Analysis of Neuronal Dendrite Patterns Using Graph Laplacians

Naoki Saito

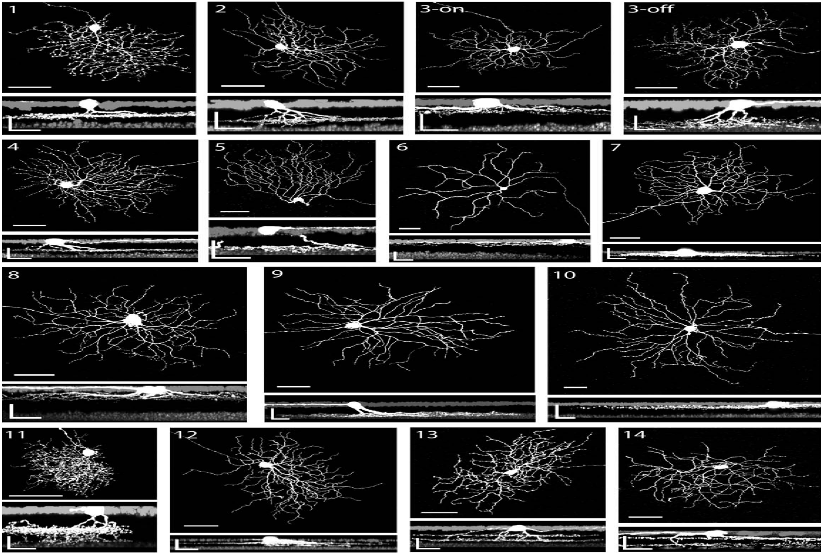
Department of Mathematics
University of California, Davis

Feb. 9, 2009

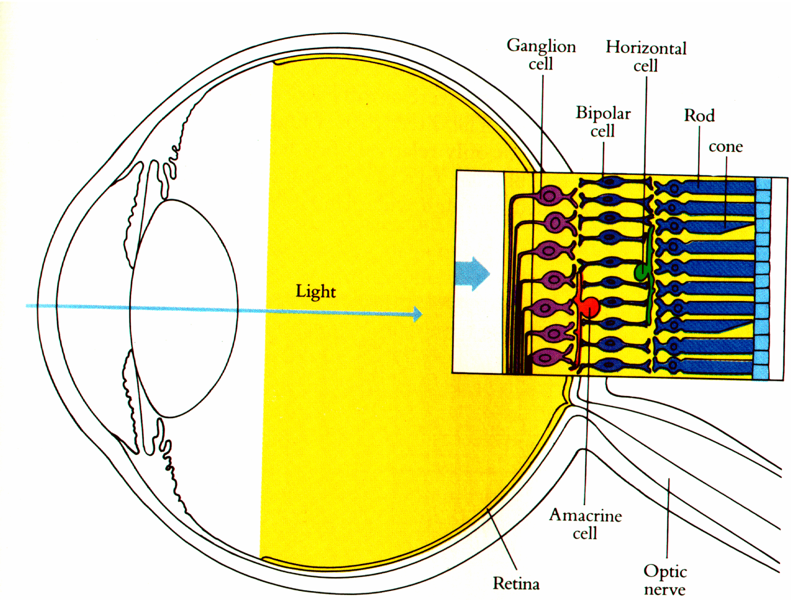
Outline

Outline

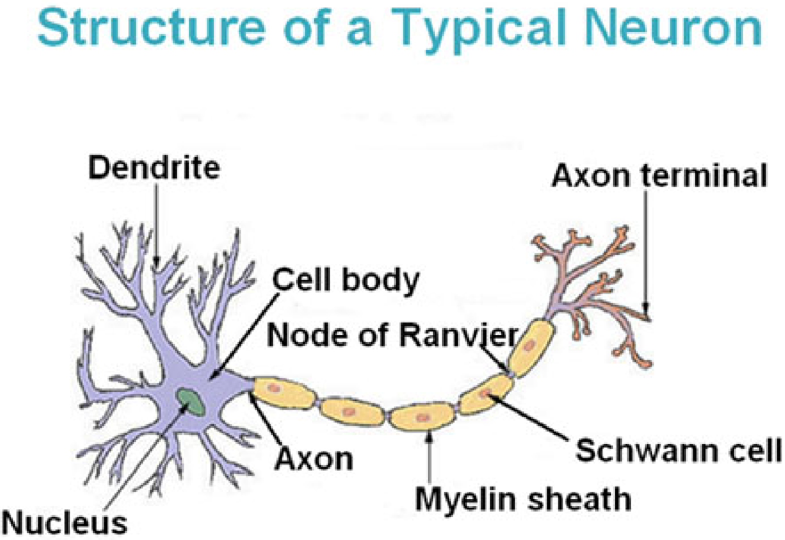
Clustering Mouse Retinal Ganglion Cells . . . 3D Data



Cells in Retina (from Hubel: Eye, Brain, and Vision, 1995)



A Typical Neuron (from Wikipedia)



Clustering Mouse's Retinal Ganglion Cells

- Neuroscientists' Objective: To understand how structural / morphological properties of mouse retinal ganglion cells (RGCs) relate to the cell types and their functionality; how such properties change / evolve from newborn to adult
- Why mouse? $\Longrightarrow$ Great possibilities for genetic manipulation
- Data: 3D images of dendrites of RGCs via a confocal microscope
- State of the art: A manually intensive procedure using specialized software[1]:
  - Trace and segment dendrite patterns from each 3D cube;
  - Extract geometric/morphological parameters (totally 14 such parameters);
  - Apply a conventional bottom-up "hierarchical clustering" algorithm
- The extracted morphological parameters include: somal size; dendritic field size; total dendrite length; branch order; mean internal branch length; branch angle; mean terminal branch length, ...
- It takes half a day per cell with a lot of human interactions!

[1] Neurolucida®, MBF Bioscience

Clustering Mouse's Retinal Ganglion Cells

- Neuroscientists' Objective: To understand how structural / morphological properties of mouse retinal ganglion cells (RGCs) relate to the cell types and their functionality; how such properties change / evolve from newborn to adult
- Why mouse? $\Longrightarrow$ Great possibilities for genetic manipulation
- Data: 3D images of dendrites of RGCs via a confocal microscope
- State of the art: A manually intensive procedure using specialized software[1]:
  - Trace and segment dendrite patterns from each 3D cube;
  - Extract geometric/morphological parameters (totally 14 such parameters);
  - Apply a conventional bottom-up "hierarchical clustering" algorithm
- The extracted morphological parameters include: somal size; dendritic field size; total dendrite length; branch order; mean internal branch length; branch angle; mean terminal branch length, . . .
- It takes half a day per cell with a lot of human interactions!

[1] Neurolucida®, MBF Bioscience

Clustering Mouse's Retinal Ganglion Cells

- Neuroscientists' Objective: To understand how structural / morphological properties of mouse retinal ganglion cells (RGCs) relate to the cell types and their functionality; how such properties change / evolve from newborn to adult
- Why mouse? $\Longrightarrow$ Great possibilities for genetic manipulation
- Data: 3D images of dendrites of RGCs via a confocal microscope
- State of the art: A manually intensive procedure using specialized software[1]:
  - Trace and segment dendrite patterns from each 3D cube;
  - Extract geometric/morphological parameters (totally 14 such parameters);
  - Apply a conventional bottom-up "hierarchical clustering" algorithm
- The extracted morphological parameters include: somal size; dendritic field size; total dendrite length; branch order; mean internal branch length; branch angle; mean terminal branch length, ...
- It takes half a day per cell with a lot of human interactions!

[1] Neurolucida®, MBF Bioscience

Clustering Mouse's Retinal Ganglion Cells

- Neuroscientists' Objective: To understand how structural / morphological properties of mouse retinal ganglion cells (RGCs) relate to the cell types and their functionality; how such properties change / evolve from newborn to adult
- Why mouse? $\Longrightarrow$ Great possibilities for genetic manipulation
- Data: 3D images of dendrites of RGCs via a confocal microscope
- State of the art: A manually intensive procedure using specialized software[1]:
  - Trace and segment dendrite patterns from each 3D cube;
  - Extract geometric/morphological parameters (totally 14 such parameters);
  - Apply a conventional bottom-up "hierarchical clustering" algorithm
- The extracted morphological parameters include: somal size; dendritic field size; total dendrite length; branch order; mean internal branch length; branch angle; mean terminal branch length, . . .
- It takes half a day per cell with a lot of human interactions!

[1] Neurolucida®, MBF Bioscience

Clustering Mouse's Retinal Ganglion Cells

- Neuroscientists' Objective: To understand how structural / morphological properties of mouse retinal ganglion cells (RGCs) relate to the cell types and their functionality; how such properties change / evolve from newborn to adult
- Why mouse? $\Longrightarrow$ Great possibilities for genetic manipulation
- Data: 3D images of dendrites of RGCs via a confocal microscope
- State of the art: A manually intensive procedure using specialized software[1]:
    - Trace and segment dendrite patterns from each 3D cube;
    - Extract geometric/morphological parameters (totally 14 such parameters);
    - Apply a conventional bottom-up "hierarchical clustering" algorithm
- The extracted morphological parameters include: somal size; dendritic field size; total dendrite length; branch order; mean internal branch length; branch angle; mean terminal branch length, ...
- It takes half a day per cell with a lot of human interactions!

