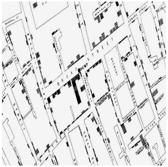
Graph Wavelets via Sparse Cuts

@SIGKDD'16, San Francisco, CA

Arlei Silva[1], Xuan-Hong Dang[1], Prithwish Basu[2], Ambuj K. Singh[1], Ananthram Swami[3]

[1]**Computer Science Department – University of California, Santa Barbara, CA**
[2]Raytheon BBN Technologies, Cambridge, MA
[3]Army Research Laboratory, Adelphi, MD

Graphs as a space



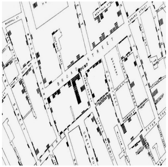
Water distribution

Social influence

Traffic

IoT

Graphs as a space



Water distribution

Social influence

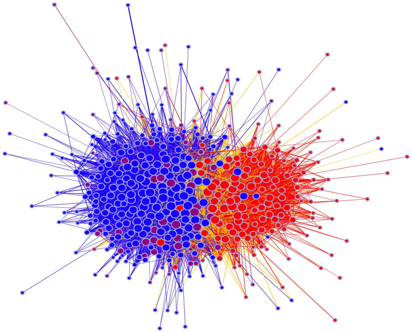
Traffic

IoT

We know a lot about structure but not much about how to represent information on top of graphs

Graphs as a space

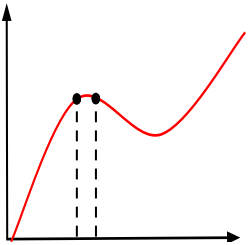
Node values as a function $W : V \to \mathbb{R}$



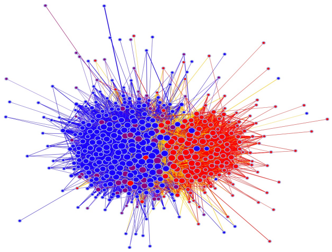
How to compress, de-noise, and sample W?

Smoothness

$$(u, v) \in E \to W(u) \approx W(v)$$



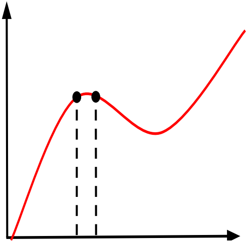
Function



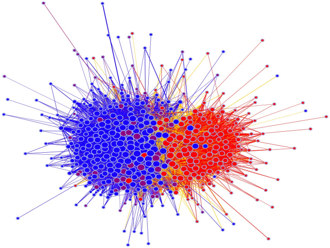
Graph

Smoothness

$$(u, v) \in E \rightarrow W(u) \approx W(v)$$



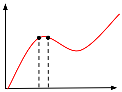
Function

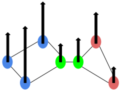


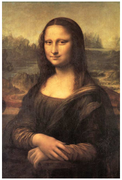
Graph

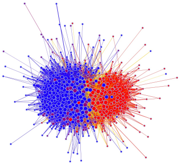
Sparse representation of W in some space

What framework can exploit the smoothness in the data? Signal processing









Euclidean space

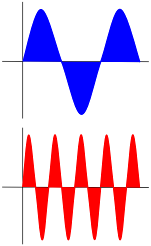
Graph space

How to generalize signal processing to graphs?

Signal Processing on Graphs

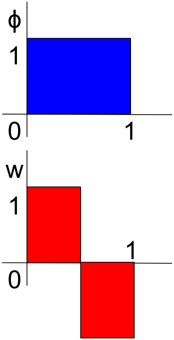
Popular approach: Graph Fourier [SNF$^+$13]

- Eigenvectors $(u_1, \dots u_n)$ of the Laplacian
- Transform: $\lambda_i(W) = \langle W, u_i \rangle$
- Inverse: $W(v) = \sum_i \lambda_i(W).u_{iv}$
- Problems handling localized signals



Alternative: Wavelets on Graphs

- Spectral theory [HVG11]
- Diffusion [MBCS05]
- **Partitioning** [GNC10]



