Last time: $S \in \mathbb{R}^{n \times n}$ $S = V \Lambda V^H$

signal $x \in \mathbb{R}^n$ $\longrightarrow$ $GFT(x) = \hat{x} = V^H x$

Properties $[\hat{x}]_i \neq 0$

Today: Graph convolutional filters
- locality
- shift invariance
- permutation invariance
- frequency domain rep.
- expressivity

Besides diffusing signals on graphs & analyzing their spectrum, what else can we do with graph signals?

- Process (filter) them into desired outputs
- Learn representations of graphs / graph signals

E.g.:

- Filter out all signal components with frequency above some threshold λ
- Fill in a signal with missing node entries
- Given a molecule graph, predict some molecular property (e.g., ability to act as a drug)

The basic building block to process signals is the graph filter:

(DEF) A (linear) graph filter is defined as an operator

$$H : \mathbb{R}^n \to \mathbb{R}^n \quad , \quad x \longmapsto y = H.S \quad , \quad H \in \mathbb{R}^{n \times n}$$

I.e., a graph filter is a linear map from graph signals to graph signals

▶ What are potential issues with such filters?

- They do not incorporate the graph sparsity pattern (i,j in $\neq$ connected components $\Rightarrow H_{ij} \neq 0$ doesn't make sense)

- $[Sx]_i = \sum_j S_{ij} x_j = \sum_{j \in N_i} S_{ij} x_j$ $\quad$ $[Hx]_i \neq \sum_{j \in N_i} h_j x_j$

  ↳ not suitable for dist. systems

- The number of params is n^2 → bad for large G

- Once we design H, it only works for graphs of size n

Solution: Linear shift-invariant / convolutional graph filters

(DEF) $\quad y = H(S)\,x = \sum_{k=0}^{K-1} h_k S^k x \quad (*)$, $\quad h_0, \dots, h_{K-1} \in \mathbb{R}$

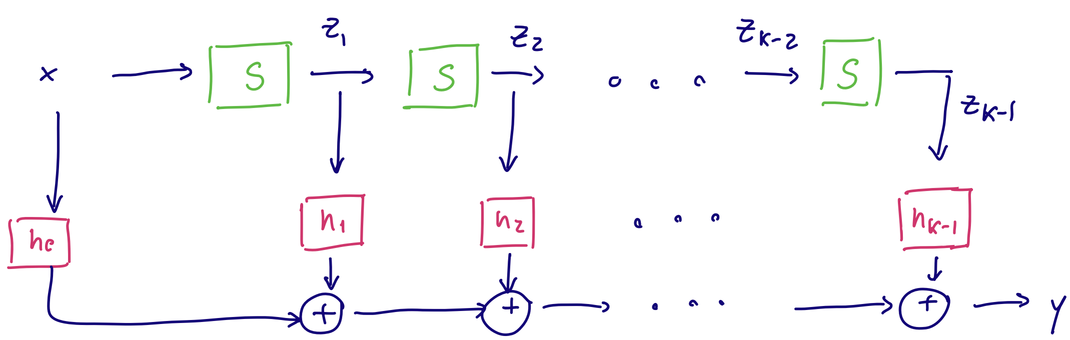
↳ filter coeffs. or taps

Properties:

1) Locality: filter $(*)$ is local

$$y = \sum_{k=0}^{K-1} h_k S^k x = \sum_{k=0}^{K-1} h_k S^{k-1} \underbrace{Sx}_{z_1} = \sum_{k=1}^{K-1} h_k S^{k-1} z_1 + h_0 x$$

$$= \sum_{k=1}^{K-1} h_k S^{k-2} \underbrace{S z_1}_{z_2} + h_0 x = \sum_{k=2}^{K-1} h_k S^{k-2} z_2 + h_1 z_1 + h_0 x = \dots$$



Recall $z_k = S z_{k-1}$ is a local operation

$\Rightarrow$

y is computed from $K-1$ successive local operations (aggregations) $\Rightarrow$ LOCAL

2) Shift equivariance

Let $y = H(S)x$. Assume $x' = Sx$. Is $y' = Sy$?