[1] Neurolucida®, MBF Bioscience

Clustering Mouse's Retinal Ganglion Cells

- Neuroscientists' Objective: To understand how structural / morphological properties of mouse retinal ganglion cells (RGCs) relate to the cell types and their functionality; how such properties change / evolve from newborn to adult
- Why mouse? $\Longrightarrow$ Great possibilities for genetic manipulation
- Data: 3D images of dendrites of RGCs via a confocal microscope
- State of the art: A manually intensive procedure using specialized software[1]:
  - Trace and segment dendrite patterns from each 3D cube;
  - Extract geometric/morphological parameters (totally 14 such parameters);
  - Apply a conventional bottom-up "hierarchical clustering" algorithm
- The extracted morphological parameters include: somal size; dendritic field size; total dendrite length; branch order; mean internal branch length; branch angle; mean terminal branch length, . . .
- It takes half a day per cell with a lot of human interactions!

[1] Neurolucida®, MBF Bioscience

Clustering Mouse's Retinal Ganglion Cells

- Neuroscientists' Objective: To understand how structural / morphological properties of mouse retinal ganglion cells (RGCs) relate to the cell types and their functionality; how such properties change / evolve from newborn to adult
- Why mouse? $\Longrightarrow$ Great possibilities for genetic manipulation
- Data: 3D images of dendrites of RGCs via a confocal microscope
- State of the art: A manually intensive procedure using specialized software[1]:
  - Trace and segment dendrite patterns from each 3D cube;
  - Extract geometric/morphological parameters (totally 14 such parameters);
  - Apply a conventional bottom-up "hierarchical clustering" algorithm
- The extracted morphological parameters include: somal size; dendritic field size; total dendrite length; branch order; mean internal branch length; branch angle; mean terminal branch length, ...
- It takes half a day per cell with a lot of human interactions!

[1] Neurolucida®, MBF Bioscience

Clustering Mouse's Retinal Ganglion Cells

- Neuroscientists' Objective: To understand how structural / morphological properties of mouse retinal ganglion cells (RGCs) relate to the cell types and their functionality; how such properties change / evolve from newborn to adult
- Why mouse? $\Longrightarrow$ Great possibilities for genetic manipulation
- Data: 3D images of dendrites of RGCs via a confocal microscope
- State of the art: A manually intensive procedure using specialized software[1]:
  - Trace and segment dendrite patterns from each 3D cube;
  - Extract geometric/morphological parameters (totally 14 such parameters);
  - Apply a conventional bottom-up "hierarchical clustering" algorithm
- The extracted morphological parameters include: somal size; dendritic field size; total dendrite length; branch order; mean internal branch length; branch angle; mean terminal branch length, ...
- It takes half a day per cell with a lot of human interactions!

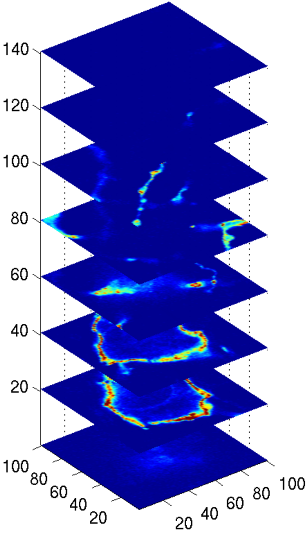
[1] Neurolucida®, MBF Bioscience

Clustering Mouse's Retinal Ganglion Cells

- Neuroscientists' Objective: To understand how structural / morphological properties of mouse retinal ganglion cells (RGCs) relate to the cell types and their functionality; how such properties change / evolve from newborn to adult
- Why mouse? $\Longrightarrow$ Great possibilities for genetic manipulation
- Data: 3D images of dendrites of RGCs via a confocal microscope
- State of the art: A manually intensive procedure using specialized software[1]:
    - Trace and segment dendrite patterns from each 3D cube;
    - Extract geometric/morphological parameters (totally 14 such parameters);
    - Apply a conventional bottom-up "hierarchical clustering" algorithm
- The extracted morphological parameters include: somal size; dendritic field size; total dendrite length; branch order; mean internal branch length; branch angle; mean terminal branch length, ...
- It takes half a day per cell with a lot of human interactions!

[1] Neurolucida®, MBF Bioscience

3D Data



Our Goal

Long-term: Develop an efficient and automatic procedure from segmentation/tracing to morphological parameter extraction to clustering and classification to assist neuroscientists

Segmentation/tracing is a tough but high-return project
$\Longrightarrow$ Tractography in Diffusion Tensor MRI, ...

Short-term: Develop algorithms for automatic morphological feature extraction and clustering

Our Goal

Long-term: Develop an efficient and automatic procedure from segmentation/tracing to morphological parameter extraction to clustering and classification to assist neuroscientists

Segmentation/tracing is a tough but high-return project
$\Longrightarrow$ Tractography in Diffusion Tensor MRI, . . .

Short-term: Develop algorithms for automatic morphological feature extraction and clustering

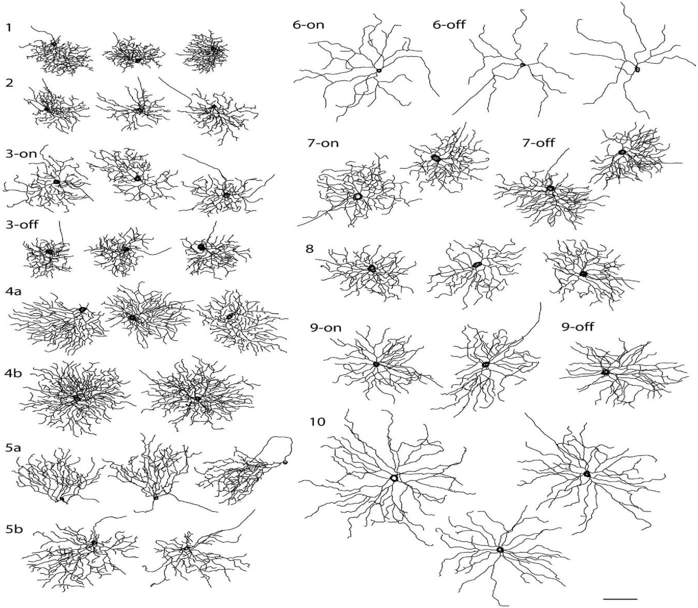
Our Goal

Long-term: Develop an efficient and automatic procedure from segmentation/tracing to morphological parameter extraction to clustering and classification to assist neuroscientists

Segmentation/tracing is a tough but high-return project
$\Longrightarrow$ Tractography in Diffusion Tensor MRI, . . .

Short-term: Develop algorithms for automatic morphological feature extraction and clustering

Clustering using Features Derived by Neurolucida®



Outline

Our Dataset

consists of 130 RGCs each of which in turn consists of

- A sequence of 3D sample points along dendrite arbors obtained by Neurolucida® (requires intensive human interaction)
- Connectivity and branching information by the same software
- Each soma is represented as a sequence of points traced along its boundary (circular/ring shape)

Our Dataset

consists of 130 RGCs each of which in turn consists of

- A sequence of 3D sample points along dendrite arbors obtained by Neurolucida® (requires intensive human interaction)
- Connectivity and branching information by the same software
- Each soma is represented as a sequence of points traced along its boundary (circular/ring shape)

Our Dataset

consists of 130 RGCs each of which in turn consists of

- A sequence of 3D sample points along dendrite arbors obtained by Neurolucida® (requires intensive human interaction)
- Connectivity and branching information by the same software
- Each soma is represented as a sequence of points traced along its boundary (circular/ring shape)

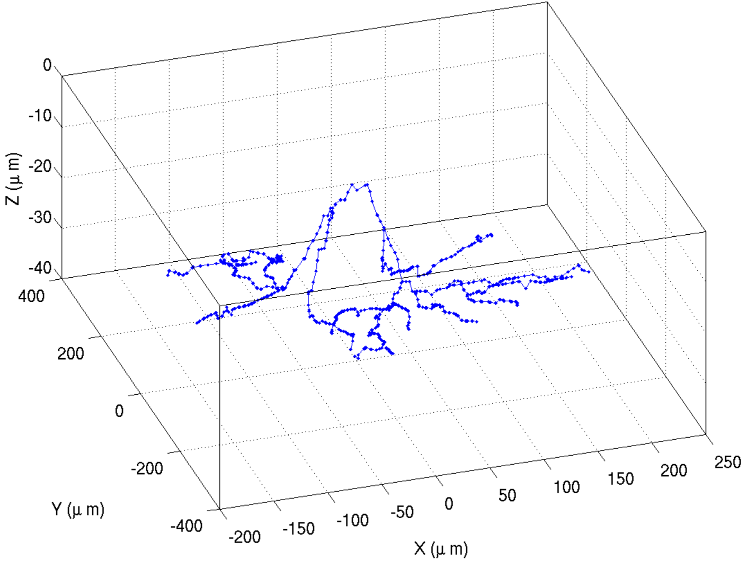
Our Dataset

consists of 130 RGCs each of which in turn consists of

- A sequence of 3D sample points along dendrite arbors obtained by Neurolucida® (requires intensive human interaction)
- Connectivity and branching information by the same software
- Each soma is represented as a sequence of points traced along its boundary (circular/ring shape)

$\Longrightarrow$ Constructing a graph representing dendrite structures per RGC is very natural and simple! In fact, we constructed a tree (i.e., a connected graph without cycles/loops) by replacing the soma ring by a single vertex representing a center of the soma.