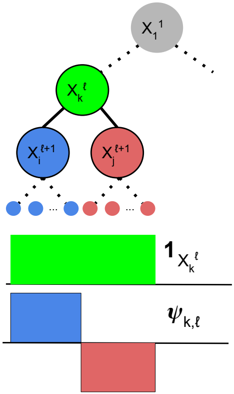
Wavelets on Graphs by Partitioning [GNC10][1]

Partitioning leads to a binary tree $\mathcal{X}(G)$

- $X_1^1 = V$
- $X_k^\ell \subseteq V$ has children $X_i^{\ell+1}$, $X_j^{\ell+1}$

Spaces of functions $\mathcal{V}_\ell$, $\mathcal{W}_\ell$

- $\mathcal{V}_\ell$ are functions constant over X_k^ℓ: $\{\mathbf{1}_{X_k^\ell}\}$
- $\mathcal{W}_\ell \perp \mathcal{V}_\ell$ are wavelet functions: $\{\psi_{k,\ell}\}$
- $\psi_{k,\ell}$ is piecewise constant on $X_i^{\ell+1}$, $X_j^{\ell+1}$
- $\psi_{k,\ell} \perp \mathbf{1}_{X_k^\ell}$



[1] [GNC10] Gavish, Nadler and Coiffman. *Multiscale wavelets on trees, graphs and high dimensional data. ICML'10.*

Wavelets on Graphs by Partitioning [GNC10]

Wavelet transform:

$$a_{k,\ell} = \frac{|X_j^{\ell+1}|}{|X_k^{\ell}|} \sum_{v \in X_i^{\ell+1}} W(v) - \frac{|X_i^{\ell+1}|}{|X_k^{\ell}|} \sum_{v \in X_j^{\ell+1}} W(v)$$

Energy of wavelet coefficient:

$$||a_{k,\ell}||_2 = \frac{a_{k,\ell}^2}{|X_i^{\ell+1}|} + \frac{a_{k,\ell}^2}{|X_j^{\ell+1}|}$$

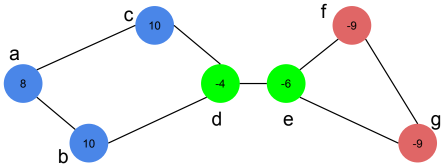
Wavelet inverse:

$$\varphi^{-1} W(v) = a_{0,0} + \sum_k \sum_\ell \nu_{k,\ell}(v) a_{k,\ell}$$

$$\nu_{k,\ell}(v) = \begin{cases} 1/|X_i^{\ell+1}|, & \text{if } v \in X_i^{\ell+1} \\ -1/|X_j^{\ell+1}|, & \text{if } v \in X_j^{\ell+1} \\ 0, & \text{otherwise} \end{cases}$$

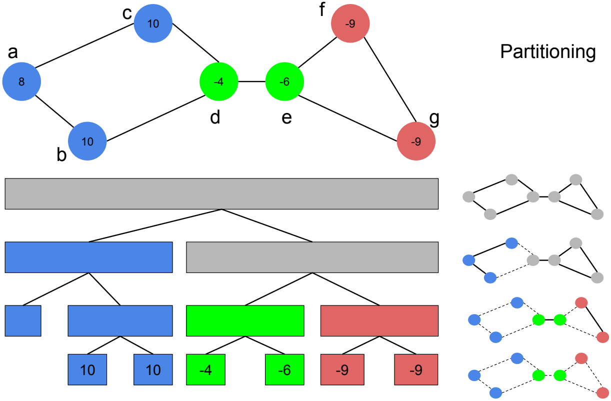
Wavelets on Graphs by Partitioning [GNC10]

