the data, and thus has higher complexity and challenges[7].

Graph filtering is a commonly used technique in graph signal processing, and the graph filter is a key tool for implementing graph signal processing[8][9]. It transforms the graph signal values into mathematical matrix information by simulating them and borrows some operations such as Gaussian filtering and median filtering to give better performance to the graph filter. Liu et al[10] proposed two different methods for ARMR graph filter design, the first one is derived from Prony's method[11], and the other is an iterative algorithm[12], which minimizes the original error and the ideal filter error during the iterative process. The method can be adapted to a wide range of graph filters, which also means that graph filters can be given a generalized design method similar to that of digital filters. In general, graph filters can be designed in two ways. The first method is inspired by digital signal processing and uses vertex sampling to design digital filters, which can also be applied to graph filters. The other approach is to design graph filters using square and integral approximation. However, graph filters designed using these two methods have some limitations. Specifically, the frequency response obtained using the first method will have good results at some significant sampling points,but fluctuates considerably between sampling points. The other method focuses mainly on the overall design and ignores the frequency response at specific important points. In order to solve the above problems and to design better graph filters, a design method that combines the vertex sampling method and the sum-of-squares integral approximation method is proposed in this chapter. By combining the optimization equations of the two graph filter design methods, a closed-form solution for the coefficients of the jointly optimized graph filter is obtained. This study also provides closed-form solutions for different graph filter designs, i.e., the vertex sampling method and the sum-of-squares integral approximation method. This joint optimization strategy can be extended to FIR and ARMA graph filters. The designs in this study are validated by simulation and the results show that the graph filters designed using the joint optimization method perform better in terms of frequency response.

2. Preliminary

2.1. Graphs and Graph Signal Processing

A graph is an important data structure used to represent relationships between things. It consists of nodes and edges, where nodes represent entity information and edges represent the connectivity between nodes. In modern engineering and computer science, graphs have become an effective tool for representing entities and their interconnections because they can clearly describe the structure and associations of complex systems. Graph signal processing, a cross-disciplinary field that has emerged in recent years, extends the concepts of traditional signal processing to the structure of graphs, providing researchers with new ways to analyze and process graph-structured data.

In graph signal processing, signals are defined on the nodes of a graph, reflecting some property of the node, such as temperature, voltage, etc. The basic theory of graph signal processing involves graph Laplace matrix, frequency domain analysis, filter design, sampling theorem, etc. The graph Laplace matrix is a central mathematical tool for describing the structural relationships between nodes in a graph and allows for frequency domain analysis. Frequency domain analysis converts the signal to the frequency domain through the Fourier transform of the graph and helps to understand the propagation and evolution of the signal over the graph structure. Graph filters are an important tool for processing graph signals, allowing weighting of different frequency components, which requires consideration of the topology of the graph and the relational weights between nodes. The sampling theorem states the sampling rate conditions required to reconstruct the signal and is important in graph signal processing.

2.2. Graph Shift Operator

In the field of digital signal processing, the basic building block of filter design is the time-shift or delay operation, which implements a single-step delay in time to the input signal such that the i term in the output signal is x_{i-1}, where N represents the signal length. Thus, the time shift can be expressed as $x_{i-1} = x_i$, where x_i is the signal after sampling on the graph. Graph signal processing extends the concept of time displacement to graph signals, in which case the relationship between the data is described by Graph G. I define the operation of displacement on the graph as graph shift, which is accomplished by the weighted sum of the sample value x_v at node v and the sample values of its neighboring nodes:

$$\tilde{x}_v = \sum_{u \in Ni} w_{uv} x_u \tag{1}$$

Where $N(v)$ denotes the set of neighboring nodes of node v and w_{uv} is the connection strength between node u and node v. Thus, the graph shift operator is a matrix describing the weighted connections between nodes, which defines the graph shift operation. $\tilde{x}_i \in R$ represents the samples at node i after graph shift.