Our Dataset ⟹ Trees



Our Dataset $\Longrightarrow$ Trees ...

- Let G be a graph (in fact a tree) representing an RGC.
- Let $V = V(G) = \{v_1, \ldots, v_n\}$ where $v_k \in \mathbb{R}^3$, be a set of vertices representing sample points along dendrite arbors. n ranges between 565 and 24474 depending on the RGCs.
- Let $E = E(G) = \{e_1, \ldots, e_m\}$ be a set of edges where $e_k = (v_i, v_j)$ represents an edge (or line segment) connecting between adjacent vertices v_i, v_j for some $1 \le i, j \le n$. Note that $|E(G)| = |V(G)| - 1$ since G is a tree.
- Let $d(v_k) = d_{v_k}$ be the degree of the vertex v_k. In our dataset,

$$\max_{130 \text{ cells}} \max_k d(v_k) = 8, \qquad \min_{130 \text{ cells}} \max_k d(v_k) = 3.$$

- In principle, we should consider the weighted graph with weights $w_{e_k} := \|v_i - v_j\|^{-1}$. But for simplicity, we only consider the unweighted graphs/trees here. (One could justify this by resampling the dendrite coordinates with a uniform sampling rate.)

Our Dataset $\Longrightarrow$ Trees . . .

- Let G be a graph (in fact a tree) representing an RGC.
- Let $V = V(G) = \{v_1, \dots, v_n\}$ where $v_k \in \mathbb{R}^3$, be a set of vertices representing sample points along dendrite arbors. n ranges between 565 and 24474 depending on the RGCs.
- Let $E = E(G) = \{e_1, \dots, e_m\}$ be a set of edges where $e_k = (v_i, v_j)$ represents an edge (or line segment) connecting between adjacent vertices v_i, v_j for some $1 \le i, j \le n$. Note that $|E(G)| = |V(G)| - 1$ since G is a tree.
- Let $d(v_k) = d_{v_k}$ be the degree of the vertex v_k. In our dataset,

$$\max_{130 \text{ cells}} \max_k d(v_k) = 8, \qquad \min_{130 \text{ cells}} \max_k d(v_k) = 3.$$

- In principle, we should consider the weighted graph with weights $w_{e_k} := \|v_i - v_j\|^{-1}$. But for simplicity, we only consider the unweighted graphs/trees here. (One could justify this by resampling the dendrite coordinates with a uniform sampling rate.)

Our Dataset $\Longrightarrow$ Trees . . .

- Let G be a graph (in fact a tree) representing an RGC.
- Let $V = V(G) = \{v_1, \ldots, v_n\}$ where $v_k \in \mathbb{R}^3$, be a set of vertices representing sample points along dendrite arbors. n ranges between 565 and 24474 depending on the RGCs.
- Let $E = E(G) = \{e_1, \ldots, e_m\}$ be a set of edges where $e_k = (v_i, v_j)$ represents an edge (or line segment) connecting between adjacent vertices v_i, v_j for some $1 \le i, j \le n$. Note that $|E(G)| = |V(G)| - 1$ since G is a tree.
- Let $d(v_k) = d_{v_k}$ be the degree of the vertex v_k. In our dataset,

$$\max_{130 \text{ cells}} \max_k d(v_k) = 8, \qquad \min_{130 \text{ cells}} \max_k d(v_k) = 3.$$

- In principle, we should consider the weighted graph with weights $w_{e_k} := \|v_i - v_j\|^{-1}$. But for simplicity, we only consider the unweighted graphs/trees here. (One could justify this by resampling the dendrite coordinates with a uniform sampling rate.)

Our Dataset $\Longrightarrow$ Trees . . .

- Let G be a graph (in fact a tree) representing an RGC.
- Let $V = V(G) = \{v_1, \ldots, v_n\}$ where $v_k \in \mathbb{R}^3$, be a set of vertices representing sample points along dendrite arbors. n ranges between 565 and 24474 depending on the RGCs.
- Let $E = E(G) = \{e_1, \ldots, e_m\}$ be a set of edges where $e_k = (v_i, v_j)$ represents an edge (or line segment) connecting between adjacent vertices v_i, v_j for some $1 \leq i, j \leq n$. Note that $|E(G)| = |V(G)| - 1$ since G is a tree.
- Let $d(v_k) = d_{v_k}$ be the degree of the vertex v_k. In our dataset,

$$\max_{130 \text{ cells}} \max_k d(v_k) = 8, \qquad \min_{130 \text{ cells}} \max_k d(v_k) = 3.$$

- In principle, we should consider the weighted graph with weights $w_{e_k} := \|v_i - v_j\|^{-1}$. But for simplicity, we only consider the unweighted graphs/trees here. (One could justify this by resampling the dendrite coordinates with a uniform sampling rate.)

Our Dataset $\Longrightarrow$ Trees . . .

- Let G be a graph (in fact a tree) representing an RGC.
- Let $V = V(G) = \{v_1, \ldots, v_n\}$ where $v_k \in \mathbb{R}^3$, be a set of vertices representing sample points along dendrite arbors. n ranges between 565 and 24474 depending on the RGCs.
- Let $E = E(G) = \{e_1, \ldots, e_m\}$ be a set of edges where $e_k = (v_i, v_j)$ represents an edge (or line segment) connecting between adjacent vertices v_i, v_j for some $1 \leq i, j \leq n$. Note that $|E(G)| = |V(G)| - 1$ since G is a tree.
- Let $d(v_k) = d_{v_k}$ be the degree of the vertex v_k. In our dataset,

$$\max_{130 \text{ cells}} \max_k d(v_k) = 8, \qquad \min_{130 \text{ cells}} \max_k d(v_k) = 3.$$

- In principle, we should consider the weighted graph with weights $w_{e_k} := \|v_i - v_j\|^{-1}$. But for simplicity, we only consider the unweighted graphs/trees here. (One could justify this by resampling the dendrite coordinates with a uniform sampling rate.)

Outline

Our Strategy

Step 1: Construct the Laplacian matrix (often called the combinatorial Laplacian matrix)

$$L(G) := D(G) - A(G)$$
$$D(G) := \mathrm{diag}(d_{v_1}, \ldots, d_{v_n}) \quad \text{the degree matrix}$$
$$A(G) = (a_{ij}) \quad \text{the adjacency matrix where}$$
$$a_{ij} := \begin{cases} 1 & \text{if } v_i \sim v_j; \\ 0 & \text{otherwise.} \end{cases}$$

Step 2: Compute the eigenvalues of $L(G)$;

Step 3: Construct features using these eigenvalues;

Step 4: Repeat the above steps for all the RGCs and feed these feature vectors to clustering algorithms.

Our Strategy

Step 1: Construct the Laplacian matrix (often called the combinatorial Laplacian matrix)

$$L(G) := D(G) - A(G)$$

$$D(G) := \mathrm{diag}(d_{v_1}, \ldots, d_{v_n}) \quad \text{the degree matrix}$$

$$A(G) = (a_{ij}) \quad \text{the adjacency matrix where}$$

$$a_{ij} := \begin{cases} 1 & \text{if } v_i \sim v_j; \\ 0 & \text{otherwise.} \end{cases}$$

Step 2: Compute the eigenvalues of $L(G)$;

Step 3: Construct features using these eigenvalues;

Step 4: Repeat the above steps for all the RGCs and feed these feature vectors to clustering algorithms.

Our Strategy

Step 1: Construct the Laplacian matrix (often called the combinatorial Laplacian matrix)

$$L(G) := D(G) - A(G)$$

$$D(G) := \operatorname{diag}(d_{v_1}, \ldots, d_{v_n}) \quad \text{the degree matrix}$$

$$A(G) = (a_{ij}) \quad \text{the adjacency matrix where}$$

$$a_{ij} := \begin{cases} 1 & \text{if } v_i \sim v_j; \\ 0 & \text{otherwise.} \end{cases}$$

Step 2: Compute the eigenvalues of $L(G)$;

Step 3: Construct features using these eigenvalues;

Step 4: Repeat the above steps for all the RGCs and feed these feature vectors to clustering algorithms.

Our Strategy

Step 1: Construct the Laplacian matrix (often called the combinatorial Laplacian matrix)

$$L(G) := D(G) - A(G)$$

$$D(G) := \mathrm{diag}(d_{v_1}, \ldots, d_{v_n}) \quad \text{the degree matrix}$$

$$A(G) = (a_{ij}) \quad \text{the adjacency matrix where}$$

$$a_{ij} := \begin{cases} 1 & \text{if } v_i \sim v_j; \\ 0 & \text{otherwise.} \end{cases}$$

Step 2: Compute the eigenvalues of $L(G)$;

Step 3: Construct features using these eigenvalues;

Step 4: Repeat the above steps for all the RGCs and feed these feature vectors to clustering algorithms.