Build a basis via partitioning + averaging/differencing



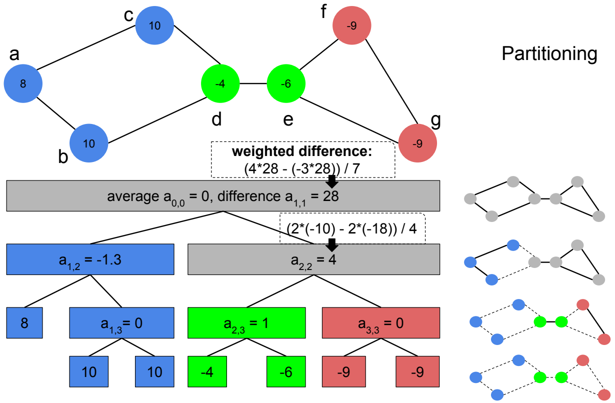
Wavelets on Graphs by Partitioning [GNC10]

Build a basis via partitioning + averaging/differencing



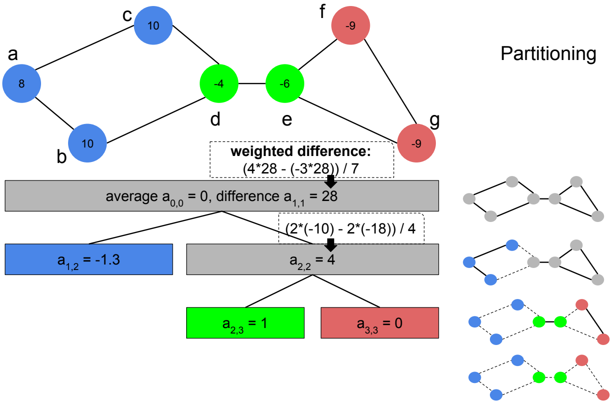
Wavelets on Graphs by Partitioning [GNC10]

Build a basis via partitioning + averaging/differencing



Wavelets on Graphs by Partitioning [GNC10]

Build a basis via partitioning + averaging/differencing

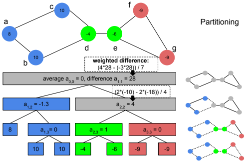


Wavelets on Graphs by Partitioning [GNC10]

Build a basis via partitioning + averaging/differencing

Advantages:

- Basis is orthogonal
- Smoothness of values is captured
- Graph partitioning is a familiar problem



Challenges:

- Sparsity of the transform depends on the basis/partitioning
- Good basis should capture the geometry of the data

Our Paper

Computing graph wavelet basis via sparse cuts

- Sparse cut leads to low-dimensional encoding

Connections with existing hard graph partitioning problems

- Graph bisection, multiway cuts
- NP-hard

Formulation as a relaxation of a vector optimization

- Spectral algorithm

More efficient approximate solution

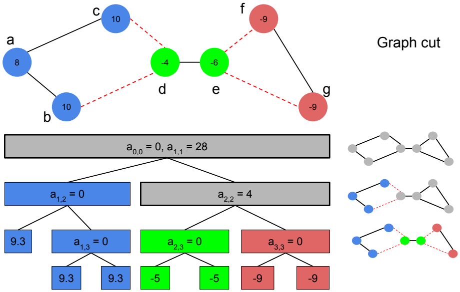
- Power method
- Chebyshev polynomials

Wavelet Basis via Sparse Cuts

$|\mathcal{X}(G)|_E$ is the size of the cut of a tree $\mathcal{X}(G)$

- Cut is a set of edges $E' \subseteq E$
- There is not path connecting leaves X_i^a, X_j^a in $G(V, E - E')$

Problem: Given a graph $G(V,E)$, signal W and a constant q compute a wavelet tree $\mathcal{X}(G)$ with cut $|\mathcal{X}(G)|_E$ of size q that minimizes the reconstruction error $||W - \varphi^{-1}\varphi W||_2$



