Ali Hajimirza,
U. Oklahoma.
Jason White,
CSU San Bernardino.

Network Summarization

A Comparison of Methods

Mentors:
Sourav Medya
Arlei Silva
Prof. Ambuj Singh

Introduction

Networks are ubiquitous – they model numerous complex structures and processes. In order to analyze such networks, the size of the network data must be reduced; thus, network summarization becomes important. We experimented with three methods of compression: Slice Tree, Spectral Graph Fourier, and Spectral Graph Wavelets.

We examine each method for its scalability and accuracy on real and synthetic datasets. We discover that the Slice Tree algorithm is scalable and outperforms the Spectral Graph methods when the network values change smoothly across the topology.

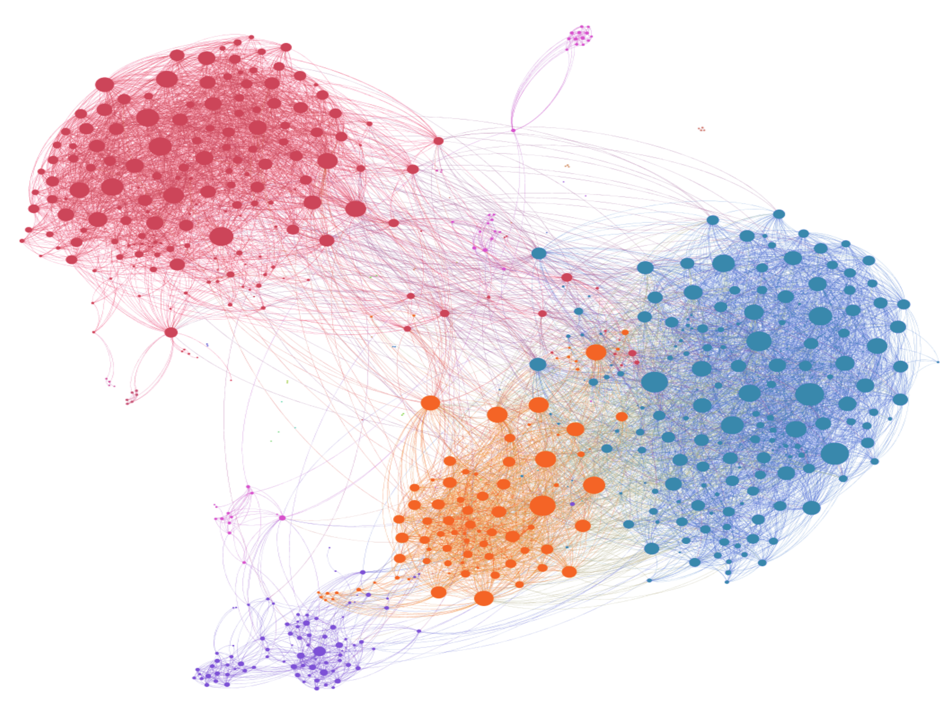

Figure 1: A snapshot of a Facebook friendship network.

Background

A **network** is a set of nodes and a set of edges that represent relationships between nodes. Networks generated by real-world complex systems are extremely large. For example, Twitter has 40 million nodes and 1.4 billion edges.

We are summarizing the dynamic **states** (or attributes) of the nodes. For example, in Traffic networks, the average speed in a location/node varies with time.

A **smooth** network contains states that do not change rapidly across the network.

Summarizing a Network

Slice Tree (ST)

- Slice Tree partitions a network into smooth regions such that each region can be compactly represented by the mean of the node values in that region.
- Each slice occupies 10 bytes and is represented by a center node, radius, and mean value.
- Finding the optimal partitioning given a budget is NP-hard. Slice Tree greedily computes the best slice at each iteration.
- The greedy algorithm is computationally expensive. Therefore, an importance sampling method is introduced to improve the runtime for very large networks.

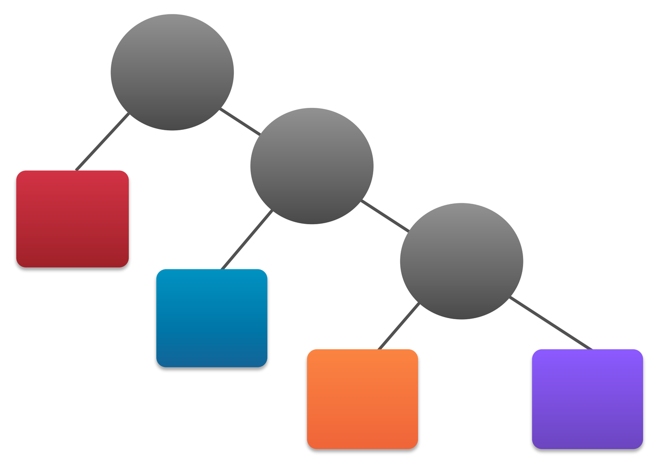

Figure 2: The Slice Tree of the network in Figure 1.