In the field of graph signal processing, the graph shift operator is a central concept in graph signal processing applications, and it simulates the counterpart of classical signal processing shift operations on the structural data of a graph. The operator aims to capture topological correlations between nodes in a graph and to enable localized migration of signals at nodes. Typical graph shift operators include graph Laplacian matrices, normalized Laplacian matrices, and adjacency matrices.

The graph Laplacian matrix L is a widely used graph shift operator defined as the degree matrix D minus the adjacency matrix A, i.e., $\mathrm{L} = \mathrm{D} - \mathrm{A}$ .This matrix characterizes the second-order proximity between vertices in the graph and can be viewed as a discrete gradient approximation of the signal on the graph. The normalized Laplacian moment L, on the other hand, is a normalized form of the graph Laplacian matrix, which is normalized by introducing the inverse of the square root of the transition matrix D to obtain $\mathrm{L} = \mathrm{I} - \mathrm{D}^{-\frac{1}{2}} \mathrm{A} \mathrm{D}^{\frac{1}{2}}$, where I denotes the unit matrix. The normalized Laplacian matrix preserves the spectral properties of the original graph while eliminating the pathological problems that may be caused by the graph Laplacian matrix.

2.3. Graph Fourier Transform

The graph Fourier transform is a process that transforms a signal defined on a graph structure from the nodal domain to the frequency domain. The transform is based on the graph Laplace matrix, which is determined by the graph's adjacency and node degree. The adjacency matrix A of the graph records the connections between nodes, while the degree matrix D reflects the number of connections per node. Typically, a set of eigenvectors and corresponding eigenvalues can be obtained by eigendecomposition of the T matrix, which form the basis of the graph Fourier transform. Using these bases, the signal x can be transformed from the nodal domain to the frequency domain:

$$\tilde{x} = \mathrm{U}^T x \tag{2}$$

Where U is the matrix containing all the eigenvectors and $\tilde{x}$ is the signal representation in the frequency domain. The inverse transform of the graph Fourier transform can restore the signal to the nodal domain $x = \mathrm{U}\tilde{x}$. The graph Fourier transform is able to analyze and process the different frequency domain components of a signal in the graph, and is useful for filtering, analyzing, and other processing of signals. Accurate processing of graph signals can be achieved by selectively manipulating specific frequency domain components.

2.4. Graph Filters

Graph filters can be used for smoothing, noise reduction, feature enhancement, and other operations on graph signals, similar to filtering operations in traditional signal processing. This is useful for analyzing and understanding local and global structures in graph data. In social networks, graph filters can help identify community structures, key nodes, and information propagation paths for analysis and modeling of social networks. Graph filters can be represented by matrix-vector multiplication, where the input signal of system H is x, and the output is Hx , satisfying the following equation

$$H(\alpha x_1 + \beta x_2) = \alpha H(x_1) + \beta H(x_2) \tag{3}$$

A shift-time invariant map filter allows for the conversion of shift and filter operations that are sequentially equivalent. Specifically, the shift-time invariant map filter may be represented as

$$HSx = SHx \tag{4}$$

Where S is the graph shift operator and H is the graph filter. For any graph signals, a shift-invariant graph filter satisfies *HS* = *SH*. Prove that a graph filter showing linear, shift-invariant graphs is a polynomial of S. If *H* is a polynomial of *S*, then the graph filter *H* is linear and shift-invariant. If *H* is a polynomial of *S*, then the graph filter *H* is linear and shift-invariant. Accordingly, the following equation is obtained:

$$\tilde{x} = Hx = h(GFT^T \Lambda GFT)x = GFT^T[h(\Lambda)GFT * x] \tag{5}$$

Where Λ is the eigenvalue diagonal matrix of the graph Laplace matrix, GFT is the graph Fourier transform, and $h(\Lambda)$ is the frequency response of the filter. Equation (5) is an application of the convolution theorem to graph signal processing. In the nodal domain, the filtering can be interpreted as matrix-vector multiplication. From equation (5), the filter can be interpreted in the frequency domain as follows: first the input graph signal is Fourier transformed , then the graph filtering is performed in the frequency domain, and finally the inverse Fourier transform is performed on the signal。

