Block Thresholding of Wavelets on Graphs

Jeffrey Williams, Raymond R. Hill, and Joseph J. Pignatiello Jr.
Air Force Institute of Technology
Wright-Patterson Air Force Base, Ohio, USA
jwilli038@gmail.com

Eric Chicken
Florida State University
Tallahasee, Florida, USA

Abstract

Graphs provide structure and context to data, both are relevant to the analysis and subsequent insight of information flow for social networks, transportation networks, and physical structures. Graph signal processing (GSP) is a collection of tools elevated from signal processing and used to analyze these abstract and physical networks. Similarly, this research elevates the idea of data neighborhoods to graphs to improve existing GSP wavelet thresholding tools for smoothing and approximating graph signals. We have found that data neighborhoods formed based on graph orientation, rather than spatial orientation, considerably improve the effectiveness of GSP tools, such as graph wavelet analysis, and ultimately understanding of networks and respective data flow.

1 Introduction

Graphs are useful for understanding network structure and information flow across abstract networks like social media and physical structures like road systems. Social media networks such as Twitter and Facebook are intangible, but the networks created by users and their connections are well defined and govern the flow of information. Graphs represent physical

systems such as road networks, which show the structure and areas on which vehicle traffic flow. In both of these instances, graphs allow us to analyze the flow of information throughout the network. "Graphs offer the ability to model such data and complex interactions among them. For example, users on Twitter can be modeled as nodes while their friend connections can be modeled as edges", (Ortega et al., 2018).

A local news tweet disperses to those users that "follow" the news account, these followers can propagate the tweet by "Retweeting" the news bulletin to their followers. Eventually, using connections like virtual highways, tweets can flow through an entire virtual community. A graph analysis shows us which local twitter networks are most likely to be reached such as subscribers to a homeowners association group which happens to follow the local news account.

These types of analysis are conducted with a group of tools called graph signal processing (GSP), which are often signal processing tools elevated to graphs. A good resource for GSP tools can be found in (Ortega et al., 2018) which present several tools that exploit the frequency of the graph. One of these GSP tools, which exploits graph frequencies, is wavelet analysis. In signal processing, wavelet analysis has smoothing and sparseness properties which make them an attractive tool for signal approximation and denoising. These properties are equally important for wavelet GSP signal approximation and denoising. Hammond and Gribonval (2011) lift wavelets to GSP with the spectral graph wavelet transform (SGWT). SGWT localizes wavelet filters applied to the spectral decompostion of a given graph to smooth the resultant approximation of a noisy graph signal. Alternatively, Irion and Saito (2014) build the Hierarchical Graph Laplacian Eigen Transform, which transforms the graph signal through a process called graph partitioning. Coifman and Maggioni (2006) present the diffusion wavelet where the basis is chosen from the graph structure based on the propagation of local and global graphical relationships. Similarly, to the concept of diffusion wavelets and best basis selections of Irion and Saito (2014), Defferrard et al. (2016) present a machine learning method to select a best basis using neural networks.

One of the challenges associated with wavelets on graphs is the limited work in graph wavelet thresholding for the various methods of generating wavelets on graphs. Irion and Saito (2014) employ the universal threshold effectively in conjunction with the HGLET partitioning method, and also prove this wavelet threshold is effective for HGLET graph transformations. de Loynes et al. (2021) present a GSP lifted SUREShrink method, originally proposed by Donoho and Johnstone (1995) for traditional signal processing. The GSP SUREShrink method has additional

terms and is used in conjuction with tight frames in order to compensate for the fact that noise is often correlated using the SGWT. This method is effectively applied to the Minnesota Road Map network, the NYC taxi cab, and a Facebook social network example. de Loynes et al. (2021) also lifts a James Stein (JS) type block thresholding estimator to GSP. This JS block thresholding technique was originally presented by Cai and Silverman (2001), but the author's note that it is not as effective as term-by-term thresholding, the GSP SUREShrink threshold estimator.

Our research augments the lifted JS type block thresholding method from Cai and Silverman (2001) and de Loynes et al. (2021), by changing how neighborhoods are defined. Rather than using a spatial definition of neighborhoods, we use graph frequency based neighborhoods for coefficients from Irion and Saito (2014). This method is common in GSP since many GSP techniques are elevated signal processing tools which leverage graph frequencies. Additionally, this method follows the observation of Ortega et al. (2018), that spectral ordering appears to be more effective than spatial ordering for graph wavelets. We will show that current JS type estimators are improved in terms of graph signal approximation, by using a different spectral definition for data neighborhoods.

2 Graph Preliminaries

A simple undirected graph consists of a set of interconnected vertices, called nodes, and edges. In application, graphs are typically weighted such that each edge describes some relationship between the vertices an edge connects. Like graphs the definition of edge weights are versatile and can adapt to the application. In some applications the edge weights represent distances between nodes or in the case of the work of Pang et al. (2015) the edge weights represent the difference in pixel intensities in a graph representation of an image. A simple undirected graph exists such that there are a finite number of vertices, edges do not connect a vertex to themselves, and at least one of the vertices does not have a predetermined flow direction. The graph theory applications in this paper have been extended to complex and directed graphs, but most of the research presented in this paper focuses on the simple undirected graph. Basic graph notation is $G := \{V, E, W\}$ where V represents the set of vertices, E represents the set of edges and W represents the edge weights. W is defined such that only vertices that are connected have a non-zero weight. The relations between the vertices in a graph are plotted in matrix form such

that the graph can be analyzed. This matrix is called the adjacency matrix, A. The adjacency matrix gets its name from the positive entries for a set, v_m and v_n where $v \in V$, which indicates there is an edge, $e \in E$, between the two vertices. The presence of an edge between two vertices indicates that the vertices are adjacent such that $A_{m,n} > 0$. In this case $a_{m,n} = 1$ or $a_{m,n} = w$ $\forall w \in W$ for a weighted graph. Note in a simple graph, $A_{m,n} = 0$ if $m = n$.