Outline

Why Graph Laplacians?

- Eigenvalues of $L(G)$ reflect various intrinsic geometric information about the graph including
  - connectivity or the number of separated components
  - diameter (the maximum distance over all pairs of vertices)
  - mean distance, ...
  - Fan Chung: *Spectral Graph Theory*, AMS, 1997

    is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs.

- Eigenvectors of $L(G)$ also play a useful role to understand a graph (e.g., the discrete nodal domain theorem useful for grouping vertices; see Bıyıkoğlu, Leydold, & Stadler, LNM, Springer, 2007)

Why Graph Laplacians?

- Eigenvalues of $L(G)$ reflect various intrinsic geometric information about the graph including
  - connectivity or the number of separated components
  - diameter (the maximum distance over all pairs of vertices)
  - mean distance, ...
  - Fan Chung: *Spectral Graph Theory*, AMS, 1997

    is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs.

- Eigenvectors of $L(G)$ also play a useful role to understand a graph (e.g., the discrete nodal domain theorem useful for grouping vertices; see Bıyıkoğlu, Leydold, & Stadler, LNM, Springer, 2007)

Why Graph Laplacians?

- Eigenvalues of $L(G)$ reflect various intrinsic geometric information about the graph including
  - connectivity or the number of separated components
  - diameter (the maximum distance over all pairs of vertices)
  - mean distance, ...
  - Fan Chung: *Spectral Graph Theory*, AMS, 1997

    is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs.

- Eigenvectors of $L(G)$ also play a useful role to understand a graph (e.g., the discrete nodal domain theorem useful for grouping vertices; see Bıyıkoğlu, Leydold, & Stadler, LNM, Springer, 2007)

Why Graph Laplacians?

- Eigenvalues of $L(G)$ reflect various intrinsic geometric information about the graph including
  - connectivity or the number of separated components
  - diameter (the maximum distance over all pairs of vertices)
  - mean distance, ...
  - Fan Chung: *Spectral Graph Theory*, AMS, 1997

    is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs.
- Eigenvectors of $L(G)$ also play a useful role to understand a graph (e.g., the discrete nodal domain theorem useful for grouping vertices; see Bıyıkoğlu, Leydold, & Stadler, LNM, Springer, 2007)

Why Graph Laplacians?

- Eigenvalues of $L(G)$ reflect various intrinsic geometric information about the graph including
  - connectivity or the number of separated components
  - diameter (the maximum distance over all pairs of vertices)
  - mean distance, ...
  - Fan Chung: *Spectral Graph Theory*, AMS, 1997

    is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs.
- Eigenvectors of $L(G)$ also play a useful role to understand a graph (e.g., the discrete nodal domain theorem useful for grouping vertices; see Bıyıkoğlu, Leydold, & Stadler, LNM, Springer, 2007)

Why Graph Laplacians?

- Eigenvalues of $L(G)$ reflect various intrinsic geometric information about the graph including
  - connectivity or the number of separated components
  - diameter (the maximum distance over all pairs of vertices)
  - mean distance, ...
  - Fan Chung: *Spectral Graph Theory*, AMS, 1997

    is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs.

- Eigenvectors of $L(G)$ also play a useful role to understand a graph (e.g., the discrete nodal domain theorem useful for grouping vertices; see Bıyıkoğlu, Leydold, & Stadler, LNM, Springer, 2007)

Why Graph Laplacians?

- Eigenvalues of $L(G)$ reflect various intrinsic geometric information about the graph including
  - connectivity or the number of separated components
  - diameter (the maximum distance over all pairs of vertices)
  - mean distance, ...
  - Fan Chung: *Spectral Graph Theory*, AMS, 1997

    is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs.

- Eigenvectors of $L(G)$ also play a useful role to understand a graph (e.g., the discrete nodal domain theorem useful for grouping vertices; see Bıyıkoğlu, Leydold, & Stadler, LNM, Springer, 2007)

Aside: Graph Laplacian of a Line $\Longrightarrow$ DCT Type II Basis

$$L(G) = \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}$$

The eigenvectors of this matrix are exactly the DCT Type II basis vectors used for the JPEG image compression standard! (See e.g., Strang, SIAM Review, 1999).

Some Properties of Graph Laplacians

- Let $f \in L^2(V)$. Then
$$L(G)f(u) = d_u f(u) - \sum_{v \sim u} f(v),$$
i.e., this is a generalization of the finite difference approximation to the Laplace operator.
- Eigenvalues of $L(G)$ cannot uniquely determine the graph G. $\sim$ Kac (1966): "Can one hear the shape of a drum?" $\Longrightarrow$ Gordon, Webb, & Wolpert (1992): "One cannot hear the shape of a drum."
- An example of "isospectral" graphs (Tan, 1998; Fujii & Katsuda, 1999):

Some Properties of Graph Laplacians

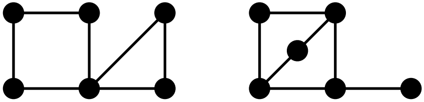
- Let $f \in L^2(V)$. Then
$$L(G)f(u) = d_u f(u) - \sum_{v \sim u} f(v),$$
i.e., this is a generalization of the finite difference approximation to the Laplace operator.
- Eigenvalues of $L(G)$ cannot uniquely determine the graph G. $\sim$ Kac (1966): "Can one hear the shape of a drum?" $\Longrightarrow$ Gordon, Webb, & Wolpert (1992): "One cannot hear the shape of a drum."
- An example of "isospectral" graphs (Tan, 1998; Fujii & Katsuda, 1999):

Some Properties of Graph Laplacians

- Let $f \in L^2(V)$. Then
$$L(G)f(u) = d_u f(u) - \sum_{v \sim u} f(v),$$
i.e., this is a generalization of the finite difference approximation to the Laplace operator.
- Eigenvalues of $L(G)$ cannot uniquely determine the graph G.
$\sim$ Kac (1966): "Can one hear the shape of a drum?" $\Longrightarrow$ Gordon, Webb, & Wolpert (1992): "One cannot hear the shape of a drum."
- An example of "isospectral" graphs (Tan, 1998; Fujii & Katsuda, 1999):

Some Properties of Graph Laplacians

- Let $f \in L^2(V)$. Then
$$L(G)f(u) = d_u f(u) - \sum_{v \sim u} f(v),$$
i.e., this is a generalization of the finite difference approximation to the Laplace operator.
- Eigenvalues of $L(G)$ cannot uniquely determine the graph G.
$\sim$ Kac (1966): "Can one hear the shape of a drum?" $\Longrightarrow$ Gordon, Webb, & Wolpert (1992): "One cannot hear the shape of a drum."
- An example of "isospectral" graphs (Tan, 1998; Fujii & Katsuda, 1999):

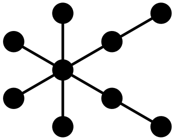


Some Properties of Graph Laplacians . . .

- However, certain classes of graphs can be completely determined by their Laplacian spectra: starlike trees (Omidi & Tajbakhsh, 2007), centipedes (Boulet, 2008), . . .
- ∃ some attempts to reconstruct graphs from their Laplacian spectra via combinatorial optimization (e.g., Comellas & Diaz-Lopez, 2008)
- Nothing prevents us from using the Laplacian spectra for characterizing dendrite patterns!

Some Properties of Graph Laplacians . . .

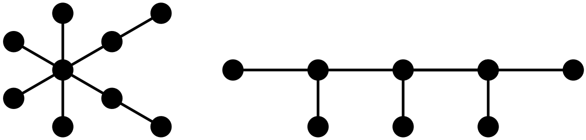
- However, certain classes of graphs can be completely determined by their Laplacian spectra: starlike trees (Omidi & Tajbakhsh, 2007), centipedes (Boulet, 2008), . . .



- ∃ some attempts to reconstruct graphs from their Laplacian spectra via combinatorial optimization (e.g., Comellas & Diaz-Lopez, 2008)
- Nothing prevents us from using the Laplacian spectra for characterizing dendrite patterns!

Some Properties of Graph Laplacians . . .

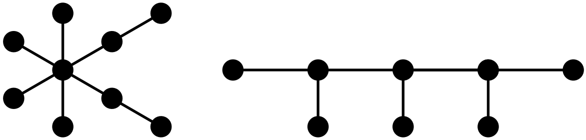
- However, certain classes of graphs can be completely determined by their Laplacian spectra: starlike trees (Omidi & Tajbakhsh, 2007), centipedes (Boulet, 2008), . . .



- ∃ some attempts to reconstruct graphs from their Laplacian spectra via combinatorial optimization (e.g., Comellas & Diaz-Lopez, 2008)
- Nothing prevents us from using the Laplacian spectra for characterizing dendrite patterns!

Some Properties of Graph Laplacians . . .

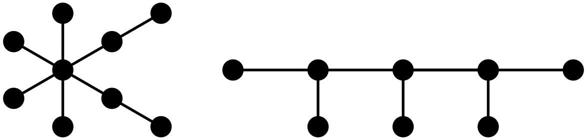
- However, certain classes of graphs can be completely determined by their Laplacian spectra: starlike trees (Omidi & Tajbakhsh, 2007), centipedes (Boulet, 2008), . . .