2.5. Signal Noise Ratio

Signal noise ratio is one of the most important metrics for evaluating the performance of a filter, which indicates the relative strength or clarity between the signal and the noise. In graph signal processing, the signal-to-noise ratio of a graph filter can be calculated by the following equation:

$$SNR = 10\log_{10}\left(\frac{\|x\|_2^2}{\|x_\sigma\|_2^2}\right) \quad (6)$$

Where $\|x\|_2^2$ denotes signal power, usually expressed as a mean square value of the signal, and $\|x_\sigma\|_2^2$ denotes noise power, usually expressed as a mean square value of the noise. In graph signal processing, the signal may be the original graph signal or the signal output from the filter, and the noise is typically noise added to the signal or residual noise in the filter output.

Specifically, for signal noise ratio analysis of a graph filter, the following steps can be considered:

Calculate the signal power: For the original graph signal or the filter output, calculate its mean square value as the signal power. This can be obtained by squaring the signal and taking its mean value.

Calculate noise power: For noise added to the signal or residual noise in the filter output, similarly calculate its mean square value as the noise power. This can also be obtained by taking the square of the noise and averaging it.

Calculate the signal noise ratio: Using the above formula, divide the signal power by the noise power and take the logarithm to get the signal-to-noise ratio. Usually expressed in decibels (dB).

In practice, by analyzing the signal-to-noise ratio of a graph filter, you can assess its effective filtering of the signal and its ability to suppress noise. A higher signal-to-noise ratio means a clearer signal and less noise, thus indicating a better performance of the filter.

3. Performance Verification Test Design

In this section, the advantages of my proposed joint optimization graph filter design method[13] over the benchmark methods will be demonstrated experimentally. The two benchmark methods used as comparisons are the RM graph filter design method[14] and the denoising method based on Joint Tikhonov Regularization (JTR)[15]. It is worth noting that for fairness, the best weighting factor is taken at each signal noise ratio. In the JTR based denoising method, the weighting factor is optimal for each signal to noise ratio. This is to show that the filter design based on graph model with joint vertex sampling and integration is superior to the filter design based on static graph model.

The experiments in this study were conducted based on three real-world datasets, which are the Global Sea Level Pressure (GSLP) dataset, the Sea Surface Temperature (SST) dataset, and the Daily Mean PM2.5 Concentration in California (TDMPMC) dataset[16].

A detailed description of these datasets is provided below:

GSLP dataset: Published by the Joint Institute for the Study of the Atmosphere and Ocean (JISAO). The 50 selected nodes are distributed around the world from 30°N to 0°N, and from 100°E to 180°E. The nodes are located in the same area as the GSLP data set. Each node collects the global sea level barometric pressure signal, which is observed over 120 sampling periods.

SST dataset: Published by the Earth System Research Laboratory. Fifty nodes were selected over 120 sampling periods and are located in the Pacific Ocean from 30°S to 60°S and from 170°W to 90°W. The nodes are located in the Pacific Ocean from 30°S to 60°S and in the Pacific Ocean from 170°W to 90°W.

TDMPMC dataset: Published by the U.S. Environmental Protection Agency. This dataset collects data for the period January 1, 2020 to December 31, 2020 from 47 observation sites. The first 120 days of data were selected for simulation.

In the experiment, these datasets were used to evaluate the performance and effectiveness of the jointly optimized graph filters and to demonstrate their advantages and applicability by comparing the results.

4. Simulation Results and Analysis

In this paper, signal noise ratio is chosen as the performance evaluation index. Through experiments, we first verify the performance of the graph filter based on the union of vertex sampling and integration to cope with real datasets, and then verify the performance advantage by comparing the performance with two benchmark graph filters.

Figures 1, 2, and 3 show the graphs constructed using the k-nearest neighbor algorithm for selected nodes in the three datasets. In this algorithm, each sensor is considered as a vertex and is connected to its nearest k sensors through edges. In order to control the comparative analysis of variables, this study uses k=5, here is the control variable consistent.