$$a_{m,n} : \begin{cases} w & e \in E \\ 0 & o.w. \end{cases} \tag{1}$$

The graph Laplacian, L, is found by $L = A - D$ where D is the degree matrix. The degree matrix is calculated by taking the sum of all adjacent weights for each vertex m such that $D(m) = \sum^{n} a_{m,n}$. D is a diagonal matrix which represents the degree of each vertex on the diagonal and zero elsewhere (Hammond and Gribonval, 2011). The normalized Laplacian is taken as $\mathcal{L}_N = D^{-1/2}\mathcal{L}D^{-1/2} = I - D^{-1/2}AD^{-1/2}$. Note that we work with the graph Laplacian in this paper, however the different graph Laplacians are equally valid quadratic forms of the graph, yet they yield different results upon spectral decomposition.

The Laplacian matrices are dealt with differently across research groups, but are generally reliant on some form of spectral decomposition or Eigen Value Decomposition (EVD) of the graph Laplacian. In other words, generating a complete, or partial approximation of the full set of the eigenvalues and associated eigenvectors of the graph Laplacian. The complete version of the EVD Equation, (2), shows the calculation for the EVD of L.

$$LX_L = \lambda_L X_L \tag{2}$$

X_L represents eigen vectors and λ_L represents the associated eigenvalues where it is assumed that the eigenvalues are ordered such that $0 <_0 \leq \lambda_1 \leq ... \leq \lambda_{N-1}$.

3 Spectral Graph Wavelet Transforms

A well known area of GSP is wavelet analysis on graphs. These are built by lifting the discrete wavelet transform to the graph. Hammond and Gribonval (2011) lift wavelet analysis to graphs and designs wavelets, such as the Meyer graph wavelet, based on the EVD of the graph Laplacian. The Meyer filters work in conjunction with the graph Fourier transform. Since the graph

Laplacian is a symmetric matrix of real values, we can express the graph Laplacian as an expansion of its eigenfunctions. The graph Fourier transform is shown in, equation (3), where the eigenvalues are positive since the graph Laplacian is positive semi-definite.

$$\hat{f}_{(l)} = \langle \mathcal{X}_l, f \rangle = \sum_{n=1}^{N} \mathcal{X}_l^*(n) f(n) \tag{3}$$

In this case the Parseval relation holds such that $\langle f, h \rangle = \langle \hat{f}, \hat{h} \rangle$ such that we can write the inverse graph Fourier transfer as in (4).

$$\hat{f}_{(n)} = \sum_{l=0}^{N-1} \hat{f}(l) \mathcal{X}_l(n) \tag{4}$$

In conjunction with the graph Fourier transform Hammond and Gribonval (2011) use wavelet operators to modulate each mode of the Fourier transform. Wavelet operators as designed in Hammond and Gribonval (2011) are measurable functions on the bounded and self-adjoint linear operator, the graph Laplacian, in a Hilbert space. Since we are using the spectral representation of the operator, or the EVD of the graph Laplacian we can effectively modulate each Fourier mode using a kernal g, or wavelet operator $T_g = g\mathcal{L}$ (Hammond and Gribonval, 2011). As stated we generate the spectral representation of the graph Laplacian and wavelet modulation according to (5) and the inverse according to (6).

$$\hat{T_g f}_{(l)} = g(\lambda_l) \hat{f}_l \tag{5}$$

$$(T_g f)(m) = \sum_{l=0}^{N-1} g(\lambda_l) \hat{f}_l \mathcal{X}_l(m) \tag{6}$$

Hammond and Gribonval (2011) use the Kroeneker Delta to isolate each wavelet coefficient to a vertex m in order to replicate the time localization properties of the classic discrete wavelet transform. This localization can be expressed according to equation (7).

$$\psi_{t,n} = T_g^t \delta n \tag{7}$$

The wavelet coefficients can be found explicitly by taking the inner product of each wavelet operator, $\psi_{t,n}$ where t is the positive real number and n is the vertex, and a function f, as in

(8). The scaling functions are determined in a similar manner such that $\phi_n = t_h\delta_n = h\mathcal{L}\delta_n$.

$$W_f(t,n) = (T_g^t f)(n) = \sum_{l=0}^{N-1} g(t\lambda_l)\hat{f}(l)\mathcal{X}_l(n) \tag{8}$$

The packages provided by Hammond and Gribonval (2011) in python, PYGSP, and de Loynes et al. (2021) in R, gasper, consolidate these equations creating a graph wavelet transform tool. In this paper, since we are interested in signal reconstruction, we use the Meyer wavelet in python for the tight frame properties.

