MAT 280: Harmonic Analysis on Graphs & Networks
Lecture 18: Wavelets on Graphs III
Organizing 'Dual' Domains of Graphs (Part 2)

Naoki Saito

Department of Mathematics
University of California, Davis

November 26, 2019

Outline

Acknowledgment

- Alex Cloninger (UC San Diego)
- Haotian Li (UC Davis)
- Qinglan Xia (UC Davis)
- NSF Grants: DMS-1418779, IIS-1631329, DMS-1912747, CCF-1934568
- ONR Grants: N00014-16-1-2255

Outline

Various Metrics for Comparing Eigenvectors

- A *similarity* measure based on the *average of local correlations* of eigenvectors (A. Cloninger & S. Steinerberger, 2018)
- The *difference of absolute gradient* (DAG) method (H. Li & N. Saito, 2019)
- The *time-stepping diffusion* (TSD) method (H. Li & N. Saito, 2019)
- Here, due to the time limitation, we will discuss only the TSD method.
- For the details of the latter two, see our paper: H. Li and N. Saito: "Metrics of graph Laplacian eigenvectors," in *Wavelets and Sparsity XVIII* (D. Van De Ville, M. Papadakis, and Y. M. Lu, eds.), Proc. SPIE 11138, Paper #111381K, 2019.

Various Metrics for Comparing Eigenvectors

- A *similarity* measure based on the *average of local correlations* of eigenvectors (A. Cloninger & S. Steinerberger, 2018)
- The *difference of absolute gradient* (DAG) method (H. Li & N. Saito, 2019)
- The *time-stepping diffusion* (TSD) method (H. Li & N. Saito, 2019)
- Here, due to the time limitation, we will discuss only the TSD method.
- For the details of the latter two, see our paper: H. Li and N. Saito: "Metrics of graph Laplacian eigenvectors," in *Wavelets and Sparsity XVIII* (D. Van De Ville, M. Papadakis, and Y. M. Lu, eds.), Proc. SPIE 11138, Paper #111381K, 2019.

Various Metrics for Comparing Eigenvectors

- A *similarity* measure based on the *average of local correlations* of eigenvectors (A. Cloninger & S. Steinerberger, 2018)
- The *difference of absolute gradient* (DAG) method (H. Li & N. Saito, 2019)
- The *time-stepping diffusion* (TSD) method (H. Li & N. Saito, 2019)
- Here, due to the time limitation, we will discuss only the TSD method.
- For the details of the latter two, see our paper: H. Li and N. Saito: "Metrics of graph Laplacian eigenvectors," in *Wavelets and Sparsity XVIII* (D. Van De Ville, M. Papadakis, and Y. M. Lu, eds.), Proc. SPIE 11138, Paper #111381K, 2019.

Various Metrics for Comparing Eigenvectors

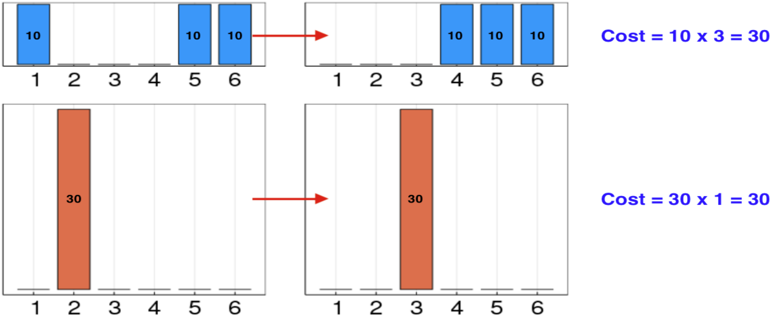
- A *similarity* measure based on the *average of local correlations* of eigenvectors (A. Cloninger & S. Steinerberger, 2018)
- The *difference of absolute gradient* (DAG) method (H. Li & N. Saito, 2019)
- The *time-stepping diffusion* (TSD) method (H. Li & N. Saito, 2019)
- Here, due to the time limitation, we will discuss only the TSD method.
- For the details of the latter two, see our paper: H. Li and N. Saito: "Metrics of graph Laplacian eigenvectors," in *Wavelets and Sparsity XVIII* (D. Van De Ville, M. Papadakis, and Y. M. Lu, eds.), Proc. SPIE 11138, Paper #111381K, 2019.

Various Metrics for Comparing Eigenvectors

- A *similarity* measure based on the *average of local correlations* of eigenvectors (A. Cloninger & S. Steinerberger, 2018)
- The *difference of absolute gradient* (DAG) method (H. Li & N. Saito, 2019)
- The *time-stepping diffusion* (TSD) method (H. Li & N. Saito, 2019)
- Here, due to the time limitation, we will discuss only the TSD method.
- For the details of the latter two, see our paper: H. Li and N. Saito: "Metrics of graph Laplacian eigenvectors," in *Wavelets and Sparsity XVIII* (D. Van De Ville, M. Papadakis, and Y. M. Lu, eds.), Proc. SPIE 11138, Paper #111381K, 2019.