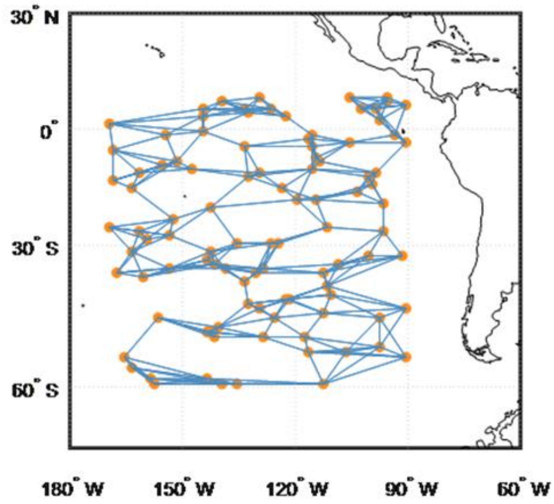

Figure 1. Sea Level Temperature Map Sensor Network Map

Tables 1, 2, Table 3 show the results of the signal-to-noise ratios of the FIR graph filters and ARMA graph filters designed based on the jointly optimized graph filter design method of vertex sampling and integration on the three datasets in comparison with the signal-to-noise ratios of the JTR-based method and the RM graph filter design method. By observing these tables, it is clear that the performance of the jointly optimized graph filters outperforms the JTR graph filter design method and the RM graph filter design method at input signal-to-noise ratios of -15, -10, -5, and 0. And the jointly optimized graph filters also have similar performance outputs when the input signal-to-noise ratio is 5, 10, and 15. These results further demonstrate the effectiveness and advantages of the joint optimization approach in graph filter design.

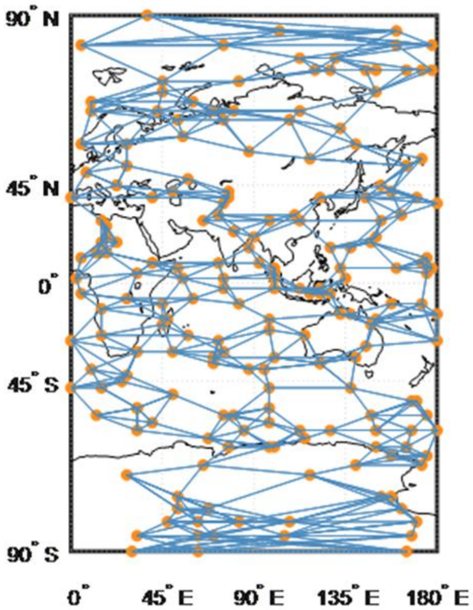

Figure 2. Sea Level Barometric Pressure Map Sensor Network Diagram

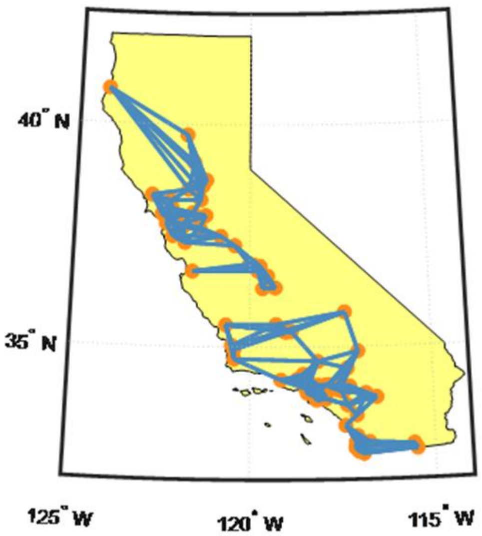

Figure 3. Observational Network Map of Daily Averages of pm2.5 Concentrations in California

Table 1. Sea level pressure data

SNR	-15	-10	-5	0	5	10	15
JTR	-9.85	-5.64	0.50	5.25	8.77	15.33	18.27
RM	-8.75	-1.88	2.46	9.11	14.78	20.33	25.29
FIR	-8.52	-0.59	2.28	10.25	15.42	22.07	24.81
ARMA	-7.52	-0.22	2.58	11.44	16.51	22.21	25.73