Spectral Graph Fourier

- Let W be an adjacency matrix and f an array of node values.
- Compute the Laplacian: $\mathcal{L} = D - W$
- Compute the eigenvalues (λ_l) and eigenvectors (u_l) of the Laplacian.
- Forward transform ↔ Inverse transform

$$\text{(I)} \quad \hat{f}(\lambda_l) := \sum_{i=1}^{N-1} f(i) u_l(i) \qquad \text{(II)} \quad f(i) = \sum_{l=0}^{N-1} \hat{f}(\lambda_l) u_l(i)$$

- Store the high signal values.
- Each signal value and its position occupy 12 bytes.
- Use the stored signal values to approximate the original values.

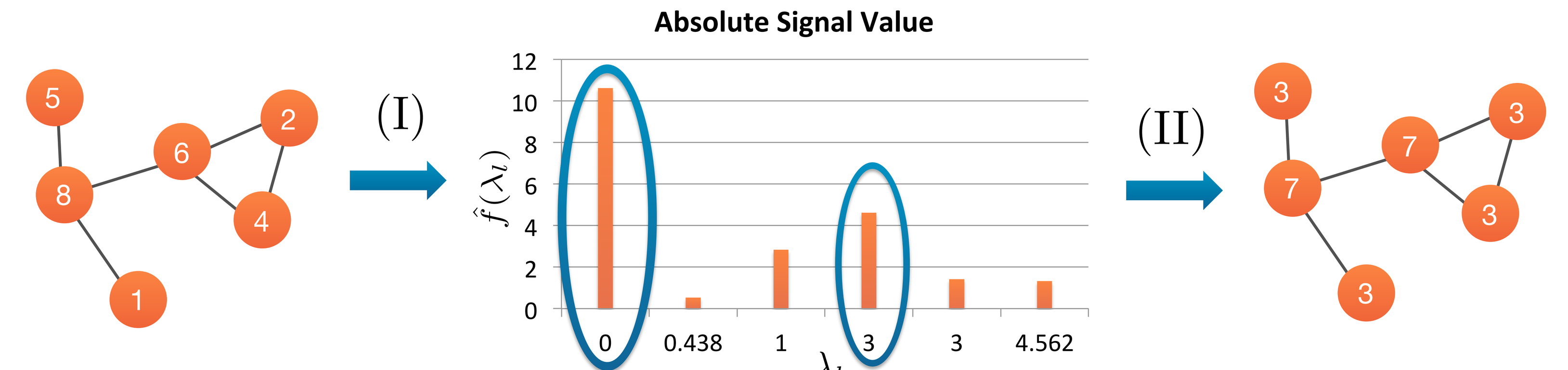

Figure 3: Fourier transformation from network to signal and back.

Experiments and Results

Scalability

- This experiment is designed to measure the scalability of the Fourier and Slice Tree algorithms.
- The compressed network size is kept constant (50 bytes) while the number of nodes are varied.
- Due to the time complexity of eigenvector decomposition, the time taken by Fourier grows exponentially.

