# Comp790-166: Computational Biology

Lecture 11

February 18, 2022

# Good Morning Question

Last time we saw the learned about the data augmentation strategy for single-cell landscapes.

1. Summarize the steps to generate these augmented points, given the initial set of points, $\mathcal{X}$
2. Talk about a concern related to this augmentation approach that is related to changing the biological interpretation of the data.

# Today

- Finish GSP basics
- Filtering - specifically low-pass filtering
- Identifying condition-specific prototypical cells with MELD

# Intermission for Announcements

- Homework 1 is due by 11:59pm eastern time on February 23
- Project proposal template is now online `https://github.com/natalies-teaching/Comp790-166-CompBio-Spring2022/blob/main/Project_Proposal/Project_Proposal.pdf`
- Sign up a time to present your project on March 7 or March 9, `https://docs.google.com/spreadsheets/d/1fX52jKWDWbJO1iB6D7FHv1DNQ53LPs0S8yNCUnUbSwQ/edit?usp=sharing`

# Discussion about Project Proposals

- **Abstract:** Sell your idea in 3-5 sentences. This is really good practice for figuring out what the story is with a project. Convince us why we should care.
- **Formal Problem Statement:** This should be a 1 to 2 sentence summary of what your problem is. Easier said than done.....
- **Contributions:** Think about a list of contributions and then put this into formal writing.
- **Intended Experiments:** Realistically you can aim for 1 to 2 experiments (more if you want!)
- **Implementation:** What is the product that you will give to the scientific community? (e.g. well-documented open source software)

# Graph Signal Processing

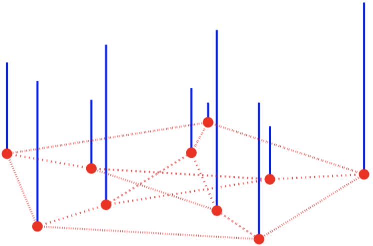


Figure: from Shuman *et al.* ArXiv. The purpose is to study the interplay between some signal and graph connectivity. Often we want to clean up or smooth out a particular signal, given the graph structure.

# How Localized is the Signal?

Remember, our friend Graph Laplacian ($\mathbf{L} = \mathbf{D} - \mathbf{A}$),

- Some very nice theory falls out about the eigenvalues of the Laplacian matrix in terms of how 'localized' a graph signal, **f**, is. For example **f** could be an expression of some protein.
  - First re-write **f** in terms of eigenvectors of the Laplacian
  - The eigenvectors corresponding to the first smallest eigenvalues of **L** are considered **low frequency**, and hence entries of the eigenvector entries corresponding to nodes that are connected should be similar
  - For higher **high frequencies** corresponding to 'later' eigenvalues, the values of the eigenvectors of adjacent nodes will be more different.

# Signal Specificity

Here we visualize eigenvector entries at nodes ($\mathbf{u}_0$,$\mathbf{u}_1$,$\mathbf{u}_{50}$)

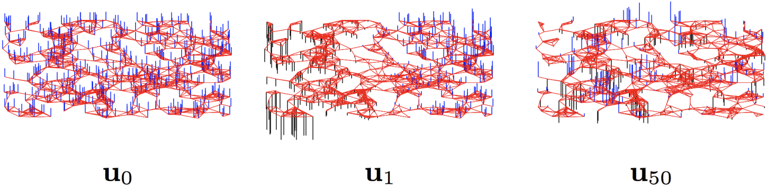


Figure: from GSP Review https://arxiv.org/abs/1211.0053

# Same Concept Visualized A Bit Differently

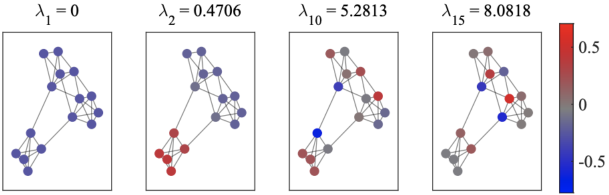



Figure: Notes are colored by their corresponding eigenvector component. From `https://arxiv.org/pdf/2008.01305.pdf`

# Similarly

Zero crossings mean that eigenvector entries are neighboring nodes will be different.

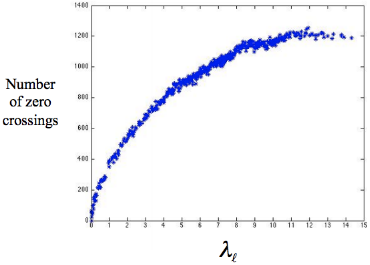



Figure: from GSP Review https://arxiv.org/abs/1211.0053

# What is Graph Fourier Transform (on a high level?)

- Explain frequency content of the graph signal (e.g. experimental measurements/labels/etc) as a weighted sum of the eigenvectors of the Graph Laplacian
- The eigenvectors of the Graph Laplacian comprise the **Graph Fourier Basis and can help to decouple high and low frequency signals**