- ∃ some attempts to reconstruct graphs from their Laplacian spectra via combinatorial optimization (e.g., Comellas & Diaz-Lopez, 2008)
- Nothing prevents us from using the Laplacian spectra for characterizing dendrite patterns!

Some Properties of Graph Laplacians . . .

- However, certain classes of graphs can be completely determined by their Laplacian spectra: starlike trees (Omidi & Tajbakhsh, 2007), centipedes (Boulet, 2008), . . .



- ∃ some attempts to reconstruct graphs from their Laplacian spectra via combinatorial optimization (e.g., Comellas & Diaz-Lopez, 2008)
- Nothing prevents us from using the Laplacian spectra for characterizing dendrite patterns!

Some Notations and Definitions

- Let $|V(G)| = n$, and let $0 = \lambda_0(G) \leq \lambda_1(G) \leq \cdots \leq \lambda_{n-1}(G)$ be the sorted eigenvalues of $L(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.
- A vertex of degree 1 is called a pendant vertex; a vertex adjacent to a pendant vertex is called pendant neighbor.
- Let $p(G)$ and $q(G)$ be the number of pendant vertices and that of pendant neighbors, respectively.
- Let $S \subset V(G)$ be a nonempty subset of vertices of G.
- $\partial S := \{e = (u, v) \in E(G) \mid u \in S, v \notin S\}$, which is called the boundary of S.
- The distance matrix $\Delta(G)$ of G represents "distances" among the vertices, i.e., $(\Delta(G))_{i,j}$ is the number of edges in a shortest path from vertex v_i to vertex v_j.

Some Notations and Definitions

- Let $|V(G)| = n$, and let $0 = \lambda_0(G) \leq \lambda_1(G) \leq \cdots \leq \lambda_{n-1}(G)$ be the sorted eigenvalues of $L(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.
- A vertex of degree 1 is called a pendant vertex; a vertex adjacent to a pendant vertex is called pendant neighbor.
- Let $p(G)$ and $q(G)$ be the number of pendant vertices and that of pendant neighbors, respectively.
- Let $S \subset V(G)$ be a nonempty subset of vertices of G.
- $\partial S := \{e = (u, v) \in E(G) \,|\, u \in S, v \notin S\}$, which is called the boundary of S.
- The distance matrix $\Delta(G)$ of G represents "distances" among the vertices, i.e., $(\Delta(G))_{i,j}$ is the number of edges in a shortest path from vertex v_i to vertex v_j.

Some Notations and Definitions

- Let $|V(G)| = n$, and let $0 = \lambda_0(G) \le \lambda_1(G) \le \cdots \le \lambda_{n-1}(G)$ be the sorted eigenvalues of $L(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.
- A vertex of degree 1 is called a pendant vertex; a vertex adjacent to a pendant vertex is called pendant neighbor.
- Let $p(G)$ and $q(G)$ be the number of pendant vertices and that of pendant neighbors, respectively.
- Let $S \subset V(G)$ be a nonempty subset of vertices of G.
- $\partial S := \{e = (u, v) \in E(G) \mid u \in S, v \notin S\}$, which is called the boundary of S.
- The distance matrix $\Delta(G)$ of G represents "distances" among the vertices, i.e., $(\Delta(G))_{i,j}$ is the number of edges in a shortest path from vertex v_i to vertex v_j.

Some Notations and Definitions

- Let $|V(G)| = n$, and let $0 = \lambda_0(G) \leq \lambda_1(G) \leq \cdots \leq \lambda_{n-1}(G)$ be the sorted eigenvalues of $L(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.
- A vertex of degree 1 is called a pendant vertex; a vertex adjacent to a pendant vertex is called pendant neighbor.
- Let $p(G)$ and $q(G)$ be the number of pendant vertices and that of pendant neighbors, respectively.
- Let $S \subset V(G)$ be a nonempty subset of vertices of G.
- $\partial S := \{e = (u, v) \in E(G) \mid u \in S, v \notin S\}$, which is called the boundary of S.
- The distance matrix $\Delta(G)$ of G represents "distances" among the vertices, i.e., $(\Delta(G))_{i,j}$ is the number of edges in a shortest path from vertex v_i to vertex v_j.

Some Notations and Definitions

- Let $|V(G)| = n$, and let $0 = \lambda_0(G) \leq \lambda_1(G) \leq \cdots \leq \lambda_{n-1}(G)$ be the sorted eigenvalues of $L(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.
- A vertex of degree 1 is called a pendant vertex; a vertex adjacent to a pendant vertex is called pendant neighbor.
- Let $p(G)$ and $q(G)$ be the number of pendant vertices and that of pendant neighbors, respectively.
- Let $S \subset V(G)$ be a nonempty subset of vertices of G.
- $\partial S := \{e = (u, v) \in E(G) \mid u \in S, v \notin S\}$, which is called the boundary of S.
- The distance matrix $\Delta(G)$ of G represents "distances" among the vertices, i.e., $(\Delta(G))_{i,j}$ is the number of edges in a shortest path from vertex v_i to vertex v_j.

Some Notations and Definitions

- Let $|V(G)| = n$, and let $0 = \lambda_0(G) \leq \lambda_1(G) \leq \cdots \leq \lambda_{n-1}(G)$ be the sorted eigenvalues of $L(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.
- A vertex of degree 1 is called a pendant vertex; a vertex adjacent to a pendant vertex is called pendant neighbor.
- Let $p(G)$ and $q(G)$ be the number of pendant vertices and that of pendant neighbors, respectively.
- Let $S \subset V(G)$ be a nonempty subset of vertices of G.
- $\partial S := \{e = (u, v) \in E(G) \mid u \in S, v \notin S\}$, which is called the boundary of S.
- The distance matrix $\Delta(G)$ of G represents "distances" among the vertices, i.e., $(\Delta(G))_{i,j}$ is the number of edges in a shortest path from vertex v_i to vertex v_j.

Some Notations and Definitions

- Let $|V(G)| = n$, and let $0 = \lambda_0(G) \leq \lambda_1(G) \leq \cdots \leq \lambda_{n-1}(G)$ be the sorted eigenvalues of $L(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.
- A vertex of degree 1 is called a pendant vertex; a vertex adjacent to a pendant vertex is called pendant neighbor.
- Let $p(G)$ and $q(G)$ be the number of pendant vertices and that of pendant neighbors, respectively.
- Let $S \subset V(G)$ be a nonempty subset of vertices of G.
- $\partial S := \{e = (u, v) \in E(G) \mid u \in S, v \notin S\}$, which is called the boundary of S.
- The distance matrix $\Delta(G)$ of G represents "distances" among the vertices, i.e., $(\Delta(G))_{i,j}$ is the number of edges in a shortest path from vertex v_i to vertex v_j.

Some Notations and Definitions

- Let $|V(G)| = n$, and let $0 = \lambda_0(G) \leq \lambda_1(G) \leq \cdots \leq \lambda_{n-1}(G)$ be the sorted eigenvalues of $L(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.
- A vertex of degree 1 is called a pendant vertex; a vertex adjacent to a pendant vertex is called pendant neighbor.
- Let $p(G)$ and $q(G)$ be the number of pendant vertices and that of pendant neighbors, respectively.
- Let $S \subset V(G)$ be a nonempty subset of vertices of G.
- $\partial S := \{e = (u, v) \in E(G) \mid u \in S, v \notin S\}$, which is called the boundary of S.
- The distance matrix $\Delta(G)$ of G represents "distances" among the vertices, i.e., $(\Delta(G))_{i,j}$ is the number of edges in a shortest path from vertex v_i to vertex v_j.

Some Notations and Definitions

- The isoperimetric number of G is defined as

$$i(G) := \inf \left\{ \frac{|\partial S|}{|S|} \,\middle|\, \emptyset \neq S \subset V,\ |S| \leq \frac{n}{2} \right\},$$

which is closely related to the conductance of a graph, i.e., how fast a random walk on G converges to a stationary distribution.

- The Wiener index[2] $W(G)$ of a graph G is the sum of the entries in the upper triangular part of the distance matrix $\Delta(G)$.
- The Wiener index of a molecular graph has been used in chemical applications because it may exhibit a good correlation with physical and chemical properties (e.g., the boiling point, density, viscosity, surface tension, ...) of the corresponding molecule/material.

[2] proposed by Harry Wiener of Brooklyn College in 1947

Some Notations and Definitions

- The isoperimetric number of G is defined as

$$i(G) := \inf \left\{ \frac{|\partial S|}{|S|} \;\middle|\; \emptyset \neq S \subset V, \, |S| \leq \frac{n}{2} \right\},$$

  which is closely related to the conductance of a graph, i.e., how fast a random walk on G converges to a stationary distribution.
- The Wiener index[2] $W(G)$ of a graph G is the sum of the entries in the upper triangular part of the distance matrix $\Delta(G)$.
- The Wiener index of a molecular graph has been used in chemical applications because it may exhibit a good correlation with physical and chemical properties (e.g., the boiling point, density, viscosity, surface tension, ...) of the corresponding molecule/material.