Table 2. Sea level temperature data

SNR	-15	-10	-5	0	5	10	15
JTR	-11.5	-7.04	-1.95	2.21	8.07	12.89	17.56
RM	-7.15	-1.24	1.23	4.01	7.80	11.33	16.21
FIR	-6.10	-1.02	1.22	4.09	8.01	11.29	16.31
ARMA	-5.98	0.17	1.92	4.88	8.74	12.02	17.30

Table 3. California pm2.5 data

SNR	-15	-10	-5	0	5	10	15
JTR	-12.7	-7.21	-2.63	1.01	8.22	12.31	17.47
RM	-5.92	-3.04	3.53	6.59	8.08	10.25	15.61
FIR	-5.10	-2.82	3.98	6.77	8.26	11.01	16.55
ARMA	-4.56	-2.01	4.37	7.63	9.74	13.17	18.26

5. Conclusion

In this paper, the feasibility and performance of the already proposed design method for graph filters based on the union of vertex sampling and integration are theoretically verified. Then experiments using real datasets are conducted to verify its actual performance. By comparing with two benchmark graph filter design methods, it is further verified that the joint optimized graph filter design method has some advantages. However, it is also noticed that the performance of this jointly optimized graph filter is a little bit insufficient in some datasets relative to the benchmark performance, which will be investigated in my subsequent research.

References

[1] F. Battiston, V. Nicosia, V. Latora, "Structural measures for multiplex networks," Physical Review E, vol.89, 2014.

[2] M. E. J. Newman,"The structure and function of complex networks," SIAM review, vol.45,pp. 167-256,2003.

[3] D. Wang, D. Pedreschi, C. Song," Human mobility, social ties, and link prediction, " Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining,pp. 1100-1108,2011.

[4] J. Leskovec, J. Kleinberg, C. Faloutsos, "Graph evolution: Densification and shrinking diameters," ACM transactions on Knowledge Discovery from Data (TKDD), vol. 1,pp. 2-es,2007.

[5] A. Sandryhaila, J. M. F. Moura," Discrete signal processing on graphs," IEEE transactions on signal processing, vol.61,pp. 1644-1656,2013.

[6] D. I. Shuman, S. K. Narang, P. Frossard, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other

irregular domains," IEEE signal processing magazine, vol. 30, pp. 83-98,2013.

[7] D. K. Hammond, P. Vandergheynst, R. Gribonval, "Wavelets on graphs via spectral graph theory," Applied and Computational Harmonic Analysis, vol. 30, pp. 129-150,2011.

[8] A. Ortega, P. Frossard, J. Kovačević," Graph signal processing: Overview, challenges, and applications," Proceedings of the IEEE, vol.106,pp. 808-828,2018.

[9] J. Zhou, G. Cui, S. Hu, ,"Graph neural networks: A review of methods and applications," AI open, vol. 1,pp. 57-81,2020.

[10] Y. Liu, Y. Zhang, W. Wang, et al, "ARMA graph filter design: Two approaches," IEEE Transactions on Signal Processing, vol. 67, pp. 4050-4063,2019.

[11] S. Tuomas, "Graph Filters: A Design Perspective," IEEE Signal Processing Magazine, vol.36,pp. 82-101,2019.

[12] S. Yangan, L. Zhang, J. Wang, et al," Wiener graph filter in graph spectral domain for image denoising," Signal Processing: Image Communication,vol. 85, 2020.

[13] B. Cui, and L. Cheng, "The joint optimization method of the graph filter design: the vertex sampling and the integration method,"Fifteenth International Conference on Signal Processing Systems (ICSPS 2023) ,2024.

[14] D. B. Tay, "Sensor network data denoising via recursive graph median filters, "Signal Process, vol.189,pp. 302,2021.

[15] F .Grassi, A. Loukas, N. Perraudin, " A time-vertex signal processing framework: scalable processing and meaningful representations for time-series on graphs, " IEEE Trans,Signal Process, vol.66,pp. 817–829,2018.

[16] K. Qiu, X .Mao, X. Shen,"Time-varying graph signal reconstruction, "IEEE J. Sel. Top. Signal Process, vol.11,pp. 870–883,2017.