Motivation of TSDM

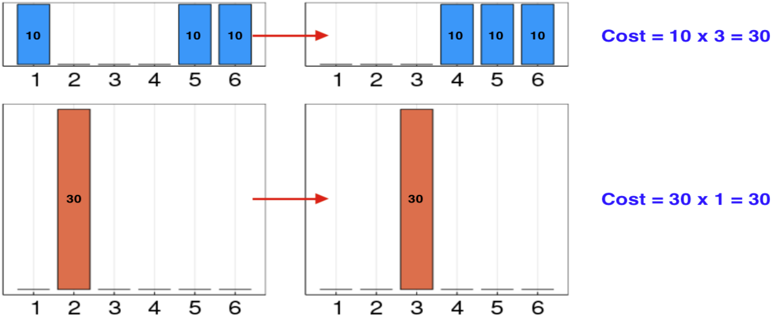
- Given any two mass distributions, one of them can be viewed as a distribution of earth (or sand) and the other a distribution of holes; then *Earth Mover's Distance* (a.k.a. *Wasserstein Distance*) between them is *the minimum cost of rearranging the mass in one distribution to obtain the other*, where *the cost = amount of earth moved × the distance by which it is moved*.



- EMD cannot distinguish these two transport schemes.

Motivation of TSDM

- Given any two mass distributions, one of them can be viewed as a distribution of earth (or sand) and the other a distribution of holes; then *Earth Mover's Distance* (a.k.a. *Wasserstein Distance*) between them is *the minimum cost of rearranging the mass in one distribution to obtain the other*, where *the cost = amount of earth moved × the distance by which it is moved*.



- EMD cannot distinguish these two transport schemes.

The Time-Stepping Diffusion Method (TSDM)

- The purpose of TSDM is to design an optimal transport-like method that *depends on time* (or *spatial scale and structure* of the underlying graph). In other words, at each given time, we want to measure a cost or a distance between eigenvectors.
- In order to measure *the optimal transport cost* between two vector measures (with the same total mass) on graphs, we need to first take the *difference* between two vector measures as the initial input, then compute *the minimum effort to flatten this initial input* on the graph.

The Time-Stepping Diffusion Method (TSDM)

- The purpose of TSDM is to design an optimal transport-like method that *depends on time* (or *spatial scale and structure* of the underlying graph). In other words, at each given time, we want to measure a cost or a distance between eigenvectors.
- In order to measure *the optimal transport cost* between two vector measures (with the same total mass) on graphs, we need to first take the *difference* between two vector measures as the initial input, then compute *the minimum effort to flatten this initial input* on the graph.

- Given a time T, let us consider a *diffusion process* on the graph. We want to measure the cost of "flatten" the initial graph signal via diffusion process up to the time T.
- We expect the graph signal will be flattened out by this process and *the final cost, as $T \to \infty$, would behave similarly with the optimal transport cost.*
- Notation: Denote the graph Laplacian matrix as L whose factorization is $L = \Phi \Lambda \Phi^{\mathsf{T}}$, where $\Phi = [\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_{n-1}]$, $\Lambda = \operatorname{diag}(\lambda_0, \lambda_1, \cdots, \lambda_{n-1})$, $0 = \lambda_0 < \lambda_1 \le \cdots \le \lambda_{n-1}$. Also, denote the directed incidence matrix of the graph G as $\tilde{Q} \in \mathbb{R}^{|V| \times |E|}$, and the graph gradient operator $\nabla_G := \tilde{Q}^{\mathsf{T}} : \mathbb{R}^{|V|} \to \mathbb{R}^{|E|}$.

- Given a time T, let us consider a *diffusion process* on the graph. We want to measure the cost of "flatten" the initial graph signal via diffusion process up to the time T.
- We expect the graph signal will be flattened out by this process and *the final cost, as $T \to \infty$, would behave similarly with the optimal transport cost.*
- Notation: Denote the graph Laplacian matrix as L whose factorization is $L = \Phi\Lambda\Phi^{\mathsf{T}}$, where $\Phi = [\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_{n-1}]$, $\Lambda = \mathrm{diag}(\lambda_0, \lambda_1, \cdots, \lambda_{n-1})$, $0 = \lambda_0 < \lambda_1 \le \cdots \le \lambda_{n-1}$. Also, denote the directed incidence matrix of the graph G as $\tilde{Q} \in \mathbb{R}^{|V| \times |E|}$, and the graph gradient operator $\nabla_G := \tilde{Q}^{\mathsf{T}} : \mathbb{R}^{|V|} \to \mathbb{R}^{|E|}$.

- Given a time T, let us consider a *diffusion process* on the graph. We want to measure the cost of "flatten" the initial graph signal via diffusion process up to the time T.
- We expect the graph signal will be flattened out by this process and *the final cost, as $T \to \infty$, would behave similarly with the optimal transport cost.*
- Notation: Denote the graph Laplacian matrix as L whose factorization is $L = \Phi\Lambda\Phi^{\mathsf{T}}$, where $\Phi = [\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_{n-1}]$, $\Lambda = \operatorname{diag}(\lambda_0, \lambda_1, \cdots, \lambda_{n-1})$, $0 = \lambda_0 < \lambda_1 \le \cdots \le \lambda_{n-1}$. Also, denote the directed incidence matrix of the graph G as $\tilde{Q} \in \mathbb{R}^{|V| \times |E|}$, and the graph gradient operator $\nabla_G := \tilde{Q}^{\mathsf{T}} : \mathbb{R}^{|V|} \to \mathbb{R}^{|E|}$.