[2] proposed by Harry Wiener of Brooklyn College in 1947

Some Notations and Definitions

- The isoperimetric number of G is defined as

$$i(G) := \inf \left\{ \frac{|\partial S|}{|S|} \,\middle|\, \emptyset \neq S \subset V,\, |S| \leq \frac{n}{2} \right\},$$

  which is closely related to the conductance of a graph, i.e., how fast a random walk on G converges to a stationary distribution.
- The Wiener index[2] $W(G)$ of a graph G is the sum of the entries in the upper triangular part of the distance matrix $\Delta(G)$.
- The Wiener index of a molecular graph has been used in chemical applications because it may exhibit a good correlation with physical and chemical properties (e.g., the boiling point, density, viscosity, surface tension, ...) of the corresponding molecule/material.

[2] proposed by Harry Wiener of Brooklyn College in 1947

Some Basic Theorems

See Chung (1997), Merris (1994), Mohar (1992), Urakawa (2002), . . .

- $m_G(0)$ is equal to the number of connected components of G.
- The number of pendant neighbors of G is bounded as:

$$p(G) - m_G(1) \leq q(G) \leq m_G(2, n],$$

  where the second inequality holds if G is connected and satisfies $2q(G) < n$.
- For $n \geq 4$, the isoperimetric number $i(G)$ satisfies

$$i(G) < \sqrt{\left(2 \max_{v \in V(G)} d_v - \lambda_1(G)\right) \lambda_1(G)}.$$

- Let G be a tree. Then

$$W(G) = \sum_{k=1}^{n-1} \frac{n}{\lambda_k}.$$

Some Basic Theorems

See Chung (1997), Merris (1994), Mohar (1992), Urakawa (2002), ...

- $m_G(0)$ is equal to the number of connected components of G.
- The number of pendant neighbors of G is bounded as:

$$p(G) - m_G(1) \leq q(G) \leq m_G(2, n],$$

  where the second inequality holds if G is connected and satisfies $2q(G) < n$.
- For $n \geq 4$, the isoperimetric number $i(G)$ satisfies

$$i(G) < \sqrt{\left(2 \max_{v \in V(G)} d_v - \lambda_1(G)\right) \lambda_1(G)}.$$

- Let G be a tree. Then

$$W(G) = \sum_{k=1}^{n-1} \frac{n}{\lambda_k}.$$

Some Basic Theorems

See Chung (1997), Merris (1994), Mohar (1992), Urakawa (2002), ...

- $m_G(0)$ is equal to the number of connected components of G.
- The number of pendant neighbors of G is bounded as:

$$p(G) - m_G(1) \leq q(G) \leq m_G(2, n],$$

  where the second inequality holds if G is connected and satisfies $2q(G) < n$.
- For $n \geq 4$, the isoperimetric number $i(G)$ satisfies

$$i(G) < \sqrt{\left(2 \max_{v \in V(G)} d_v - \lambda_1(G)\right) \lambda_1(G)}.$$

- Let G be a tree. Then

$$W(G) = \sum_{k=1}^{n-1} \frac{n}{\lambda_k}.$$

Some Basic Theorems

See Chung (1997), Merris (1994), Mohar (1992), Urakawa (2002), ...

- $m_G(0)$ is equal to the number of connected components of G.
- The number of pendant neighbors of G is bounded as:

$$p(G) - m_G(1) \leq q(G) \leq m_G(2, n],$$

  where the second inequality holds if G is connected and satisfies $2q(G) < n$.
- For $n \geq 4$, the isoperimetric number $i(G)$ satisfies

$$i(G) < \sqrt{\left(2 \max_{v \in V(G)} d_v - \lambda_1(G)\right) \lambda_1(G)}.$$

- Let G be a tree. Then

$$W(G) = \sum_{k=1}^{n-1} \frac{n}{\lambda_k}.$$

Outline

Features Used in Our Experiments

Feature 1: $(p(G) - m_G(1))/|V(G)|$ as a lower bound of the number of the pendant neighbors $q(G)$ with the normalization by $n = |V(G)|$;

Feature 2: The normalized Wiener index $W(G)/|V(G)|$;

Feature 3: $m_G(4,\infty)/|V(G)|$, i.e., the number of eigenvalues of $L(G)$ larger than 4 (normalized) ;

Feature 4: $\sqrt{\left(2 \max_{v \in V(G)} d_v - \lambda_1(G)\right) \lambda_1(G)}$, i.e., the upper bound of the isoperimetric number $i(G)$.

- We normalized Features 1, 2, 3, by $n = |V(G)|$ because we wanted to make features less dependent on the number of samples or how the dendrite arbors are sampled. Of course, the number of vertices itself could be a feature although it may not be a decisive one.
- Feature 4 was not explicitly normalized because the isoperimetric number $i(G)$ itself is a normalized quantity in terms of number of vertices.

Features Used in Our Experiments

Feature 1: $(p(G) - m_G(1))/|V(G)|$ as a lower bound of the number of the pendant neighbors $q(G)$ with the normalization by $n = |V(G)|$;

Feature 2: The normalized Wiener index $W(G)/|V(G)|$;

Feature 3: $m_G(4, \infty)/|V(G)|$, i.e., the number of eigenvalues of $L(G)$ larger than 4 (normalized) ;

Feature 4: $\sqrt{\left(2 \max_{v \in V(G)} d_v - \lambda_1(G)\right) \lambda_1(G)}$, i.e., the upper bound of the isoperimetric number $i(G)$.

- We normalized Features 1, 2, 3, by $n = |V(G)|$ because we wanted to make features less dependent on the number of samples or how the dendrite arbors are sampled. Of course, the number of vertices itself could be a feature although it may not be a decisive one.
- Feature 4 was not explicitly normalized because the isoperimetric number $i(G)$ itself is a normalized quantity in terms of number of vertices.

Features Used in Our Experiments

Feature 1: $(p(G) - m_G(1))/|V(G)|$ as a lower bound of the number of the pendant neighbors $q(G)$ with the normalization by $n = |V(G)|$;

Feature 2: The normalized Wiener index $W(G)/|V(G)|$;

Feature 3: $m_G(4, \infty)/|V(G)|$, i.e., the number of eigenvalues of $L(G)$ larger than 4 (normalized) ;

Feature 4: $\sqrt{(2 \max_{v \in V(G)} d_v - \lambda_1(G))\, \lambda_1(G)}$, i.e., the upper bound of the isoperimetric number $i(G)$.

- We normalized Features 1, 2, 3, by $n = |V(G)|$ because we wanted to make features less dependent on the number of samples or how the dendrite arbors are sampled. Of course, the number of vertices itself could be a feature although it may not be a decisive one.
- Feature 4 was not explicitly normalized because the isoperimetric number $i(G)$ itself is a normalized quantity in terms of number of vertices.

Features Used in Our Experiments

Feature 1: $(p(G) - m_G(1))/|V(G)|$ as a lower bound of the number of the pendant neighbors $q(G)$ with the normalization by $n = |V(G)|$;

Feature 2: The normalized Wiener index $W(G)/|V(G)|$;

Feature 3: $m_G(4, \infty)/|V(G)|$, i.e., the number of eigenvalues of $L(G)$ larger than 4 (normalized) ;

Feature 4: $\sqrt{\left(2 \max_{v \in V(G)} d_v - \lambda_1(G)\right) \lambda_1(G)}$, i.e., the upper bound of the isoperimetric number $i(G)$.

- We normalized Features 1, 2, 3, by $n = |V(G)|$ because we wanted to make features less dependent on the number of samples or how the dendrite arbors are sampled. Of course, the number of vertices itself could be a feature although it may not be a decisive one.
- Feature 4 was not explicitly normalized because the isoperimetric number $i(G)$ itself is a normalized quantity in terms of number of vertices.

Features Used in Our Experiments

Feature 1: $(p(G) - m_G(1))/|V(G)|$ as a lower bound of the number of the pendant neighbors $q(G)$ with the normalization by $n = |V(G)|$;

Feature 2: The normalized Wiener index $W(G)/|V(G)|$;

Feature 3: $m_G(4, \infty)/|V(G)|$, i.e., the number of eigenvalues of $L(G)$ larger than 4 (normalized) ;

Feature 4: $\sqrt{\left(2 \max_{v \in V(G)} d_v - \lambda_1(G)\right) \lambda_1(G)}$, i.e., the upper bound of the isoperimetric number $i(G)$.

- We normalized Features 1, 2, 3, by $n = |V(G)|$ because we wanted to make features less dependent on the number of samples or how the dendrite arbors are sampled. Of course, the number of vertices itself could be a feature although it may not be a decisive one.
- Feature 4 was not explicitly normalized because the isoperimetric number $i(G)$ itself is a normalized quantity in terms of number of vertices.

Features Used in Our Experiments

Feature 1: $(p(G) - m_G(1))/|V(G)|$ as a lower bound of the number of the pendant neighbors $q(G)$ with the normalization by $n = |V(G)|$;

Feature 2: The normalized Wiener index $W(G)/|V(G)|$;

Feature 3: $m_G(4, \infty)/|V(G)|$, i.e., the number of eigenvalues of $L(G)$ larger than 4 (normalized) ;

Feature 4: $\sqrt{\left(2 \max_{v \in V(G)} d_v - \lambda_1(G)\right) \lambda_1(G)}$, i.e., the upper bound of the isoperimetric number $i(G)$.