4 Graph Wavelet Thresholding

Graphs and graph data clarify the behavior of networks and physical systems and how relevant data will interact with them. These insights are improved and obtained when noisy graph signals are denoised by relevant methods. There are a few instances of thresholding wavelets in GSP research. Irion and Saito (2014) add a soft thresholding scheme to both their Hierarchical Graph Laplacian Eigen Transform (HGLET) and the Generalized Haar Walsh Transform (GHWT). The universal threshold, $_{universal} = \sigma\sqrt{2log(n)}$, where n is the length of the signal and σ is the standard deviation of the graph wavelet coefficients, removes excess noisy GHWT and HGLET coefficients to improve the GHWT and HGLET approximations of a graph signal. In the Minnesota road network simulation from the work of Irion and Saito (2014), only 11 percent of GHWT transformed best-basis coefficients were retained and 48 percent of HGLET coefficients were retained, yet the approximation of the graph signal improved beyond the full set of coefficients.

de Loynes et al. (2021) adapt James Stein's (JS) SURE metric and JS block metrics to threshold SGWT coefficients and provide a justification for the use of the metric in the graph wavelet transform domain. They specifically include the work of Donoho and Johnstone (1995) and Cai (1999) among others as JS estimators which optimize MSE risk for graph wavelet estimators. They suggest additional terms in the estimators to include correlated noise in the graph wavelet domain. Equation (9) includes the covariance of the correlated noise in the divergence term for the JS type estimators. This is suggested since the SGWT are not completely orthogonal in rare cases.

$$SURE(h) = -n\sigma^2 + \|h(\tilde{F}) - \tilde{(}F)\|^2 + 2\sum i,j = 1n(J+1)Cov(_{i,j})\Sigma_j h_i(\tilde{(}F)) \tag{9}$$

The SGWT SUREShrink threshold estimator is term-by-term and level dependent. Term-by-term suggest each SGWT coefficient is thresholded separately without considering other SGWT coefficients generated according to equation (8). Level dependent implies that a SURE threshold value is determined for each graph wavelet filter level, j, such that the variance term is only impacted by the SGWT level of interest. This improved the SGWT graph signal approximation. The SUREShrink method is effective for multiple applications in de Loynes et al. (2021) and outperforms methods which are not using graph wavelet thresholding techniques, such as graph trend filtering.

We leverage the fact that de Loynes et al. (2021) lifts JS estimators to include block estimators to graph wavelets for the estimator of interest in this paper. We use the JS type estimator, NeighCoeff from Cai and Silverman (2001). NeighCoeff leverages information from immediate neighbors for a particular wavelet coefficient to estimate the threshold value for that particular coefficient. Due to the additional thresholding information, NeighCoeff decreases the MSE risk in signal processing estimation. In equation (10), we extend each coeff $L_0 = L_1 = 1$ by $L = 3$ coefficients and set $\lambda = \frac{2}{3}log(n)$. σ^2 is the variance of the graph wavelet coefficients. $S^2_{j,i} = \sum_{(j,k)\in B^j_i} \tilde{\theta}^2_{j,k}$ is the variance of the wavelet coefficient block which includes the L closest wavelet coefficients to the coefficient of interest. i represents the i^{th} block for coefficient k at resolution level j.

$$\hat{\theta_{j,k}} = B^j_k \tilde{\theta_{j,k}} = (1 - \lambda L \sigma^2 / S^2_{j,i})_+ \tilde{\theta_{j,k}} \tag{10}$$

In signal processing block thresholding leverages information from wavelet coefficients which are spatial neighbors. This is effective as wavelet coefficients have persistence and depth (Crouse et al., 1998). Therefore, spatial neighbors contain relevant and perhaps additional information to improve threshold criteria estimation since information from wavelet transforms persist through nearby wavelet coefficients. However, in graphs, spatial neighbors may not be similar. In the case of the Toronto road network, two spatially neighboring graph nodes could have different status, such as urban and rural areas. Considering dissimilar graph nodes for thresholding decisions may decrease the accuracy of the threshold estimation. Figure 1 shows the difference between spatial and spectral neighbors. The spectral neighbors include similar nodes in terms of graph frequencies based on the Euclidean distances between road intersections in Toronto.

In the context of the Minnesota road network, Irion and Saito (2014) argue that partitioning

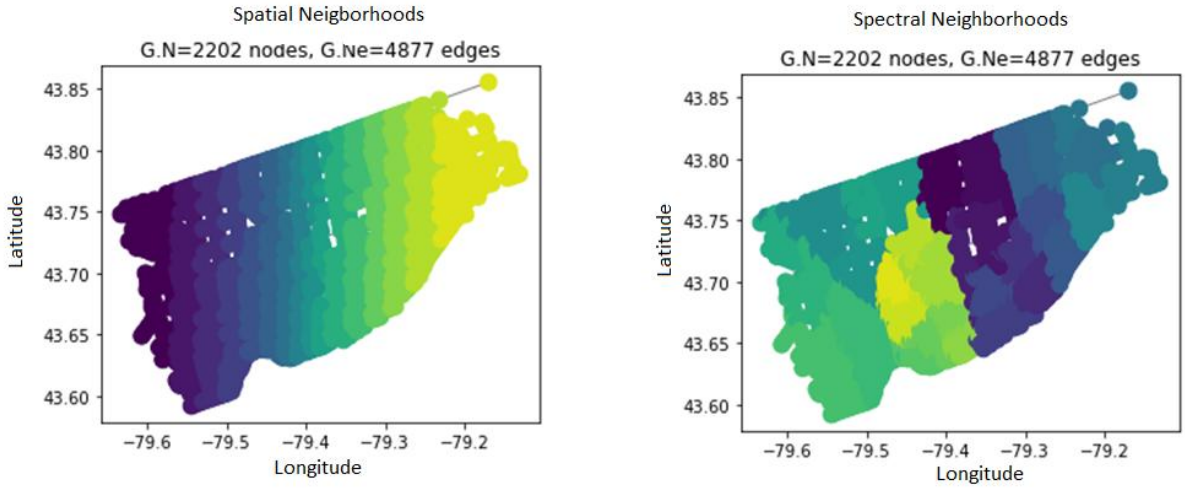

Figure 1: These graphs depict the Toronto road network where each node is an intersection and the edges are the roads that connect the intersections. The graph on the right shows the graph partitioned based solely on location. Notice the different blocks are just slices from South to North. The graph on the right shows the data neighborhoods based on spectral graph partitions. The neighborhoods are of different shapes and sized based on the frequency of each node.

graphs in a spectral manner is more effective than spatial organizations. Further, de Loynes et al. (2021) finds that block thresholding SGWT coefficients is less effective than term by term thresholding for graph signal estimation. In this paper, we adapt JS block estimators for SGWT coefficient thresholding to make wavelet block thresholding useful for graph signal approximation. Instead of using spatial neighbors for block thresholding we suggest using spectral neighbors based on graph frequencies.