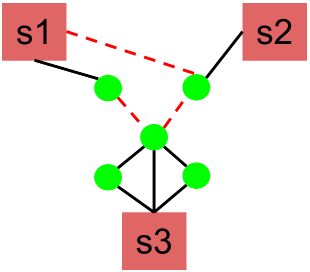
Wavelet Basis via Sparse Cuts: Hardness

Problem is NP-hard to approximate by a constant

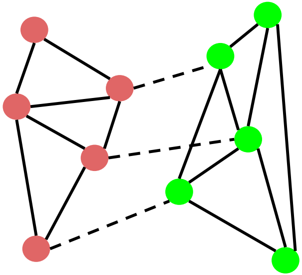
- Reduction from *3-Multiway Cut* [DJP^{+}92]

Performing each of the cuts is NP-hard

- Reduction *Graph Bisection* [BCLS87]



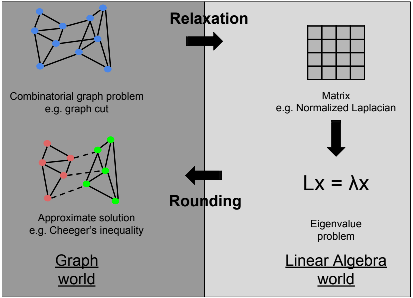
Multiway cut



Graph bisection

A Spectral Algorithm: Overview

Spectral Graph Theory [Chu97]



Our approach:

- Build matrices that capture graph structure and signal values
- Formulate our problem as a vector optimization problem
- Relax constraints and solve it as eigenvalue problem
- Round solution to discover sparse cut

A Spectral Algorithm: Formulation

Three matrices (L, C, S):

- Laplacian matrix of G: $L = D - A$
- Laplacian of a complete graph: $C = n\mathbf{I} - \mathbf{1}_{n\times n}$
- Signal matrix: $S_{u,v} = (W(u) - W(v))^2$

Indicator vector $\mathbf{x}$:

$$\mathbf{x}_v = \begin{cases} 1, & \text{if } v \in X_i^{\ell+1} \\ -1, & \text{if } v \in X_j^{\ell+1} \\ 0, & \text{otherwise} \end{cases}$$

Finding a sparse graph wavelet cut is equivalent to:

$$\mathbf{x}* = \max_{\mathbf{x}\in\{-1,1\}^n} ||a_{k,\ell}||_2 = \min_{\mathbf{x}\in\{-1,1\}^n} \frac{\mathbf{x}^\intercal CSC\mathbf{x}}{\mathbf{x}^\intercal C\mathbf{x}} \qquad st. \quad \mathbf{x}^\intercal L\mathbf{x} \leq 4q$$

where q is the cut size

A Spectral Algorithm: Relaxation

Regularized eigenvalue problem:

$$\mathbf{x}* = \min_{\mathbf{x} \in [-1,1]^n} \frac{\mathbf{x}^\intercal CSC\mathbf{x}}{\mathbf{x}^\intercal C\mathbf{x} + \beta \mathbf{x}^\intercal L\mathbf{x}}$$

where β can be searched over a line.

Using substitution $\mathbf{x} = ((C + \beta L)^+)^{\frac{1}{2}}\mathbf{y}$: $\mathbf{y}* = \min_{\mathbf{y}} \frac{\mathbf{y}^\intercal M\mathbf{y}}{\mathbf{y}^\intercal\mathbf{y}}$

Assuming W has 0-mean:

$$M_{ij} = 2n^2 \sum_{v=1}^{n} \sum_{u=1}^{n} \left(\left(\sum_{r=2}^{n} \frac{1}{\sqrt{\lambda_r}} e_{r,i} e_{r,u} \right) W(u).W(v) \right) \left(\sum_{r=2}^{n} \frac{1}{\sqrt{\lambda_r}} e_{r,v} e_{r,j} \right)$$

where (λ_r, e_r) is an eigenvalue-eigenvector pair of $(C + \beta L)$

A Spectral Algorithm: Relaxation

Regularized eigenvalue problem:

$$\mathbf{x}* = \min_{\boxed{\mathbf{x} \in [-1,1]^n}} \frac{\mathbf{x}^\intercal CSC\mathbf{x}}{\mathbf{x}^\intercal C\mathbf{x} + \boxed{\beta \mathbf{x}^\intercal L\mathbf{x}}}$$

Relaxed constraint **Regularization**

where β can be searched over a line.