- We normalized Features 1, 2, 3, by $n = |V(G)|$ because we wanted to make features less dependent on the number of samples or how the dendrite arbors are sampled. Of course, the number of vertices itself could be a feature although it may not be a decisive one.
- Feature 4 was not explicitly normalized because the isoperimetric number $i(G)$ itself is a normalized quantity in terms of number of vertices.

Features Used in Our Experiments . . .

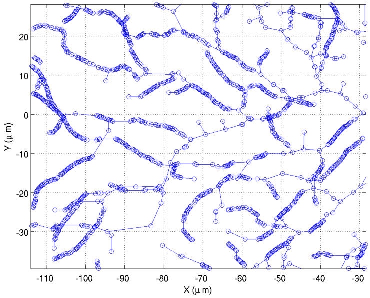
- Feature 1 was used because the number of pendant neighbors seems to be strongly related to the so-called spines, short protrusions from the dendrite arbors.
- Hence, we expect that the larger this lower bound $p(G) - m_G(1)$ is, the more likely for the RGC to have spines.

Features Used in Our Experiments . . .

- Feature 1 was used because the number of pendant neighbors seems to be strongly related to the so-called spines, short protrusions from the dendrite arbors.
- Hence, we expect that the larger this lower bound $p(G) - m_G(1)$ is, the more likely for the RGC to have spines.

Features Used in Our Experiments . . .

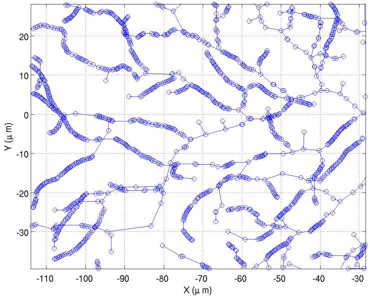
- Feature 1 was used because the number of pendant neighbors seems to be strongly related to the so-called spines, short protrusions from the dendrite arbors.
- Hence, we expect that the larger this lower bound $p(G) - m_G(1)$ is, the more likely for the RGC to have spines.



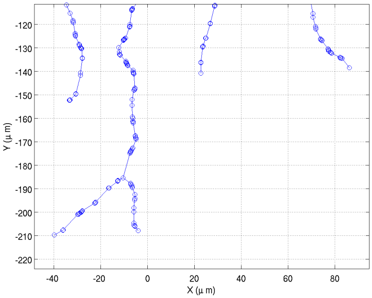
(a) RGC #60; F_1 large

Features Used in Our Experiments . . .

- Feature 1 was used because the number of pendant neighbors seems to be strongly related to the so-called spines, short protrusions from the dendrite arbors.
- Hence, we expect that the larger this lower bound $p(G) - m_G(1)$ is, the more likely for the RGC to have spines.



(a) RGC #60; F_1 large



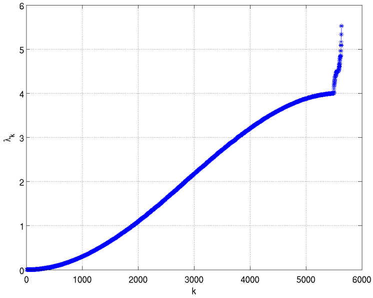
(b) RGC #100; F_1 small

Features Used in Our Experiments ...

- Feature 3, the normalized version of $m_G(4, \infty)$, was used because of the following observation:
- The eigenvalue distribution of each RGC consists of a smooth bell-shaped curve that ranges over $[0, 4]$ and the sudden burst above the value 4.

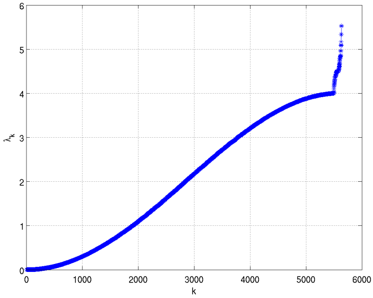
Features Used in Our Experiments . . .

- Feature 3, the normalized version of $m_G(4, \infty)$, was used because of the following observation:
- The eigenvalue distribution of each RGC consists of a smooth bell-shaped curve that ranges over $[0, 4]$ and the sudden burst above the value 4.

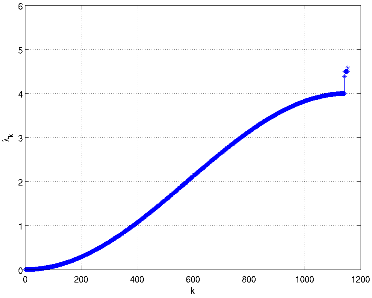


(a) RGC #60

- Feature 3, the normalized version of $m_G(4, \infty)$, was used because of the following observation:
- The eigenvalue distribution of each RGC consists of a smooth bell-shaped curve that ranges over $[0, 4]$ and the sudden burst above the value 4.



(a) RGC #60



(b) RGC #100

Features Used in Our Experiments . . .

We have observed that this value 4 is critical since:

- the eigenfunctions corresponding to the eigenvalues below 4 are semi-global oscillations (like Fourier cosines/sines) over the entire dendrites or one of the dendrite arbors;
- those corresponding to the eigenvalues above 4 are much more localized (like wavelets) around branches.

Features Used in Our Experiments . . .

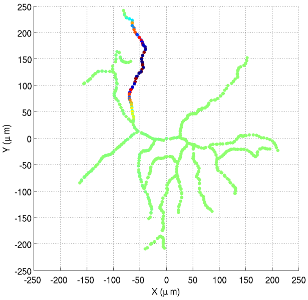
We have observed that this value 4 is critical since:

- the eigenfunctions corresponding to the eigenvalues below 4 are semi-global oscillations (like Fourier cosines/sines) over the entire dendrites or one of the dendrite arbors;
- those corresponding to the eigenvalues above 4 are much more localized (like wavelets) around branches.

Features Used in Our Experiments . . .

We have observed that this value 4 is critical since:

- the eigenfunctions corresponding to the eigenvalues below 4 are semi-global oscillations (like Fourier cosines/sines) over the entire dendrites or one of the dendrite arbors;
- those corresponding to the eigenvalues above 4 are much more localized (like wavelets) around branches.

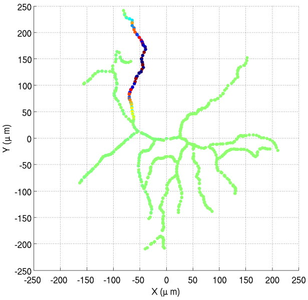


(a) RGC #100; $\lambda_{1141} = 3.9994$

Features Used in Our Experiments . . .

We have observed that this value 4 is critical since:

- the eigenfunctions corresponding to the eigenvalues below 4 are semi-global oscillations (like Fourier cosines/sines) over the entire dendrites or one of the dendrite arbors;
- those corresponding to the eigenvalues above 4 are much more localized (like wavelets) around branches.

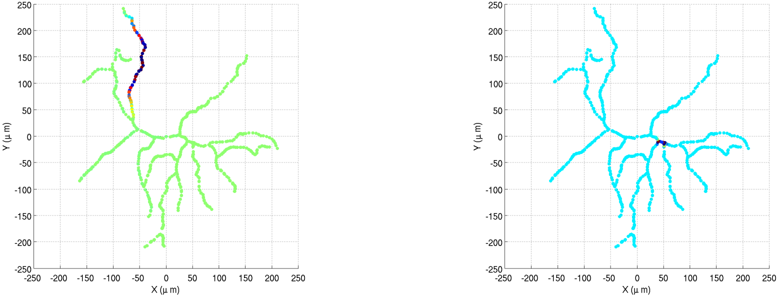


(a) RGC #100; $\lambda_{1141} = 3.9994$

Features Used in Our Experiments . . .

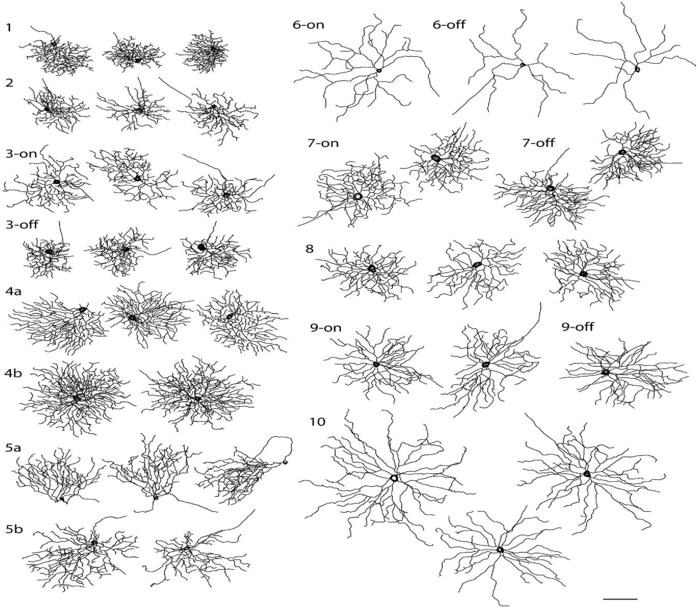
We have observed that this value 4 is critical since:

- the eigenfunctions corresponding to the eigenvalues below 4 are semi-global oscillations (like Fourier cosines/sines) over the entire dendrites or one of the dendrite arbors;
- those corresponding to the eigenvalues above 4 are much more localized (like wavelets) around branches.