TSDM and the Heat Equation

Given initial $\boldsymbol{f}_0$, the governing ODE system which describes the graph signal $\boldsymbol{u}(t)$'s ($\in \mathbb{R}^n$) evolution is following:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) + L \cdot \boldsymbol{u}(t) = \boldsymbol{0} \qquad \boldsymbol{u}(0) = \boldsymbol{f}_0 \in \mathbb{R}^n$$

Since $\{\boldsymbol{\phi}_0, \cdots, \boldsymbol{\phi}_{n-1}\}$ forms an ONB of $\mathbb{R}^n$, we have $\boldsymbol{u}(t) = \sum_{k=0}^{n-1} C_k(t) \cdot \boldsymbol{\phi}_k$.

Then, after plugging it into the above ODE system and solving for $C_k(t)$, we get $C_k(t) = \langle \boldsymbol{f}_0, \boldsymbol{\phi}_k \rangle e^{-\lambda_k t}$. Now, we have the solution:

$$\boldsymbol{u}(t) = \sum_{k=0}^{n-1} \langle \boldsymbol{f}_0, \boldsymbol{\phi}_k \rangle e^{-\lambda_k t} \boldsymbol{\phi}_k$$

At a certain time T, let us define the cost of the TSDM, $K(\boldsymbol{f}_0; T)$, by:

$$K(\boldsymbol{f}_0; T) := \int_0^T \|\nabla_G \boldsymbol{u}(t)\|_1 \,\mathrm{d}t$$

Convergence of TSDM

Theorem (Convergence of TSDM)

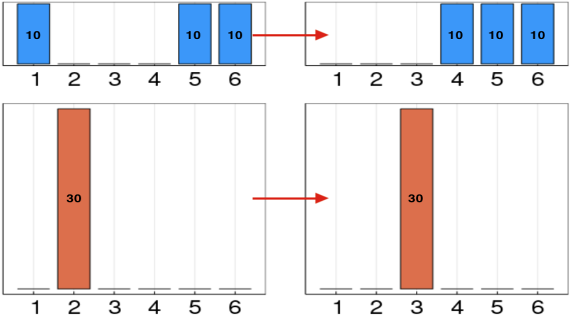
Let $G = (V, E, W)$ *be a connected undirected graph and* $\boldsymbol{f}_0$ *as the initial graph signal.* $K(\boldsymbol{f}_0; T)$ *converges as* $T \to \infty$, *i.e.,*

$$\lim_{T\to\infty} K(\boldsymbol{f}_0; T) = \lim_{T\to\infty} \int_0^T \|\nabla_G \boldsymbol{u}(t)\|_1 \, \mathrm{d}t < \infty$$

- Furthermore, we can show that for any fixed $T > 0$ (including $T = \infty$), $K(\cdot\,; T)$ is a *norm* on $L_0^2(V) := \{f \in L^2(V) : \sum_{x\in V} f(x) = 0\}$.

Comparative Results

- Optimal transport cost:



Cost = 10 x 3 = 30

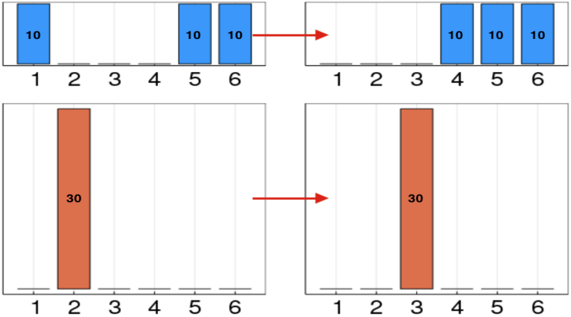
Cost = 30 x 1 = 30

- TSDM cost:

time	$T = 0.1$	$T = 1$	$T = 10$	$T = \infty$
blue cost	2.79	16.66	38.30	40.32
orange cost	10.41	38.42	63.65	65.87

Comparative Results

- Optimal transport cost:



Cost = 10 x 3 = 30

Cost = 30 x 1 = 30

- TSDM cost:

time	$T = 0.1$	$T = 1$	$T = 10$	$T = \infty$
blue cost	2.79	16.66	38.30	40.32
orange cost	10.41	38.42	63.65	65.87

The Cost Conjecture

- As time $T \to \infty$, one might expect the TSDM cost to be close to the optimal transport cost (i.e., the 1^{st} Wasserstein distance) between any two vector measures defined on the graph.

Conjecture

Given any two probability distributions p, q on a connected graph $G = (V, E, W)$ with graph geodesic distance metric $d : V \times V \to \mathbb{R}_{\geq 0}$,

$$W_1(p,q) \leq K(p-q;\infty) \leq C \cdot W_1(p,q)$$

in which $W_1(p,q) := \inf_{\gamma \in \Gamma(p,q)} \int_{V \times V} d(x,y)\,\mathrm{d}\gamma(x,y)$, where $\Gamma(p,q)$ denotes the collection of all measures on $V \times V$ with marginals p and q in the first and second factors respectively and C is a constant dependent on G.