Pf.: $y' = H(S)x' = \sum_{k=0}^{K-1} h_k S^k . Sx = \sum_{k=0}^{K-1} h_k S^{k+1} x$

$= S \underbrace{\sum_{k=0}^{K-1} h_k S^k x}_{y} = Sy$

If x is shifted / diffused, y is shifted in the same way

3) Permutation equivariance

doubly stochastic matrix

Suppose we relabel nodes $\mathcal{V} = \{1, \dots, n\}$ using a permutation matrix $P \in \{0,1\}^{n \times n}$ $\boxed{P.\mathbb{1} = \mathbb{1} \qquad P^T \mathbb{1} = \mathbb{1}}$ $\to P^T P = I$

Given S and x, this corresponds to:

$S' = PSP^T \qquad x' = Px$

$\to$ What happens to $y = H(S)x$?

$y' = H(S')x' = \sum_{k=0}^{K-1} h_k (PSP^T)^k Px = \sum_{k=0}^{K-1} h_k PS^k \overbrace{P^T P}^{I} x$

$= P \sum_{k=0}^{K-1} h_k S^k x = Py$

I.e., if S & x are permuted/relabeled, y is permuted/relabeled in the same way

4) Spectral/frequency representation

inverse GFT

Given $S = V \Lambda V^H$, recall $\hat{x} = GFT\{x\} = V^H x$

→ what is $\hat{y}$?

$$\hat{y} = V^H y = V^H \sum_{k=0}^{K-1} h_k S^k x$$

$$= V^H \sum_{k=0}^{K-1} h_k S^k V \hat{x}$$

$$= \sum_{k=0}^{K-1} h_k V^H (V \Lambda V^H)^k V \hat{x} \qquad (V\Lambda V^H)^k \to V \Lambda^k V^H$$

$$= \sum_{k=0}^{K-1} h_k \cancel{V^H V} \Lambda^k \cancel{V^H V} \hat{x} = \sum_{k=0}^{K-1} h_k \Lambda^k \hat{x}$$

I.e., in the spectral/frequency domain, we have:

$$\boxed{\hat{y} = \sum_{k=0}^{K-1} h_k \Lambda^k \hat{x}}$$

so that the spectral representation/frequency response of $H(S)$ is:

$$\boxed{\widehat{H(S)} = \sum_{k=0}^{K-1} h_k \Lambda^k}$$

Note that:

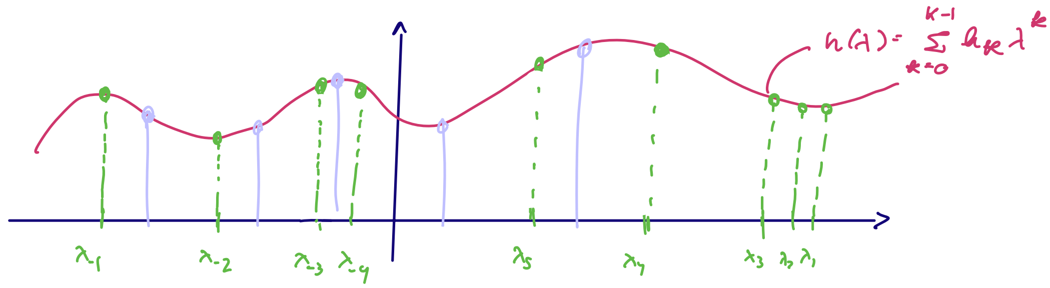
- The filter freq. response only depends on coeff. h_k & on the eigenvalues Λ
- $\tilde{G}$ w/ $\tilde{S}$ w/ $\tilde{\Lambda}$ defined by $h(\lambda) = \sum_{k=0}^{K-1} h_k \lambda^k$

$H(S) = h(S)$; $\widehat{H(S)} = h(\Lambda)$; $H(\tilde{S}) = h(\tilde{S})$ etc.

- $\widehat{H(S)}$ is pointwise in spectral domain

$$[\hat{y}]_i = \sum_{k=0}^{K-1} h_k \lambda_i^k [\hat{x}]_i$$

i.e., the ith spectral component of output y only depends on λ_i & $[\hat{x}]_i$



5) Expressivity: what functions can $H(S)$ represent?

Let's say we want to design a filter with spectral response $f: \mathbb{R} \to \mathbb{R}$, $\lambda \mapsto f(\lambda)$. Can this be implemented as a graph convolutional filter?