Using substitution $\mathbf{x} = ((C + \beta L)^+)^{\frac{1}{2}}\mathbf{y}$**:** $\mathbf{y}* = \min_{\mathbf{y}} \frac{\mathbf{y}^\intercal M\mathbf{y}}{\mathbf{y}^\intercal\mathbf{y}}$

Assuming W **has 0-mean:**

$$M_{ij} = 2n^2 \sum_{v=1}^{n}\sum_{u=1}^{n} \left(\left(\sum_{r=2}^{n} \frac{1}{\sqrt{\lambda_r}} e_{r,i} e_{r,u} \right) W(u).W(v) \right) \left(\sum_{r=2}^{n} \frac{1}{\sqrt{\lambda_r}} e_{r,v} e_{r,j} \right)$$

where (λ_r, e_r) is an eigenvalue-eigenvector pair of $(C + \beta L)$

A Spectral Algorithm: Relaxation

Regularized eigenvalue problem:

$$\mathbf{x}* = \min_{\mathbf{x} \in [-1,1]^n} \frac{\mathbf{x}^\mathsf{T} C S C \mathbf{x}}{\mathbf{x}^\mathsf{T} C \mathbf{x} + \beta \mathbf{x}^\mathsf{T} L \mathbf{x}}$$

where β can be searched over a line.

Using substitution $\mathbf{x} = ((C + \beta L)^+)^{\frac{1}{2}} \mathbf{y}$**:** $\mathbf{y}* = \min_{\mathbf{y}} \frac{\mathbf{y}^\mathsf{T} M \mathbf{y}}{\mathbf{y}^\mathsf{T} \mathbf{y}}$

Assuming W **has 0-mean:**

$$M_{ij} = 2n^2 \sum_{v=1}^{n} \sum_{u=1}^{n} \left(\left(\sum_{r=2}^{n} \frac{1}{\sqrt{\lambda_r}} e_{r,i} e_{r,u} \right) W(u).W(v) \right) \left(\sum_{r=2}^{n} \frac{1}{\sqrt{\lambda_r}} e_{r,v} e_{r,j} \right)$$

where (λ_r, e_r) is an eigenvalue-eigenvector pair of $(C + \beta L)$

A Spectral Algorithm: Relaxation

Regularized eigenvalue problem:

$$\mathbf{x}* = \min_{\mathbf{x} \in [-1,1]^n} \frac{\mathbf{x}^\mathsf{T} CSC\mathbf{x}}{\mathbf{x}^\mathsf{T} C\mathbf{x} + \beta \mathbf{x}^\mathsf{T} L\mathbf{x}}$$

where β can be searched over a line.

Using substitution $\mathbf{x} = ((C + \beta L)^+)^{\frac{1}{2}}\mathbf{y}$**:** $\mathbf{y}* = \min_{\mathbf{y}} \frac{\mathbf{y}^\mathsf{T} M\mathbf{y}}{\mathbf{y}^\mathsf{T}\mathbf{y}}$

Assuming W **has 0-mean:**

$$M_{ij} = 2n^2 \sum_{v=1}^{n} \sum_{u=1}^{n} \left(\left(\underbrace{\boxed{\textstyle\sum_{r=2}^{n} \frac{1}{\sqrt{\lambda_r}} e_{r,i} e_{r,u}}}_{\sum_r g(\lambda_r) e_i e_i^\mathsf{T}} \right) \underbrace{\boxed{W(u).W(v)}}_{\text{Signal}} \right) \left(\boxed{\textstyle\sum_{r=2}^{n} \frac{1}{\sqrt{\lambda_r}} e_{r,v} e_{r,j}} \right)$$