- We manage to prove the first half of the conjecture, i.e., $W_1(p,q) \leq K(p-q;\infty)$, but it is still a mystery about the explicit formulation of the constant C in second half.

The Cost Conjecture

- As time $T \to \infty$, one might expect the TSDM cost to be close to the optimal transport cost (i.e., the 1^{st} Wasserstein distance) between any two vector measures defined on the graph.

Conjecture

Given any two probability distributions p, q on a connected graph $G = (V, E, W)$ with graph geodesic distance metric $d : V \times V \to \mathbb{R}_{\geq 0}$,

$$W_1(p, q) \leq K(p - q; \infty) \leq C \cdot W_1(p, q)$$

in which $W_1(p, q) := \inf_{\gamma \in \Gamma(p,q)} \int_{V \times V} d(x, y) \, \mathrm{d}\gamma(x, y)$, where $\Gamma(p, q)$ denotes the collection of all measures on $V \times V$ with marginals p and q in the first and second factors respectively and C is a constant dependent on G.

- We manage to prove the first half of the conjecture, i.e., $W_1(p, q) \leq K(p - q; \infty)$, but it is still a mystery about the explicit formulation of the constant C in second half.

The Cost Conjecture

- As time $T \to \infty$, one might expect the TSDM cost to be close to the optimal transport cost (i.e., the 1^{st} Wasserstein distance) between any two vector measures defined on the graph.

Conjecture

Given any two probability distributions p, q on a connected graph $G = (V, E, W)$ with graph geodesic distance metric $d : V \times V \to \mathbb{R}_{\geq 0}$,

$$W_1(p,q) \leq K(p-q;\infty) \leq C \cdot W_1(p,q)$$

in which $W_1(p,q) := \inf_{\gamma \in \Gamma(p,q)} \int_{V \times V} d(x,y)\,\mathrm{d}\gamma(x,y)$, where $\Gamma(p,q)$ denotes the collection of all measures on $V \times V$ with marginals p and q in the first and second factors respectively and C is a constant dependent on G.

- We manage to prove the first half of the conjecture, i.e., $W_1(p,q) \leq K(p-q;\infty)$, but it is still a mystery about the explicit formulation of the constant C in second half.

TSDM ($T = 0.1$) on the Dendritic Tree

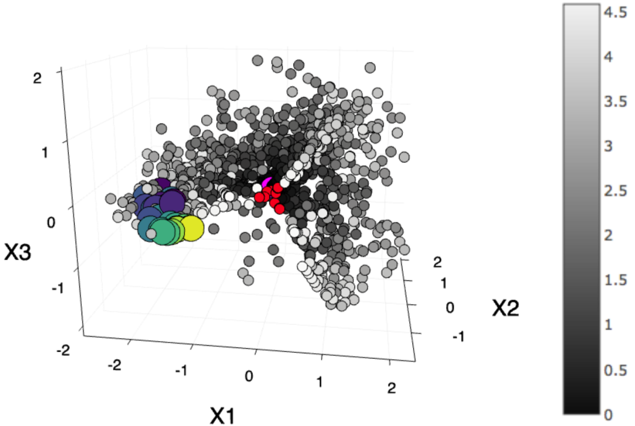


Figure: The magenta circle = the DC vector; the cyan circle = the Fiedler vector; the red circles = the localized eigenvectors; the larger colored circles = the eigenvectors supported on the upper-left branch

Outline

Natural Graph Wavelet Packet Dictionary

There may be a number of different ways to group and organize the graph Laplacian eigenvectors once the mutual distances are computed. Below, we discuss one of the simplest ones.

1. Given a graph $G = \{V, E, W\}$ with $|V| = n$ and the distance matrix $D = (D_{ij})$ of its eigenvectors, construct a *dual graph* $G^\star = \{V^\star, E^\star, W^\star\}$ where the ith node in $V^\star$ represents $\boldsymbol{\phi}_i$, and the edge weight $W^\star_{ij}$ reflects the affinity between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$, e.g., $W^\star_{ij} = 1/D_{ij}$ or $\exp(-D^2_{ij}/\sigma^2)$ for some appropriate scale parameter σ.
2. Construct a *hierarchical partition tree* of $G^\star$ using, e.g., the recursive bi-partition method that was used to construct our other graph wavelet packets such as the Hierarchical Graph Laplacian Eigen Transform (HGLET) and the Generalized Haar-Walsh Transform (GHWT). This corresponds to *hierarchical partitioning of the frequency domain in the conventional time-frequency analysis, which generates classical wavelet packets*.