Yes, as long as f is analytic — i.e., a function w/ convergent Taylor series

E.g. $f(\lambda) = e^{-\frac{(\lambda-a)^2}{b}} \Rightarrow h(\lambda) = ?$

(Taylor series)

$$h(\lambda) = f(0) + \lambda \cdot f'(0) + \frac{\lambda^2 f''(0)}{2} + \frac{\lambda^3 f'''(0)}{6} + \dots$$

$h_0 = f(0) = e^{-a^2/b}$

$$f'(\lambda) = -\frac{2(\lambda-a)}{b} e^{-\frac{(\lambda-a)^2}{2}} \Rightarrow f'(0) = \frac{2a}{b} f(0) \Rightarrow h_1 = \frac{2a}{b} f(0)$$

$$f''(\lambda) = \left(-\frac{2}{b}(\lambda-a)\right)^2 e^{-\frac{(\lambda-a)^2}{b}} - \frac{2}{b} e^{-\frac{(\lambda-a)^2}{b}}$$

$$f''(0) = \left(\left(\frac{2a}{b}\right)^2 - \frac{2}{b}\right) f(0) \quad \text{etc...} \qquad h_2 = \frac{1}{2} f''(0)$$

$$H(S) = \sum_{k=0}^{K-1} h_k S^k$$

Obs.: We can, and in practice will, truncate K. This corresponds to the order K Taylor approximation of $f(\lambda)$.

What are the limitations of the above expressivity result?

- it is spectral; does not apply to graph domain
- it does not take into account $S(G)$ & x

In general, we want to answer the question:

Can we use $H(S)$ to represent or approximate any signal / representation y?

Let $S \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$. Find $h_0, \ldots, h_{K-1}$ s.t. $\tilde{y} = H(S)x = y$

Recall $\quad \hat{y} = h(\Lambda)\,\hat{x} \quad \Rightarrow \quad \hat{y} = \sum_{k=0}^{K-1} h_k \Lambda^k \hat{x}$

$$\Rightarrow (\hat{y})_i = \sum_{k=0}^{K-1} h_k \lambda_i^k [\hat{x}]_i$$

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \underbrace{\begin{bmatrix} \hat{x}_1 & \hat{x}_1\lambda_1 & \cdots & \hat{x}_1\lambda_1^{K-1} \\ \hat{x}_2 & \hat{x}_2\lambda_2 & \cdots & \hat{x}_2\lambda_2^{K-1} \\ \vdots & & & \\ \hat{x}_n & \hat{x}_n\lambda_n & \cdots & \hat{x}_n\lambda_n^{K-1} \end{bmatrix}} \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_K \end{pmatrix}$$

$$= \operatorname{diag}(\hat{x}) \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{K-1} \\ \vdots & & & & \\ 1 & \lambda_n & \lambda_n^2 & \cdots & \lambda_n^{K-1} \end{bmatrix} \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_{K-1} \end{pmatrix}$$

$$\operatorname{diag}(\hat{x})^{-1} \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{K-1} \\ \vdots & & & & \\ 1 & \lambda_n & \lambda_n^2 & \cdots & \lambda_n^{K-1} \end{bmatrix} \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_{K-1} \end{pmatrix}$$

1st assumption:

$\operatorname{diag}(\hat{x})$ is inv. $\Leftrightarrow$

$[\hat{x}]_i \neq 0 \quad \forall i$

x must span all of $\mathbb{R}^n$

Vandermonde matrix V

$$\det(V) = \prod_{1 \leq i \leq j \leq n} (\lambda_i - \lambda_j)$$

In the simplest case $K=n$, we only need the Vandermonde matrix to have an inverse. This happens when all λ_i are distinct, as $\det(V) = \prod_{1\leq i\leq j\leq n} (\lambda_i - \lambda_j)$

For arbitrary K, the Rouché-Capelli theorem states that a LS with n equations in K unknowns is consistent (i.e., has solution) iff the ranks of V and the augmented matrix are the same:

$$\left[\begin{array}{cccccc|c} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{K-1} & & \hat{y}_1/\hat{x}_1 \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{K-1} & & \hat{y}_2/\hat{x}_2 \\ \vdots & & & & \vdots & & \vdots \\ 1 & \lambda_n & \lambda_n^2 & \cdots & \lambda_n^{K-1} & & \hat{y}_n/\hat{x}_n \end{array}\right]$$

Hence: if $[\hat{x}]_i \neq 0 \ \forall i$, $\lambda_i \neq \lambda_j \ \forall i \neq j$, there always exist $K \leq n$ coeffs $h_0, \ldots h_{K-1}$ s.t. $y = H(S)x$