5 WAVEBT

To address these practical issues with spatial block thresholding on graphs, we adopt the graph partition methods of Irion and Saito (2014) and create a fine-to-coarse partition dictionary. The coarsest level contains the original graph with N nodes and the finest levels contain N graphs with one node. Our contribution is that, our algorithm navigates the fine-to-coarse dictionary to form spectral blocks of graph wavelets where the block length is $L_0 = log(n)/2$. NeighCoeff, equation 10, is then used to threshold the graph wavelets in each graph partition.

6 Minnesota Road Network

In this section we conduct two separate experiments involving the Minnesota road network. The first is a synthetic signal with varying degrees of impulses from de Loynes et al. (2021). The intent of the different impulses is to simulate different data for the different areas of the graph. The signal is a Bernouli random variable with $f_{\nu,k} = W^k x_\nu / \lambda_1^k$. The two signals used in this experiment may be generated in the gasper library in R. The parameters $\nu = 0.01$ and $k = 2$ form one test signal and the parameters $\nu = 0.001$ and $k = 4$ form the second test signal. We test the first signal in this study with varying levels of white Gaussian noise where $\sigma = 0.005, 0.01, 0.02$. Table 1 shows that the block thresholding method outperforms SUREShrink for three separate noise values for the reconstruction of these graph signals with varying degrees of impulses.

Comparison of SureShrink method with the Block Thresholding Method

SNR/Method	SUREShrink	WAVEBT
0.005	115.62	**125.50**
0.010	38.73	**40.11**
0.020	12.70	**13.95**

Table 1: Comparison of SureShrink and Block Thresholding for the Minnesota Road Network and the Bernoulli Random Variable graph signal. The WaveBT method outperforms the SUREShrink method for various SNR for the Bernoulli Random Variable.

Irion and Saito (2014) present the HGLET and the HGWT methods which generate an orthonormal basis for a graph function based on graph partitioning. They successfully employ these methods for several simulated and real world data sets. One simulation involves plotting a mutilated Gaussian signal on the Minnesota graph network. The authors of this work provide a multi scale transforms for signal on graphs (MTSG) toolbox as well as the data described in this simulation, `https://github.com/JeffLIrion/MTSG_Toolbox`. The noise added to the signal in this simulation is white Gaussian noise with a signal to noise ratio (SNR) of 5.00 dB.

Figure 2 shows the denoising results for the noisy mutulated Gaussian graph signal. The SnR is calculated using $20log_{10}(||f||_2/||\hat{f} - f||_2)$. The GHWT performs the best with a SNR of 11.56 dB, Block Thresholding second at 7.01 dB, HGLET third at 6.77 dB, and SURE at 4.99 dB.

WaveBT uses a similar methodology to the GHWT, however, instead of graph partitioning and generating an orthonormal basis based on the EVD we partition the graph and apply a wavelet transform according to equation (5). This method is robust to many applications of graph signal denoising.