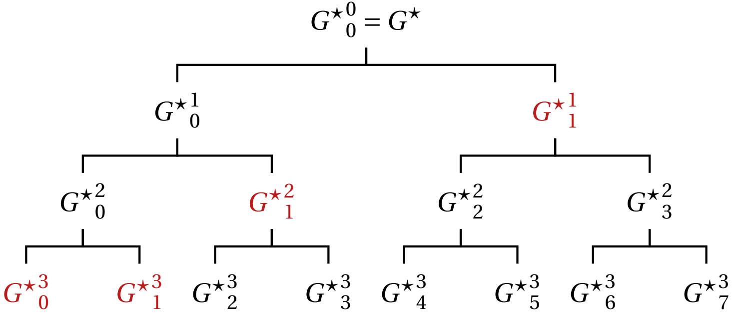


Figure: A binary partition tree of the dual graph $G^{\star}$

3. The graph wavelet packet vectors in $G^{\star j}_{k}$ can be generated as follows:

$$\boldsymbol{\psi}^{j}_{k,l} = \Phi F^{j}_{k} \Phi^{\top} \boldsymbol{e}_{l} \quad \text{for } j = 0 : j_{\max},\ k = 0 : 2^{j} - 1,\ l = 0 : n - 1$$

in which, the diagonal matrix $F^{j}_{k} \in \mathbb{R}^{n \times n}$ with $F^{j}_{k}[l, l] = \chi_{V^{\star j}_{k}}(l)$, $l = 0 : n - 1$, which selects the eigenvectors corresponding $V^{\star j}_{k}$, Φ is the eigenvector matrix, and $\boldsymbol{e}_{l}$ is the canonical basis vector at the lth vertex.

An Ideal Case: Shannon Wavelets from DCT

- We can generate *Shannon wavelets* from the graph Laplacian eigenvectors of a 1D path (i.e., the DCT-II basis vectors) by simply setting $F_0^j = \mathrm{diag}(\mathbf{1}_{n/2^j}, \mathbf{0}_{n-n/2^j})$; $F_1^j = \mathrm{diag}(\mathbf{0}_{n/2^j}, \mathbf{1}_{n/2^j}, \mathbf{0}_{n-n/2^{j-1}})$, and computing $\boldsymbol{\phi}_l^j = \Phi F_0^j \Phi^{\mathsf{T}} \boldsymbol{e}_l$ (father); $\boldsymbol{\psi}_l^j = \Phi F_1^j \Phi^{\mathsf{T}} \boldsymbol{e}_l$ (mother).

- Can generate smoother wavelets (e.g., Meyer wavelets) by using *smoother partition of unity* in the diagonals of F_*^j's

An Ideal Case: Shannon Wavelets from DCT

- We can generate *Shannon wavelets* from the graph Laplacian eigenvectors of a 1D path (i.e., the DCT-II basis vectors) by simply setting $F_0^j = \mathrm{diag}(\mathbf{1}_{n/2^j}, \mathbf{0}_{n-n/2^j})$; $F_1^j = \mathrm{diag}(\mathbf{0}_{n/2^j}, \mathbf{1}_{n/2^j}, \mathbf{0}_{n-n/2^{j-1}})$, and computing $\boldsymbol{\phi}_l^j = \Phi F_0^j \Phi^\top \boldsymbol{e}_l$ (father); $\boldsymbol{\psi}_l^j = \Phi F_1^j \Phi^\top \boldsymbol{e}_l$ (mother).

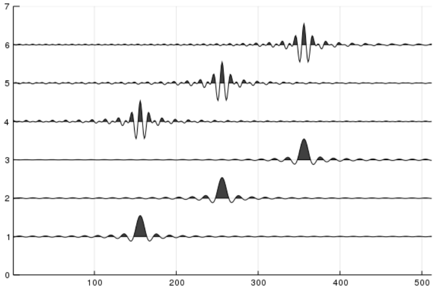


Figure: From DCT to Shannon wavelets ($j = 3$)

- Can generate smoother wavelets (e.g., Meyer wavelets) by using *smoother partition of unity* in the diagonals of F_*^j's

An Ideal Case: Shannon Wavelets from DCT

- We can generate *Shannon wavelets* from the graph Laplacian eigenvectors of a 1D path (i.e., the DCT-II basis vectors) by simply setting $F_0^j = \mathrm{diag}(\mathbf{1}_{n/2^j}, \mathbf{0}_{n-n/2^j})$; $F_1^j = \mathrm{diag}(\mathbf{0}_{n/2^j}, \mathbf{1}_{n/2^j}, \mathbf{0}_{n-n/2^{j-1}})$, and computing $\boldsymbol{\phi}_l^j = \Phi F_0^j \Phi^\top \boldsymbol{e}_l$ (father); $\boldsymbol{\psi}_l^j = \Phi F_1^j \Phi^\top \boldsymbol{e}_l$ (mother).