# Local Variation of a Signal

The local variation of a signal or the sum of differences around a node can be written as,

$$
\begin{aligned}
(\mathcal{L}\mathbf{f})(i) &= ([\mathbf{D} - \mathbf{A}]\mathbf{f})(i) && (1)\\
&= d(i)\mathbf{f}(i) - \sum_j A_{ij}\mathbf{f}(j) && (2)\\
&= \sum_j A_{ij}(\mathbf{f}(i) - \mathbf{f}(j)) && (3)
\end{aligned}
$$

# Local Variation Leads to Total Variation

The total variation of a signal on a graph is defined as follows and is also known as the Laplacian Quadratic Form

$$TV(\mathbf{f}) = \sum_{i,j} A_{ij}(\mathbf{f}(i) - \mathbf{f}(j))^2 \quad (4)$$

$$= \mathbf{f}^T \mathcal{L} \mathbf{f} \quad (5)$$

- Note here I have been assuming that we have an unweighted graph, but you could certainly substitute $A_{ij}$ with a weighted version, $W_{ij}$

# Getting to Graph Fourier Basis

- Start with the eigendecomposition of of $\mathbf{L}$ as $\mathbf{L} = \mathbf{\Psi\Lambda\Psi}^T$
- We can look at eigenvectors, $\mathbf{\Psi} = [\psi_1, \psi_2, \dots, \psi_N]$ of $\mathcal{L}$
- and eigenvalues, $\mathbf{\Lambda} = [0 = \lambda_1 \leq \cdots \leq \lambda_N]$ of $\mathcal{L}$

# The Graph Fourier Transform of a Signal

The $i$th frequency component of a signal, $\mathbf{f}$ is the inner product between $\boldsymbol{\psi}_i$ and $\mathbf{f}$ and can be written as,

$$\hat{f}_i = \psi_i^T \mathbf{f} \tag{6}$$

The Graph Fourier Transform (GFT) is written as,

$$\hat{\mathbf{f}} = \boldsymbol{\Psi}^T \mathbf{f} \tag{7}$$

# GFT Will Be Used to Filter

- A filter on the graph will take in a signal and attenuate it according to a frequency response function.
- **Low-Pass Filter:** We filter or preserve only frequencies corresponding to eigenvalues below some threshold, $\lambda_k$. So, consider frequencies $\lambda_b$, with $\lambda_b < \lambda_k$
- **High-Pass Filters**: Preserve only frequencies corresponding to eigenvalues above some threshold, $\lambda_k$. So, consider frequencies $\lambda_b$, with $\lambda_b \geq \lambda_{k+1}$

# A Simple Low-Pass Filter

Define some filter $h$ as,

$$h : [0, \max(\boldsymbol{\Lambda})] \rightarrow [0, 1] \tag{8}$$

Assuming the cutoff is $\lambda_k$,
$h(x) > 0$, for $x < \lambda_k$ and $h(x) = 0$, otherwise

# Defining Notation and Applying Filter to GFT

Define $h(\mathbf{\Lambda})$ as a diagonal matrix of eigenvalues with the filter applied. Based on what we computed with GFT, the filtered signal, $\hat{\mathbf{f}_{filt}}$ can be computed as,

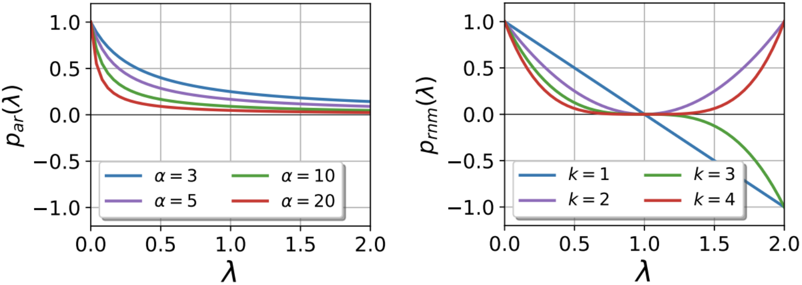
$$\hat{\mathbf{f}}_{\text{f}ilt} = h(\mathbf{\Lambda})\hat{\mathbf{f}} \tag{9}$$

# Applying a Filter in General

In general, you can filter an original signal, $\mathbf{f}$ in general as,

$$\underbrace{\mathbf{\Psi}(\mathbf{I} + \alpha\mathbf{\Lambda})^{-1}\mathbf{\Psi}^T}_{\text{Filtered Graph Laplacian}} \ \mathbf{f}. \tag{10}$$

# Example Filters




(a) $p_{\text{ar}}(\lambda) = (1 + \alpha\lambda)^{-1}$ (b) $p_{\text{rnm}}(\lambda) = (1 - \lambda)^{k}$

Figure: from `https://openaccess.thecvf.com/content_CVPR_2019/papers/Li_Label_Efficient_Semi-Supervised_Learning_via_Graph_Filtering_CVPR_2019_paper.pdf`