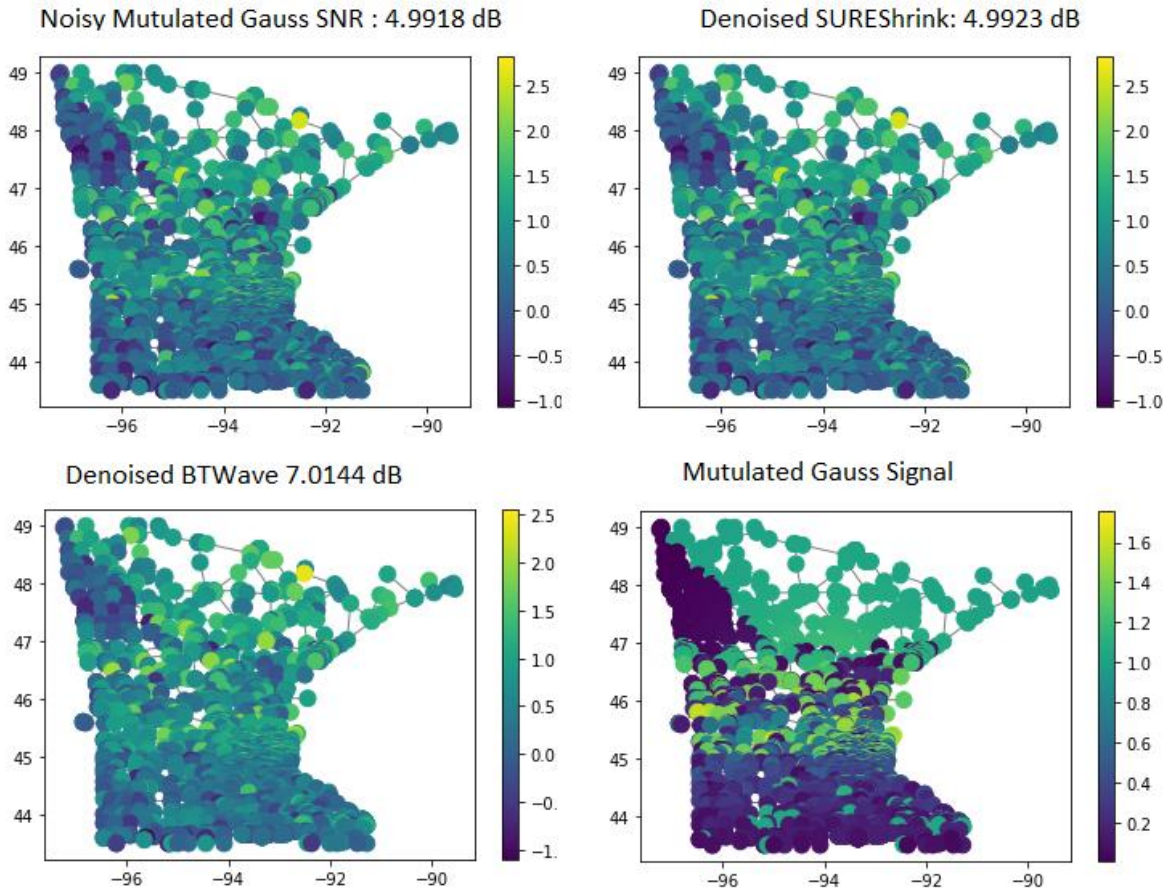

Figure 2: These figures represent the Minnesota road network with the mutilated Gaussian graph signal from (Irion and Saito, 2014). The first graph (from left to right) is the noisy signal with SNR 4.9918. The second graph is an approximation of the signal using SUREShrink, SnR 4.9923. The third graph is the approximation of the signal using WaveBT, SnR 7.01. The fourth graph is the original signal. Visually, the WaveBT graph signal, in particular regions of the graph, are more homogeneous than the noisy graph signal and the SUREShrink approximation.

Road Network-converted.pdf

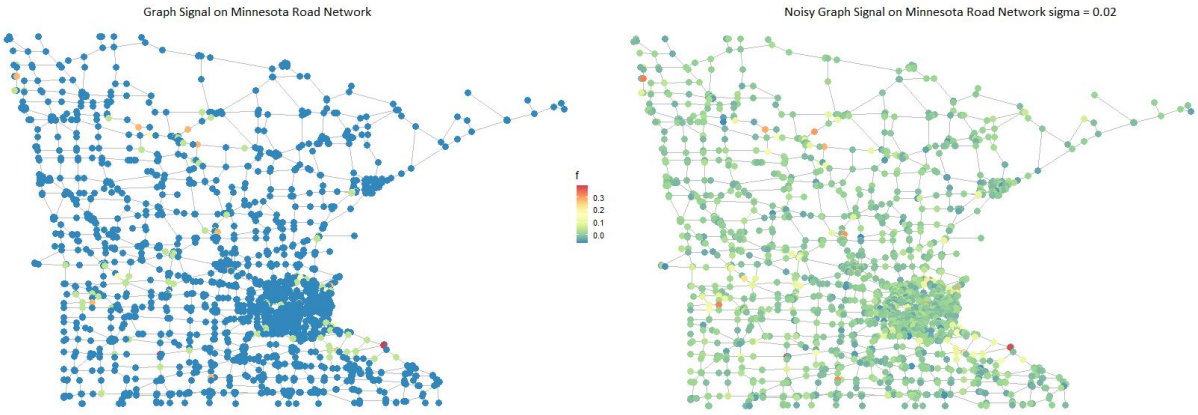

Figure 3: Minnesota Road Network with Bernoulli random variable signals from (de Loynes et al., 2021) overlayed. The graph on the left is the original function. The graph on the right is the original function with added noise.

7 Social Network

This Facebook graph from `Snap.Stanford.edu` is a nonplanar graph which represents Facebook users as nodes connected to other users or friends. The purpose of this type of graph is to analyze interconnected friend networks and analyze the flow of different types of data across these networks. This is valuable to marketing, political, and media organizations since it provides insight into how their data and informational and marketing products will spread throughout these types of social media networks. Figure 4 shows the graph of the 4,039 Facebook users with 88,234 "friend" connections.

Wang et al. (2015) simulate several different types of graph data which can be generated in R using the code provided by de Loynes et al. (2021) at `https://github.com/fabnavarro/SGWT-SURE`. Denoising these different types of signals gives a better approximation of how the signals behave on the graph. Figure 6 shows the different methods used for denoising the different signals. Block Thresholding, Sureshrink (de Loynes et al., 2021) and trend filtering (Wang et al., 2015) are used for denoising the different signals at -SnR of [-25,-20,-15,-10,5,0,-5,10]. Block wavelet thresholding has a larger negative signal to noise ratio across the different noise levels for the dense Poisson signal. Trend Filtering and SureShrink perform better than the Block thresholding technique for the lowest noise level of the inhomogoneous random walk model. The block thresholding model has a larger -SnR for the larger noise levels.