where (λ_r, e_r) is an eigenvalue-eigenvector pair of $(C + \beta L)$

A Spectral Algorithm

Require: Graph G, values W, set X_k^ℓ,
regularization constant β, cut size q

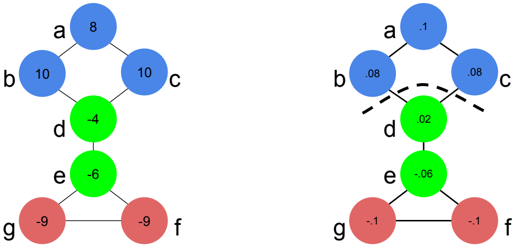
Ensure: Partitions $X_i^{\ell+1}$ and $X_j^{\ell+1}$

1: Create matrices C, L, S

2: $x* \leftarrow \min_x a(x)$ /*eigenvalue problem*/

3: $(X_1, X_2)_z \leftarrow$ cut $(\{1, 2 \ldots z\}, \{z+1 \ldots n\})$

4: $(X_i^{\ell+1}, X_j^{\ell+1}) \leftarrow \max_{(X_1,X_2)_j} ||a_{k,\ell}||_2$ st. cut size $|(X_1, X_2)| \leq q$



Graph signal Eigenvector/cut

A Spectral Algorithm: Performance

$$\mathbf{x} = ((C + \beta L)^{+})^{\frac{1}{2}}\mathbf{y}$$

$$\mathbf{y}* = \min_{\mathbf{y}} \frac{\mathbf{y}^{\mathsf{T}} M \mathbf{y}}{\mathbf{y}^{\mathsf{T}}\mathbf{y}}$$

Straighforward implementation is $O(sn^3)$

- s iterations to find β
- Pseudo-inverse computation $\approx O(n^3)$
- Eigenvalue computation $\approx O(n^3)$

Faster solution $O(pmn + tn^2)$**:**

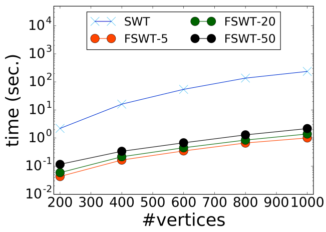
- Drop constant β
- $((L^{+})^{\frac{1}{2}} \times CSC)$ via *Chebyshev polynomials*: $O(pmn)$ [HVG11]
- Power method: $O(tn^2)$

Results: Scalability and Approximation

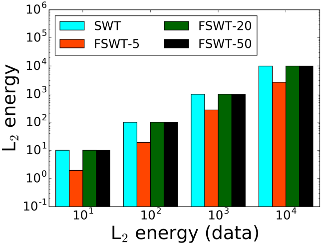
Synthetic data with best cut known

Exact algorithm (SWT) vs. fast version (FSWT)

FWST-X, X is the degree of Chebyshev polynomials



Scalability



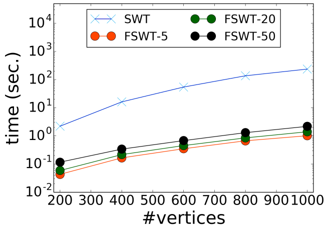
Approximation

Results: Scalability and Approximation

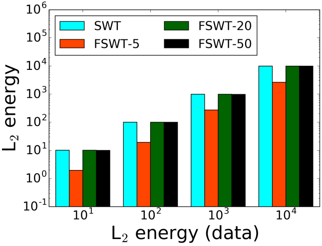
Synthetic data with best cut known

Exact algorithm (SWT) vs. fast version (FSWT)

FWST-X, X is the degree of Chebyshev polynomials



Scalability

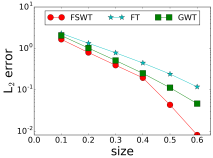


Approximation

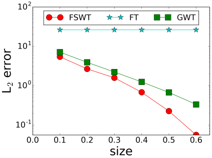
FSWT is up to 100 times faster than SWT
Competitive results for degree $\geq$ 20