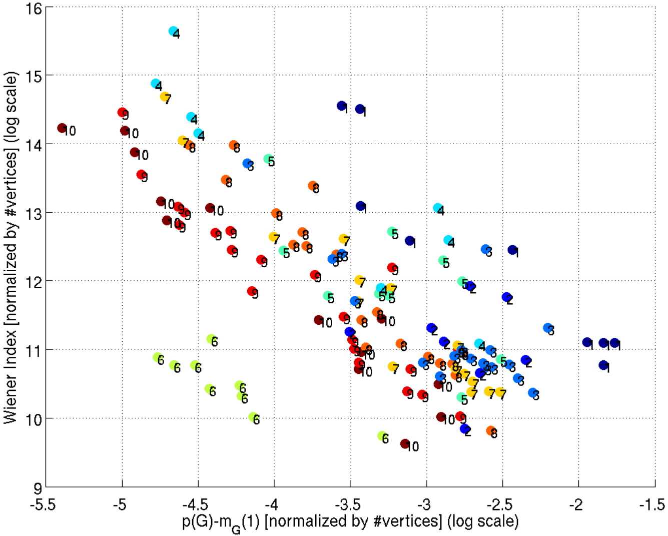


(a) RGC #100; $\lambda_{1141} = 3.9994$ (b) RGC #100; $\lambda_{1142} = 4.3829$

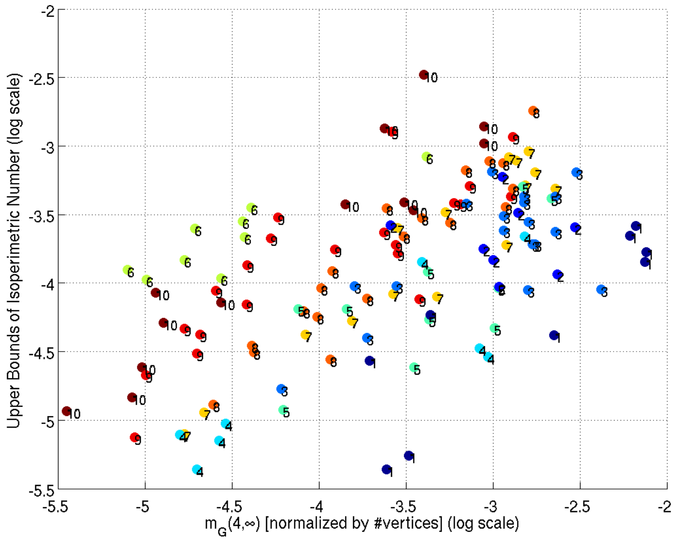
Recap: Clustering using Features Derived by Neurolucida®



Results: Scatter Plot; Feature 1 vs Feature 2



Results: Scatter Plot; Feature 3 vs Feature 4



Interpretation of the Results

- Cluster 6 RGCs separate themselves quite well from the other RGC clusters.
- In fact, the sparse and distributed dendrite patterns such as those in Clusters 6 and 10 are located below the major axis of the point clouds in the $F_1 - F_2$ scatter plot and above the major axis of the point clouds in the $F_3 - F_4$ scatter plot. $\Longrightarrow$ the dendrite patterns belonging to Cluster 6 and 10 have smaller number of spines and smaller Wiener indices compared to the other denser dendrite patterns such as Clusters 1 to 5.
- Considerable feature variability in Clusters 7 and 8.

Interpretation of the Results

- Cluster 6 RGCs separate themselves quite well from the other RGC clusters.
- In fact, the sparse and distributed dendrite patterns such as those in Clusters 6 and 10 are located below the major axis of the point clouds in the $F_1 - F_2$ scatter plot and above the major axis of the point clouds in the $F_3 - F_4$ scatter plot. $\Longrightarrow$ the dendrite patterns belonging to Cluster 6 and 10 have smaller number of spines and smaller Wiener indices compared to the other denser dendrite patterns such as Clusters 1 to 5.
- Considerable feature variability in Clusters 7 and 8.

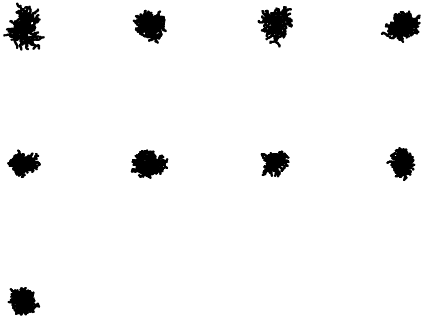
Interpretation of the Results

- Cluster 6 RGCs separate themselves quite well from the other RGC clusters.
- In fact, the sparse and distributed dendrite patterns such as those in Clusters 6 and 10 are located below the major axis of the point clouds in the $F_1 - F_2$ scatter plot and above the major axis of the point clouds in the $F_3 - F_4$ scatter plot. $\Longrightarrow$ the dendrite patterns belonging to Cluster 6 and 10 have smaller number of spines and smaller Wiener indices compared to the other denser dendrite patterns such as Clusters 1 to 5.
- Considerable feature variability in Clusters 7 and 8.

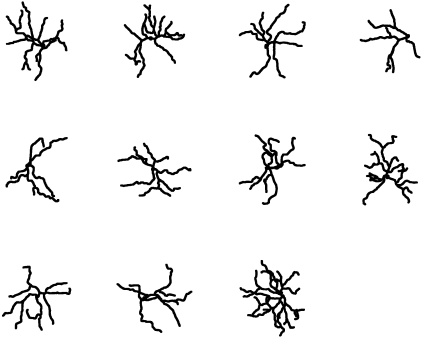
Interpretation of the Results

- Cluster 6 RGCs separate themselves quite well from the other RGC clusters.
- In fact, the sparse and distributed dendrite patterns such as those in Clusters 6 and 10 are located below the major axis of the point clouds in the $F_1 - F_2$ scatter plot and above the major axis of the point clouds in the $F_3 - F_4$ scatter plot. $\Longrightarrow$ the dendrite patterns belonging to Cluster 6 and 10 have smaller number of spines and smaller Wiener indices compared to the other denser dendrite patterns such as Clusters 1 to 5.
- Considerable feature variability in Clusters 7 and 8.

Cluster 1 vs Cluster 6 . . .



(a) Cluster 1



(b) Cluster 6

Outline

Conclusions & Future Plans

- Demonstrated the usefulness of the eigenvalues of graph Laplacians for dendrite pattern analysis although the results are still preliminary.
- Observed a global-to-local phase transition phenomenon of the eigenvalues and eigenfunctions of such dendrite patterns $\Longrightarrow$ leads to a theorem?
- Investigate the resampling of dendrite arbor samples.
- Analyze the features derived by Neurolucida®: are they derivable from the Laplacian eigenvalues?
- Compare the cost of features derivable by directly analyzing a graph with that by Laplacian eigenvalues (e.g., features related to $i(G)$).

Future Plans . . .

- Impose the Dirichlet boundary condition on the terminal nodes $\Longrightarrow$ eigenvalue problems of a graph with boundary; the discrete Dirichlet problem; the Faber-Krahn inequality, . . .
- Solve Poisson's equation with mixed boundary condition $\Longleftarrow$ the mean exit time $u(\mathbf{x})$ of particles released at a point $\mathbf{x}$ inside a bounded domain driven by Brownian motion is the solution of Poisson's equation $\Delta u = -1$ satisfying the zero Dirichlet boundary condition.
- Investigate metric (or quantum) graphs.
- Investigate how to model dendrite pattern generation and evolution, e.g., percolation on trees.

Outline

References

- Laplacian Eigenfunction Resource Page http://www.math.ucdavis.edu/~saito/lapeig/ contains
  - All the talk slides of the previous minisymposia "Laplacian Eigenfunctions and Their Applications, " which Mauro Maggioni, Xiaoming Huo, and I organized for ICIAM 2007 (Zürich) and SIAM Imaging Conference 2008 (San Diego); and
  - My Course Note (elementary) on "Laplacian Eigenfunctions: Theory, Applications, and Computations"
- The following articles are available at http://www.math.ucdavis.edu/~saito/publications/

  - N. Saito and E. Woei: "Analysis of neuronal dendrite patterns using eigenvalues of graph Laplacian," *Japan SIAM Letter*, vol. 1, pp. 13–16, 2009, Invited paper.
  - N. Saito: "Data analysis and representation using eigenfunctions of Laplacian on a general domain," *Applied & Computational Harmonic Analysis*, vol. 25, no. 1, pp. 68–97, 2008.

Acknowledgment

- Ernest Woei (UCD, Math)
- Linh Lieu (UCD, Math)
- Leo Chalupa (UCD, Neurobiology)
- Julie Coombs (UCD, Neurobiology)
- NSF
- ONR

Thank you very much for your attention!