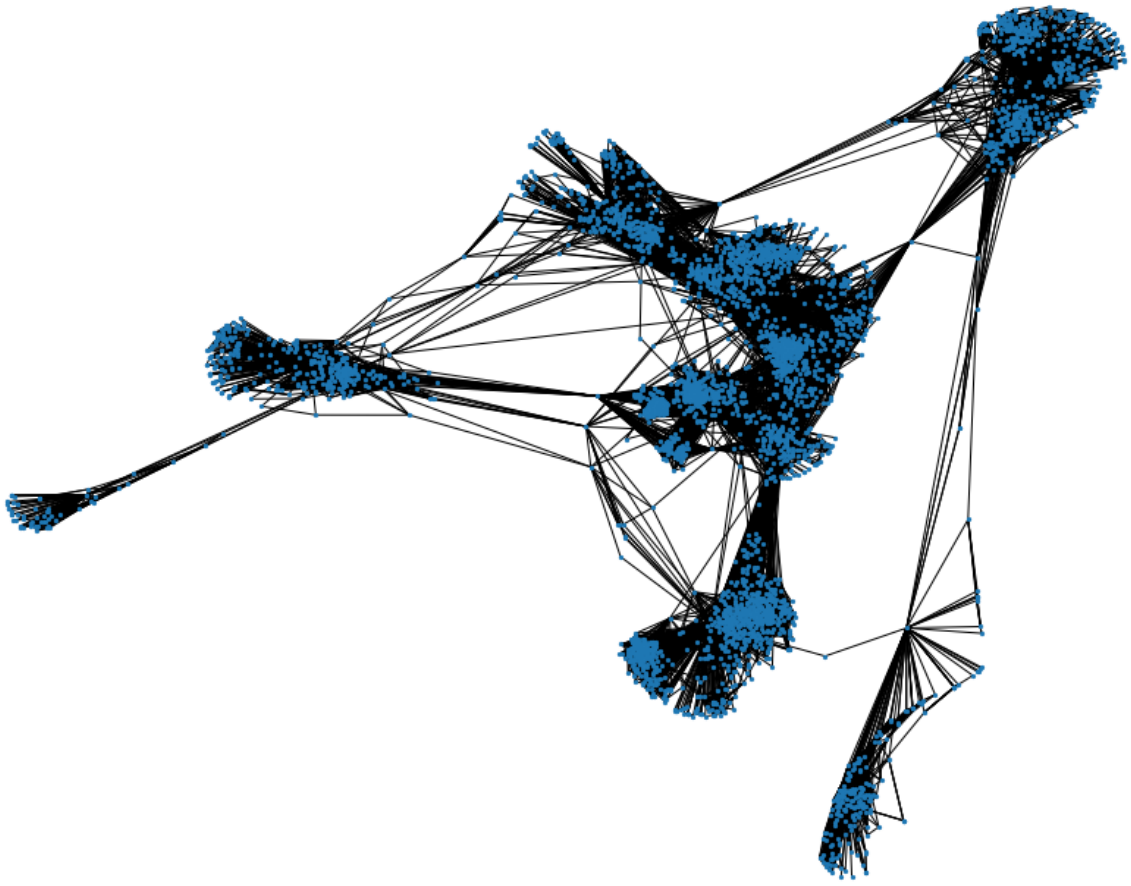

Figure 4: Graph from SNAP of 4039 social media users and their connections which provides an analytical structure representing the flow of information between friends on Facebook. The different clusters are ego centers, representing a single user with many connections. The graph shows an interesting degree of connection between these ego users and other ego users that are only connected through mutual friends

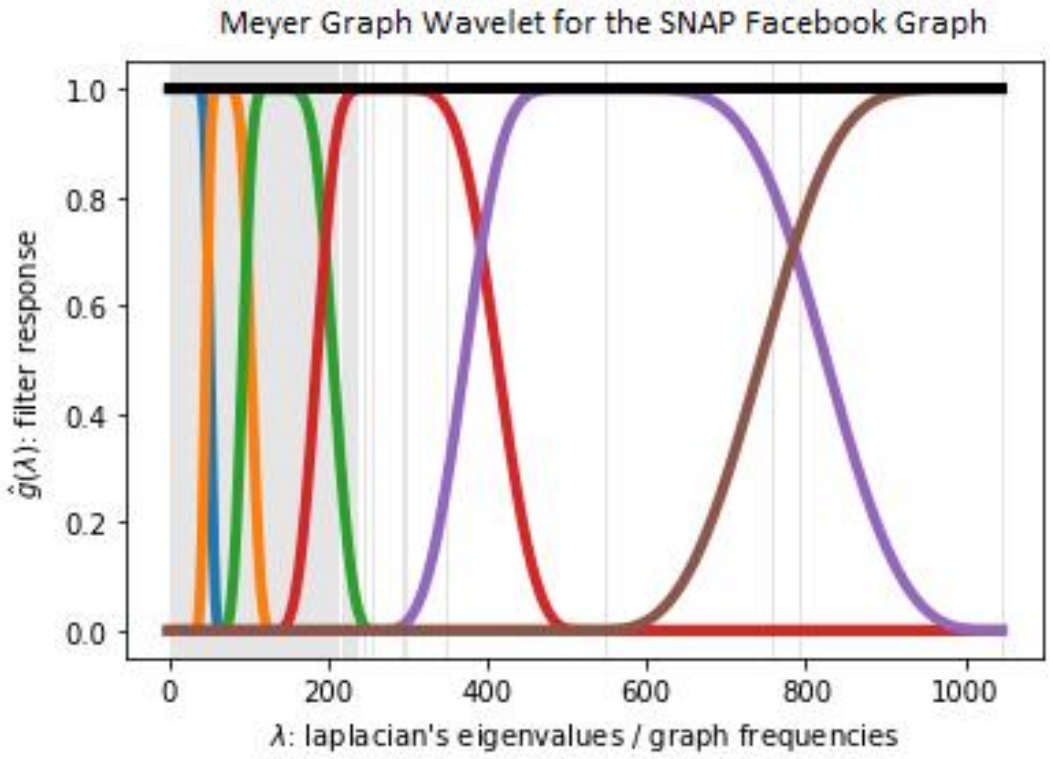

Figure 5: Graph Wavelet Frequencies: the graph frequencies are mostly lower frequencies with few higher frequencies. The lower frequencies represent the clusters of friends that are closely connected and the higher frequencies represent the few friends that connect clusters. The wavelet filters reflect this and the detailed wavelet filters from the SGWT cover the lower frequencies. The black lines at zero and one show the tight frame property of the Meyer graph wavelet