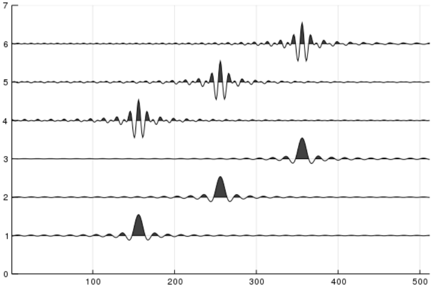
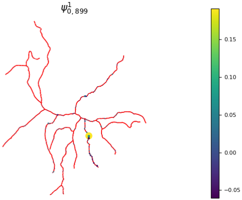


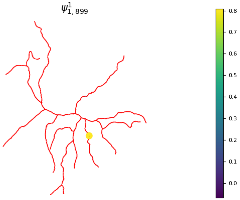
Figure: From DCT to Shannon wavelets ($j = 3$)

- Can generate smoother wavelets (e.g., Meyer wavelets) by using *smoother partition of unity* in the diagonals of F_*^j's

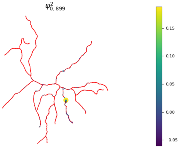
Some wavelet packet vectors on the RGC dendritic tree



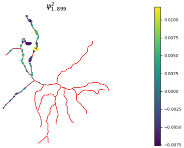
(a) $\psi^1_{0,899}$



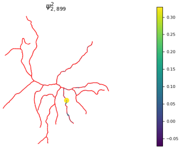
(b) $\psi^1_{1,899}$



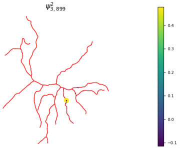
(c) $\psi^2_{0,899}$



(d) $\psi^2_{1,899}$



(e) $\psi^2_{2,899}$



(f) $\psi^2_{3,899}$

Natural Graph Wavelet Basis

- Obviously, the above natural graph wavelet packet dictionary are hugely redundant, containing approximately $n(2n-1)$ basis vectors.
- Constructing a standard wavelet packet dictionary with $n(1+\log_2 n)$ basis vectors, we only need a subset of $\{\boldsymbol{e}_l\}_{l=0:n-1}$ so that the number of basis vectors to generate on $G^{\star j}_{k}$ is $|V^{\star j}_{k}|$ (if $n = 2^{j_{\max}}$ with the perfectly balanced binary tree, $|V^{\star j}_{k}| = 2^{j_{\max}-j}$ where $j_{\max} = \log_2 n$).
- One possibility is to pick the appropriate number (i.e., $|V^{\star j}_{k}|$) of the original nodes in V using the energy concentration of the eigenvectors in $G^{\star j}_{k}$ and use those nodes for $\{\boldsymbol{e}_l\}$ (this is a promising new idea proposed by Haotian Li).
- Once this is done, one can apply the *best-basis selection algorithm* of Coifman-Wickerhauser or its variants by the Saito group to choose the most suitable basis for a given task (e.g., efficient approximation, denoising, classification, regression, etc.). Note that the best-basis algorithm searches the best one among more than $(1.5)^n$ possible ONBs from the wavelet packet dictionary.

Natural Graph Wavelet Basis

- Obviously, the above natural graph wavelet packet dictionary are hugely redundant, containing approximately $n(2n-1)$ basis vectors.
- Constructing a standard wavelet packet dictionary with $n(1+\log_2 n)$ basis vectors, we only need a subset of $\{\boldsymbol{e}_l\}_{l=0:n-1}$ so that the number of basis vectors to generate on $G^{\star}{}_k^j$ is $|V^{\star}{}_k^j|$ (if $n = 2^{j_{\max}}$ with the perfectly balanced binary tree, $|V^{\star}{}_k^j| = 2^{j_{\max}-j}$ where $j_{\max} = \log_2 n$).
- One possibility is to pick the appropriate number (i.e., $|V^{\star}{}_k^j|$) of the original nodes in V using the energy concentration of the eigenvectors in $G^{\star}{}_k^j$ and use those nodes for $\{\boldsymbol{e}_l\}$ (this is a promising new idea proposed by Haotian Li).
- Once this is done, one can apply the *best-basis selection algorithm* of Coifman-Wickerhauser or its variants by the Saito group to choose the most suitable basis for a given task (e.g., efficient approximation, denoising, classification, regression, etc.). Note that the best-basis algorithm searches the best one among more than $(1.5)^n$ possible ONBs from the wavelet packet dictionary.

Natural Graph Wavelet Basis

- Obviously, the above natural graph wavelet packet dictionary are hugely redundant, containing approximately $n(2n-1)$ basis vectors.
- Constructing a standard wavelet packet dictionary with $n(1+\log_2 n)$ basis vectors, we only need a subset of $\{\boldsymbol{e}_l\}_{l=0:n-1}$ so that the number of basis vectors to generate on ${G^\star}_k^j$ is $|{V^\star}_k^j|$ (if $n = 2^{j_{\max}}$ with the perfectly balanced binary tree, $|{V^\star}_k^j| = 2^{j_{\max}-j}$ where $j_{\max} = \log_2 n$).
- One possibility is to pick the appropriate number (i.e., $|{V^\star}_k^j|$) of the original nodes in V using the energy concentration of the eigenvectors in ${G^\star}_k^j$ and use those nodes for $\{\boldsymbol{e}_l\}$ (this is a promising new idea proposed by Haotian Li).
- Once this is done, one can apply the *best-basis selection algorithm* of Coifman-Wickerhauser or its variants by the Saito group to choose the most suitable basis for a given task (e.g., efficient approximation, denoising, classification, regression, etc.). Note that the best-basis algorithm searches the best one among more than $(1.5)^n$ possible ONBs from the wavelet packet dictionary.