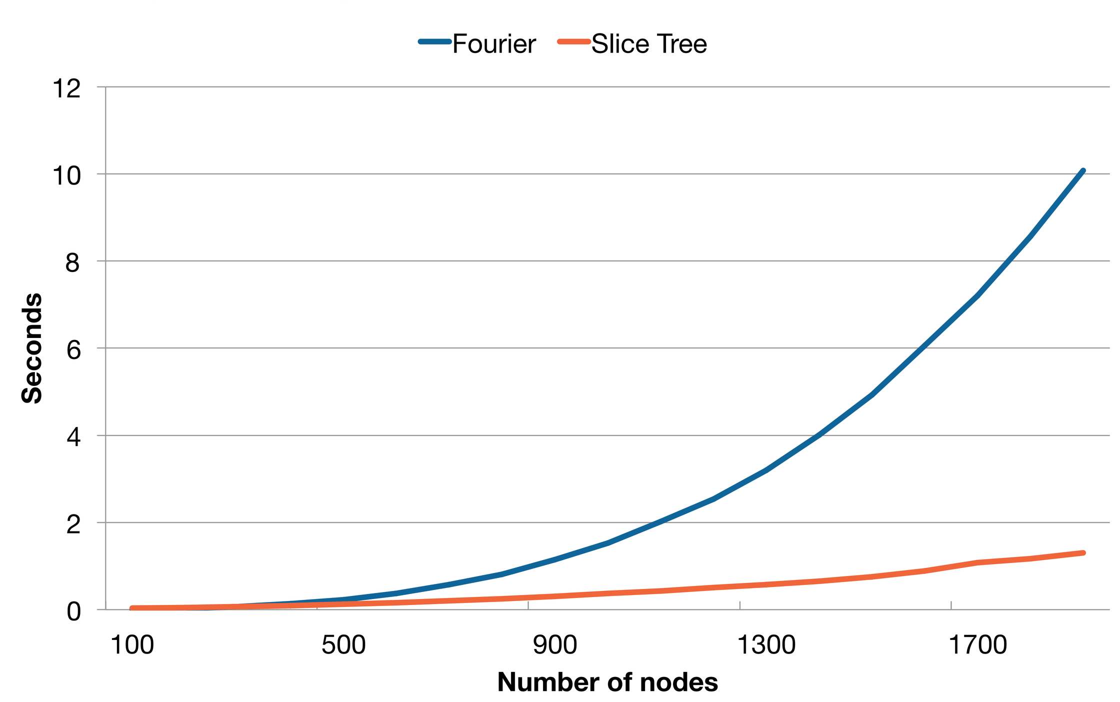

Figure 4: A comparison of runtimes for each algorithm on a traffic network.

Datasets

- We measure the error between the original and compressed network states.
- Each network is reduced to 1000 nodes and compressed to 600 bytes.
- The error is computed by the following formula:

$$Error(V) = \sum_{v \in V} (V(v)_{\text{orig}} - V(v)_{\text{final}})^2$$

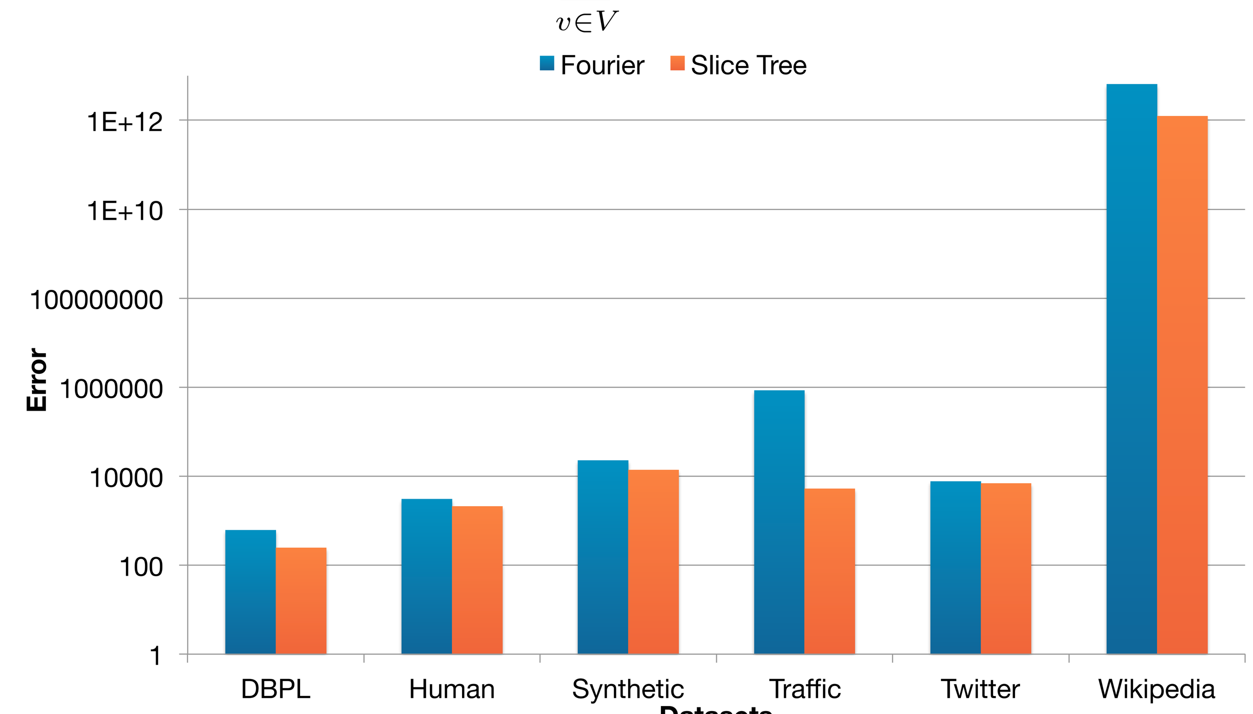

Figure 5: A comparison of error for each algorithm on networks with different value distributions (in logarithmic scale).