SNR-converted.pdf

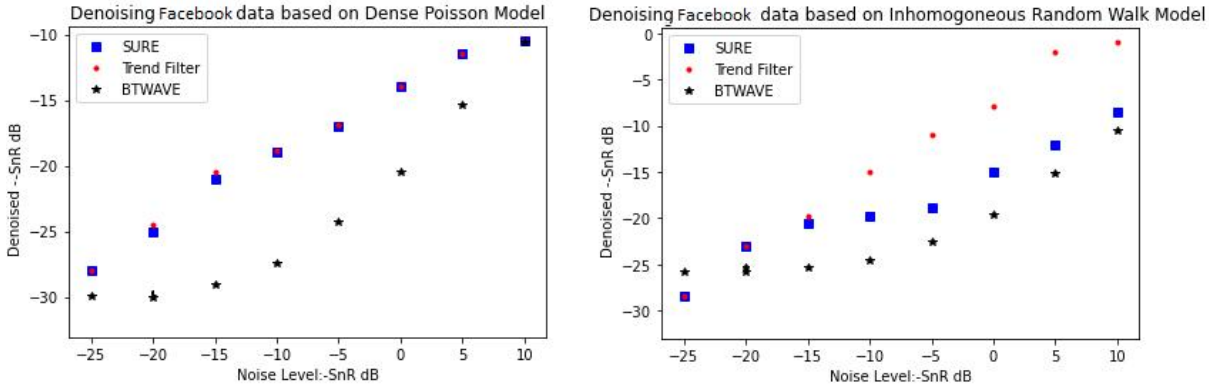

Figure 6: Graph of 4039 social media users denoised for two different test models applicable to social networks. The first graph shows the performance of three denoising methods for the Dense Poisson signal from (de Loynes et al., 2021). WaveBT outperforms SURE denoising and Trend Filtering for 8 of 9 noise levels, in terms of -SnR for the Dense Possion graph signal. The graph on the right shows that WaveBT outperforms the other methods for 8 of 9 noise levels for denoising the Inhomogonous Random Walk graph signal.

WaveBT leverages signals on neighboring graph nodes, with similar frequencies, to make a thresholding decision for a graph signal at a particular node. In this case, graph signals at nodes with similar frequencies are Facebook users with similar status. Those Facebook users that connect large ego circles, or large networks of friends, to other large ego circles have similar status. Those Facebook users whose connections lie within a specific ego cluster also have similar status. Figure 5 shows the different graph frequencies and the respective wavelet filters. Blocking graph wavelet coefficients to reflect similar user status is an intuitive and effective method for approximating and denoising signals on social network graphs.

8 Real World Toronto Road Map Traffic Data

In this section we apply WaveBT to real world traffic data for the Toronto road map. The graph of the Toronto road map is a network of nodes and edges where the nodes represent intersections in the city of Toronto and the edges represent the roads which connect them. The traffic data from Irion and Saito (2014) was collected over a 24 hour period at each intersection in the graph. The weighted adjacency matrix is based on the inverse Euclidean distance between intersecting nodes.