Natural Graph Wavelet Basis

- Obviously, the above natural graph wavelet packet dictionary are hugely redundant, containing approximately $n(2n-1)$ basis vectors.
- Constructing a standard wavelet packet dictionary with $n(1+\log_2 n)$ basis vectors, we only need a subset of $\{\boldsymbol{e}_l\}_{l=0:n-1}$ so that the number of basis vectors to generate on $G^{\star j}_{k}$ is $|V^{\star j}_{k}|$ (if $n=2^{j_{\max}}$ with the perfectly balanced binary tree, $|V^{\star j}_{k}| = 2^{j_{\max}-j}$ where $j_{\max} = \log_2 n$).
- One possibility is to pick the appropriate number (i.e., $|V^{\star j}_{k}|$) of the original nodes in V using the energy concentration of the eigenvectors in $G^{\star j}_{k}$ and use those nodes for $\{\boldsymbol{e}_l\}$ (this is a promising new idea proposed by Haotian Li).
- Once this is done, one can apply the *best-basis selection algorithm* of Coifman-Wickerhauser or its variants by the Saito group to choose the most suitable basis for a given task (e.g., efficient approximation, denoising, classification, regression, etc.). Note that the best-basis algorithm searches the best one among more than $(1.5)^n$ possible ONBs from the wavelet packet dictionary.

Outline

Summary and Future Projects

- Found a *natural* method to order graph Laplacian eigenvectors $\{\boldsymbol{\phi}_i\}_{i=0:n-1}$ using the transportation cost as their mutual distances based on the ROT theory on a fixed graph
- How to examine all possible solutions of

$$\tilde{\tilde{Q}} \boldsymbol{w}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i, \quad \boldsymbol{w}_{ij} \in \mathbb{R}^{2m}_{\geq 0}. \qquad (*)$$

  and find the true cost minimizing transportation plan?
- How to find the *sparsest* nonnegative solution of Eqn. (*) ?
- How to select the best $\alpha \in [0,1]$ for $M_\alpha(G) := \sum_{e \in E(G)} w(e)^\alpha \, \mathrm{length}(e)$?
- Which way should we turn $\boldsymbol{\phi}_i$ into $\boldsymbol{p}_i$?
- How to choose a good metric among several possibilities (ROT; TSD; DAG; ...)?
- How to improve the hierarchical partitioning of the dual graph without forcing recursive binary partitions?

Summary and Future Projects

- Found a *natural* method to order graph Laplacian eigenvectors $\{\boldsymbol{\phi}_i\}_{i=0:n-1}$ using the transportation cost as their mutual distances based on the ROT theory on a fixed graph
- How to examine all possible solutions of

$$\tilde{\tilde{Q}} \boldsymbol{w}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i, \quad \boldsymbol{w}_{ij} \in \mathbb{R}^{2m}_{\geq 0}. \qquad (*)$$

  and find the true cost minimizing transportation plan?
- How to find the *sparsest* nonnegative solution of Eqn. (*) ?
- How to select the best $\alpha \in [0, 1]$ for $M_\alpha(G) := \sum_{e \in E(G)} w(e)^\alpha \, \mathrm{length}(e)$?
- Which way should we turn $\boldsymbol{\phi}_i$ into $\boldsymbol{p}_i$?
- How to choose a good metric among several possibilities (ROT; TSD; DAG; ...)?
- How to improve the hierarchical partitioning of the dual graph without forcing recursive binary partitions?

Summary and Future Projects

- Found a *natural* method to order graph Laplacian eigenvectors $\{\boldsymbol{\phi}_i\}_{i=0:n-1}$ using the transportation cost as their mutual distances based on the ROT theory on a fixed graph
- How to examine all possible solutions of

$$\tilde{\tilde{Q}}\boldsymbol{w}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i, \quad \boldsymbol{w}_{ij} \in \mathbb{R}^{2m}_{\geq 0}. \qquad (*)$$

  and find the true cost minimizing transportation plan?
- How to find the *sparsest* nonnegative solution of Eqn. (*) ?
- How to select the best $\alpha \in [0,1]$ for $M_\alpha(G) := \sum_{e \in E(G)} w(e)^\alpha \, \mathrm{length}(e)$?
- Which way should we turn $\boldsymbol{\phi}_i$ into $\boldsymbol{p}_i$?
- How to choose a good metric among several possibilities (ROT; TSD; DAG; ...)?
- How to improve the hierarchical partitioning of the dual graph without forcing recursive binary partitions?

Summary and Future Projects

- Found a *natural* method to order graph Laplacian eigenvectors $\{\boldsymbol{\phi}_i\}_{i=0:n-1}$ using the transportation cost as their mutual distances based on the ROT theory on a fixed graph
- How to examine all possible solutions of

$$\tilde{\tilde{Q}}\boldsymbol{w}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i, \quad \boldsymbol{w}_{ij} \in \mathbb{R}^{2m}_{\geq 0}. \qquad (*)$$

  and find the true cost minimizing transportation plan?
- How to find the *sparsest* nonnegative solution of Eqn. (*) ?
- How to select the best $\alpha \in [0,1]$ for $\boldsymbol{M}_\alpha(G) := \sum_{e \in E(G)} w(e)^\alpha \,\mathrm{length}(e)$?
- Which way should we turn $\boldsymbol{\phi}_i$ into $\boldsymbol{p}_i$?
- How to choose a good metric among several possibilities (ROT; TSD; DAG; ...)?
- How to improve the hierarchical partitioning of the dual graph without forcing recursive binary partitions?