Results: Value Compression

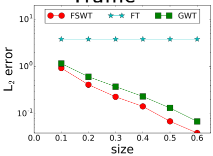
Baselines: Graph Fourier[2] (GF), Wavelets[34] (GWT, HWT)



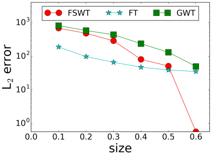
Traffic



Human



Wiki



Blogs

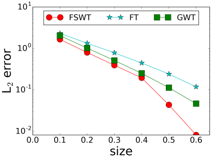
[2] [SNF+13] Shuman et al. *The emerging field of signal processing on graphs*

[3] [GNC10] Gavish et al. *Multiscale wavelets on trees, graphs and high dimensional data*

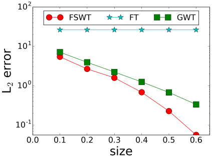
[4] [HVG11] Hammond et al. Wavelets on graphs via spectral graph theory

Results: Value Compression

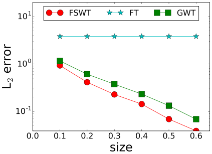
Baselines: Graph Fourier[2] (GF), Wavelets[3][4] (GWT, HWT)



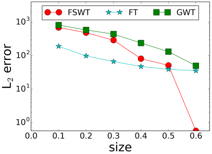
Traffic



Human



Wiki



Blogs

FSWT achieves up to 8 times lower error than the baselines

---

[2] [SNF+13] Shuman et al. *The emerging field of signal processing on graphs*

[3] [GNC10] Gavish et al. *Multiscale wavelets on trees, graphs and high dimensional data*

[4] [HVG11] Hammond et al. Wavelets on graphs via spectral graph theory

Final Remarks

In this paper we have studied the problem of computing data-driven graph wavelets via sparse cuts in order to represent graph signals in a compact and accurate manner

- Problem is NP-hard, even to approximate by constant
- Novel algorithm using Spectral Graph Theory
- More efficient solution using several techniques
- Better compression results than existing baselines

Future work:

- Study hardness of approximating a single cut
- Generalize approach to different types of wavelets
- Generalize approach to time-varying graphs

Graph Wavelets via Sparse Cuts

@SIGKDD'16, San Francisco, CA

Arlei Silva[1], Xuan-Hong Dang[1], Prithwish Basu[2], Ambuj K. Singh[1], Ananthram Swami[3]

[1]**Computer Science Department – University of California, Santa Barbara, CA**
[2]Raytheon BBN Technologies, Cambridge, MA
[3]Army Research Laboratory, Adelphi, MD

Research sponsored by the Army Lab under the Network Science CTA Cooperative Agreement Number W911NF-09-2-0053

References I

[BCLS87] Thang Bui, Soma Chaudhuri, Frank Leighton, and Michael Sipser.
Graph bisection algorithms with good average case behavior.
Combinatorica, 7:171–191, 1987.

[Chu97] Fan RK Chung.
Spectral graph theory.
American Mathematical Society, 1997.

[DJP+92] Elias Dahlhaus, David Johnson, Christos Papadimitriou, Paul Seymour, and Mihalis Yannakakis.
The complexity of multiway cuts.
In *STOC*, 1992.

[GNC10] Matan Gavish, Boaz Nadler, and Ronald Coifman.
Multiscale wavelets on trees, graphs and high dimensional data.
In *ICML*, 2010.

[HVG11] David Hammond, Pierre Vandergheynst, and Rémi Gribonval.
Wavelets on graphs via spectral graph theory.
Applied and Computational Harmonic Analysis, 30:129–150, 2011.

[MBCS05] Mauro Maggioni, James BremerJr, Ronald Coifman, and Arthur Szlam.
Biorthogonal diffusion wavelets for multiscale representations on manifolds and graphs.
In *SPIE*, 2005.

[SNF+13] David Shuman, Sunil Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst.
The emerging field of signal processing on graphs.
IEEE Signal Processing Magazine, 2013.