Accuracy

- The accuracy of the algorithms are determined by measuring the loss of information for different compressed sizes.
- Due to the smoothness of the network, Slice Tree gives a more accurate compressed network state.

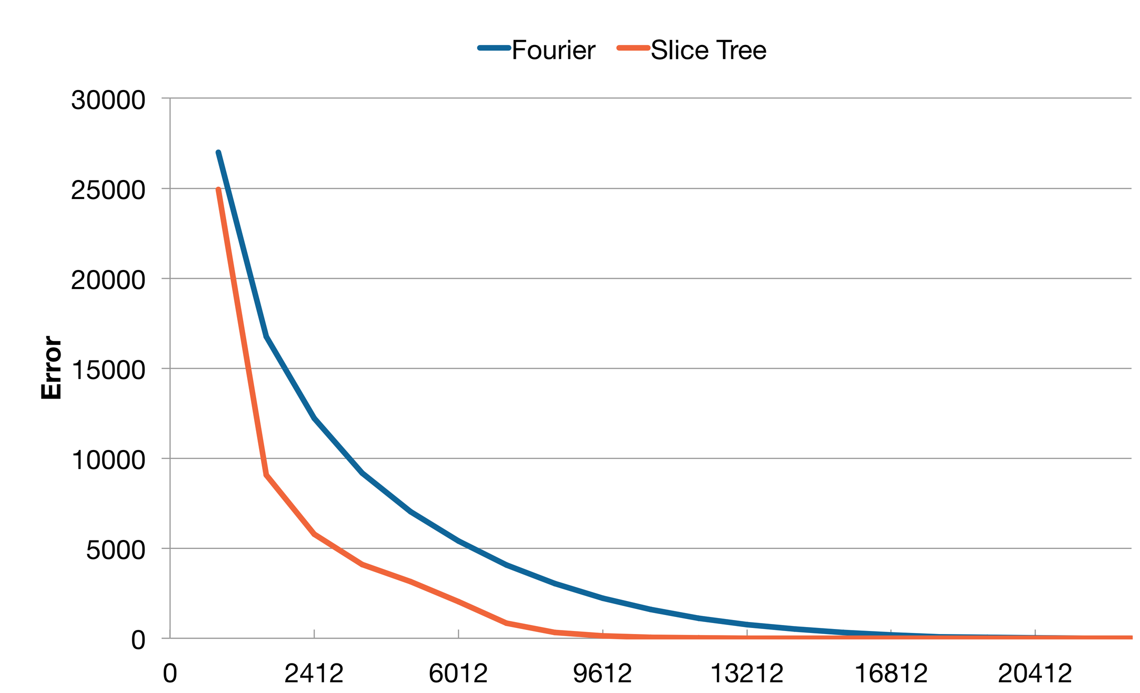

Figure 6: Accuracy of each algorithm with respect to the compressed network state using the traffic network.

Conclusion

- The Slice Tree algorithm scales much better than the Fourier algorithm.
- For smooth networks, Slice Tree gives better compression results.
- For "non-smooth" networks, the accuracy of Fourier is comparable to Slice Tree.

Future Work

- Implement the Spectral Graph Wavelets and compare against the implemented algorithms.
- Use Slice Tree to detect outlier node values in a network.
- Explore the possibilities of using the Slice Tree method to predict the values of missing nodes in a graph.

Acknowledgments:

- Special thanks to Prof. Ambuj Singh, Arlei Silva, Sourav Medya, and Tim Robinson.
- We would also like to thank the National Science Foundations and University of California Santa Barbara for funding our research.
- This work was supported in part by the National Science Foundation through grant numbers DGE-1258507 and IIS-1219254.

UCSB
UNIVERSITY OF CALIFORNIA
SANTA BARBARA

References:

- Silva, Arlei, Petko Bogdanov, and Ambuj Singh. "Network State Summarization via In-Graph Compression."
- D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *Signal Processing Magazine, IEEE*, 30(3):83–98, 2013.
- Hammond, David K., Pierre Vandergheynst, and Rémi Gribonval. Wavelets on Graphs via Spectral Graph Theory. *Applied and Computational Harmonic Analysis*: 129-50. Print.