Summary and Future Projects

- Found a *natural* method to order graph Laplacian eigenvectors $\{\boldsymbol{\phi}_i\}_{i=0:n-1}$ using the transportation cost as their mutual distances based on the ROT theory on a fixed graph
- How to examine all possible solutions of

$$\tilde{\tilde{Q}}\boldsymbol{w}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i, \quad \boldsymbol{w}_{ij} \in \mathbb{R}^{2m}_{\geq 0}. \qquad (*)$$

  and find the true cost minimizing transportation plan?
- How to find the *sparsest* nonnegative solution of Eqn. (*) ?
- How to select the best $\alpha \in [0,1]$ for $\boldsymbol{M}_\alpha(G) := \sum_{e \in E(G)} w(e)^\alpha \,\mathrm{length}(e)$?
- Which way should we turn $\boldsymbol{\phi}_i$ into $\boldsymbol{p}_i$?
- How to choose a good metric among several possibilities (ROT; TSD; DAG; ...)?
- How to improve the hierarchical partitioning of the dual graph without forcing recursive binary partitions?

Summary and Future Projects

- Found a *natural* method to order graph Laplacian eigenvectors $\{\boldsymbol{\phi}_i\}_{i=0:n-1}$ using the transportation cost as their mutual distances based on the ROT theory on a fixed graph
- How to examine all possible solutions of

$$\tilde{\tilde{Q}}\boldsymbol{w}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i, \quad \boldsymbol{w}_{ij} \in \mathbb{R}^{2m}_{\geq 0}. \qquad (*)$$

  and find the true cost minimizing transportation plan?
- How to find the *sparsest* nonnegative solution of Eqn. (*) ?
- How to select the best $\alpha \in [0,1]$ for $\boldsymbol{M}_\alpha(G) := \sum_{e \in E(G)} w(e)^\alpha \, \mathrm{length}(e)$?
- Which way should we turn $\boldsymbol{\phi}_i$ into $\boldsymbol{p}_i$?
- How to choose a good metric among several possibilities (ROT; TSD; DAG; ...)?
- How to improve the hierarchical partitioning of the dual graph without forcing recursive binary partitions?

Summary and Future Projects

- Found a *natural* method to order graph Laplacian eigenvectors $\{\boldsymbol{\phi}_i\}_{i=0:n-1}$ using the transportation cost as their mutual distances based on the ROT theory on a fixed graph
- How to examine all possible solutions of

$$\tilde{\tilde{Q}}\boldsymbol{w}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i, \quad \boldsymbol{w}_{ij} \in \mathbb{R}^{2m}_{\geq 0}. \qquad (*)$$

  and find the true cost minimizing transportation plan?
- How to find the *sparsest* nonnegative solution of Eqn. (*) ?
- How to select the best $\alpha \in [0,1]$ for $\boldsymbol{M}_\alpha(G) := \sum_{e \in E(G)} w(e)^\alpha \,\mathrm{length}(e)$?
- Which way should we turn $\boldsymbol{\phi}_i$ into $\boldsymbol{p}_i$?
- How to choose a good metric among several possibilities (ROT; TSD; DAG; . . .)?
- How to improve the hierarchical partitioning of the dual graph without forcing recursive binary partitions?

References

- N. Saito & E. Woei: "On the phase transition phenomenon of graph Laplacian eigenfunctions on trees," *RIMS Kôkyûroku*, vol. 1743, pp. 77–90, 2011.
- Y. Nakatsukasa, N. Saito, & E. Woei: "Mysteries around graph Laplacian eigenvalue 4," *Linear Algebra Appl.*, vol. 438, no. 8, pp. 3231–3246, 2013.
- J. Irion & N. Saito: "The generalized Haar-Walsh transform," in *Proc. 2014 IEEE Workshop on Statistical Signal Processing*, pp. 472–475, 2014.
- J. Irion & N. Saito: "Applied and computational harmonic analysis on graphs and networks," in *Wavelets and Sparsity XVI, Proc. SPIE 9597*, Paper # 95971F, 2015.
- J. Irion & N. Saito: "Efficient approximation and denoising of graph signals using the multiscale basis dictionaries," *IEEE Trans. Signal and Inform. Process. Netw.*, vol. 3, no. 3, pp. 607–616, 2017.
- N. Saito: "How can we naturally order and organize graph Laplacian eigenvectors?" in *Proc. 2018 IEEE Workshop on Statistical Signal Processing*, pp. 483–487, 2018. Also arXiv: 1801.06782 [math.SP].
- H. Li & N. Saito: "Metrics of graph Laplacian eigenvectors," in *Wavelets and Sparsity XVIII, Proc. SPIE 11138*, 2019, to appear.
- A. Cloninger, H. Li, and N. Saito: "Natural graph wavelet packets," in preparation, 2019.