Partitioning of the Toronto Road Network.pdf

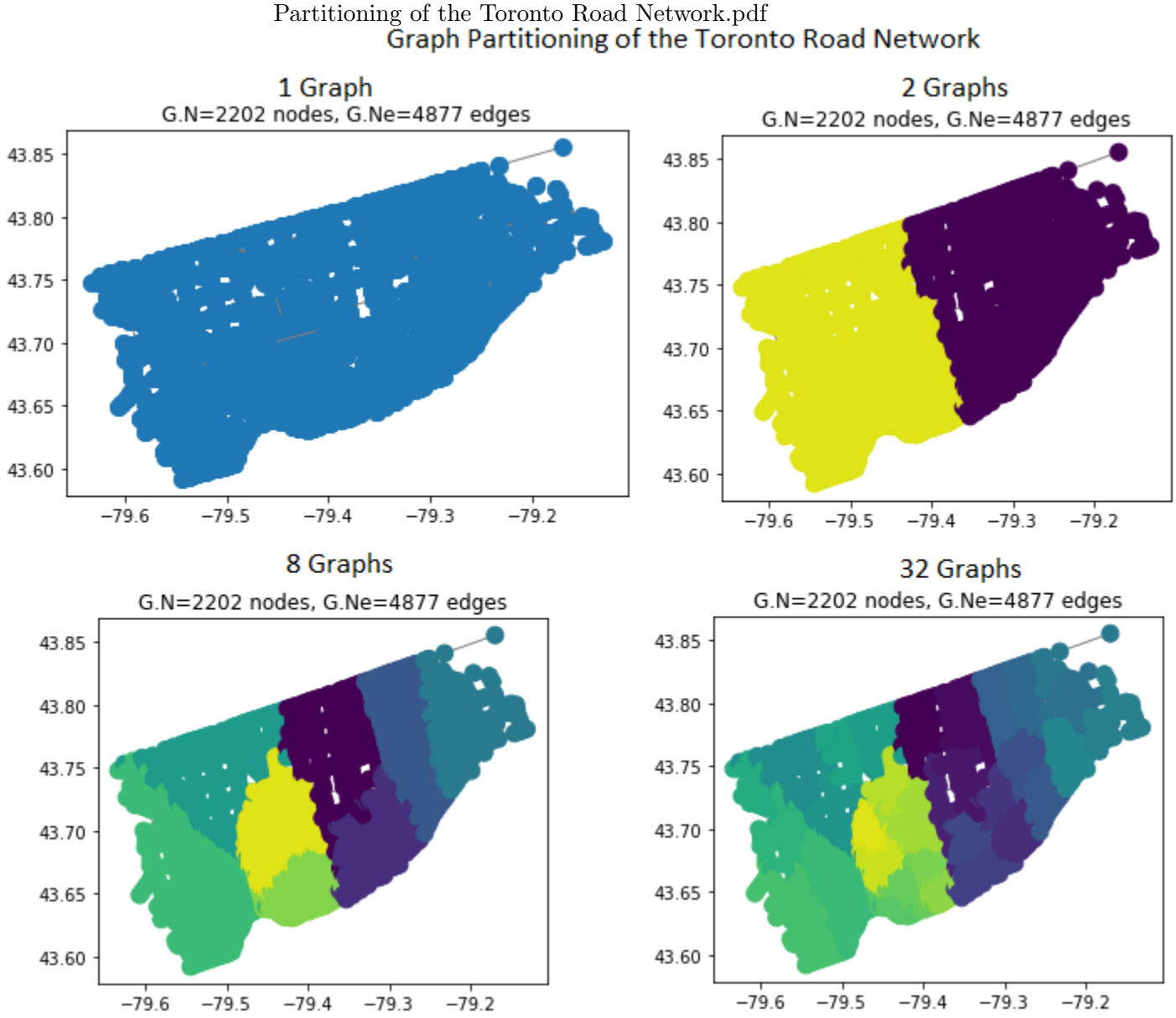

Figure 7: Graph partitioning of the Toronto road network for wavelet graph thresholding. Based on the inverse Euclidean distance between each node, the graph (top left) is partitioned into two graphs (top right), and progressively larger levels with finer graphs. A few examples of these are included such as 8 graphs (bottom left) and 32 graphs (bottom right). The puzzle peice shapes of the partition are a result of spectral partitioning rather than spatial. Spatial partitions are more block shaped, given the network has a rectangular shape, such as in road networks. In WaveBT, the partitions typically have different size, n, where n = log(N)/2

Table 2: Comparison of SureShrink and Probabilistic Block Thresholding

Comparison of SureShrink method with the Block Thresholding Method	
Method	Toronto
WaveBT	**9.10**
GHWT	8.23 dB
HGLET	8.96 dB

Partitioning graph wavelet coefficients based on the Toronto road network graph frequency is effective and intuitive. Coefficients thresholding decisions leverage information from similar graph signals at intersections with similar frequencies.

Figure 7 shows the original graph of the Toronto road network as well as the different levels of partitioning for the graph network. Different sectors of the city with similar road networks, based on distance between nodes and connectivity, are partitioned together. This is an effective method for blocking and subsequently thresholding graph wavelet signals for real world traffic data.

WaveBT improves upon other graph denoising methods such as the HGLET, GHWT and the SGWT with SUREshrink. Table 2, shows that graph wavelet block thresholding method outperforms the other methods in terms of SNR at 9.10 dB.

9 Conclusion

Using graph frequencies is an effective method for blocking wavelet coefficients. We use this block thresholding scheme for graph signals to improve wavelet approximation and denoising. We show this through several real world physical graphs with simulated and real world signals, achieving competitive and sometimes superior results with block thresholding. We also show block thresholding is an effective technique through social network graphs. Code and examples from the work are stored at `https://github.com/JeffreyDWilliams/GSPWaves`.

References

Cai, T. T. (1999). Adaptive wavelet estimation: A block thresholding and oracle inequality approach. *The Annals of Statistics 27*(3), 898–924.

Cai, T. T. and B. W. Silverman (2001). Incorporating information on neighbouring coefficients into wavelet estimation. *The Indian Journal of Statistics B*, 127–148.

Coifman, R. and M. Maggioni (2006). Diffusion wavelets. *Applied and Computational Harmonic Analysis 21(1)*, 53–94.

Crouse, M., R. Nowak, and R. Baraniuk (1998). Wavelet-based statistical signal processing using hidden markov models. *IEEE Transactions on signal processing 46*(4), 886–902.

de Loynes, B., F. Navarro, and B. Olivier (2021). Data-driven thresholding in denoising with spectral graph wavelet transform. *Journal of Computational and Applied Mathematics 389*, 113319.

Defferrard, M., X. Bresson, and P. Vandergheynst (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *ArXiv 1*, 1.

Donoho, D. L. and I. M. Johnstone (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association 90*(432), 1200–1224.

Hammond, D.K., V. P. and R. Gribonval (2011). Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis 30(2)*, 129–150.

Irion, J. and N. Saito (2014). Wavelets on graphs via spectral graph theory. *JSIAM Letters 6*, 21–24.

Ortega, A., P. Frossard, J. Kovaevi, J. Moura, and P. Vandergheynst (2018). Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE 106(5)*, 808–828.

Pang, J., G. Cheung, A. Ortega, and O. Au (2015, April). Optimal graph laplacian regularization for natural image denoising. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2294–2298. IEEE.

Wang, Y., J. Sharpnack, A. Smola, and R. Tibshirani (2015, February). Trend filtering on graphs. In *In Artificial Intelligence and Statistics*, pp. 1042–1050. PMLR.