Lecture 19 - Introduction to Graph Signal Processing

ECE 5260 – ORIE 5735

Anna Scaglione

Cornell Tech

April 13, 2026

Content

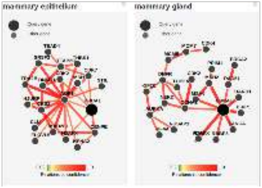
Cornell University

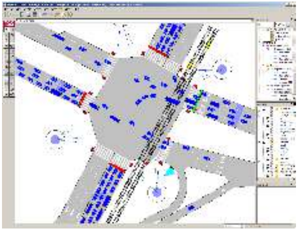
Outline

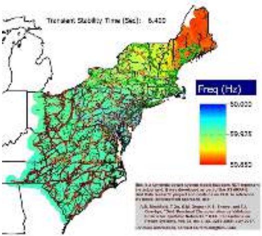
Cornell University

'Big Data" from network processes

Cornell University







Network dynamics produce data (Module 2)

- Processes affected by **relational** parameters: social, economic, biological.
- Processes due to **networked** infrastructures: transportation, gas, water, electric power

From shallow embeddings to GSP

Cornell University

- In discussing network embeddings and GRL last time, we represented each node by a low-dimensional vector derived from graph structure;
  - E.g. in spectral methods, those coordinates come from leading or trailing eigenvectors of a graph matrix such as the Laplacian.

  Essentially, we refined what we discussed in Module 1 as nodal characteristics (centrality etc.).
- We also mentioned that nodal labels can be used in (semi) supervised GRL. These would be **graph signals**.

Graph Signal Processing (GSP)

GSP is a generalization Digital Signal Processing (DSP) and Fourier analysis for such signals.

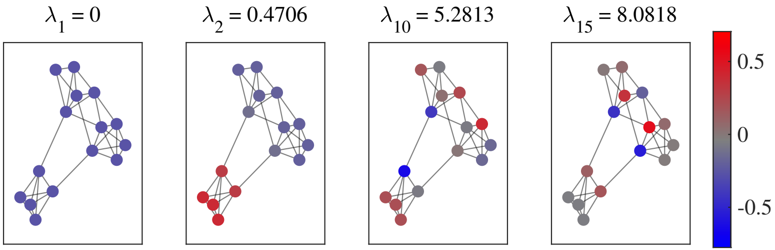
Two views of the same spectrum

Cornell University

Spoiler alert

The natural Fourier basis for GSP is given by the eigenvector of the Graph Laplacian.

- **Embedding view**: use a few (least significant) eigenvectors to assign coordinates to nodes.
- **GSP view**: use all eigenvectors as a basis to analyze signals on the graph. *Low eigenvalue* $\Leftrightarrow$ *low graph frequency.*

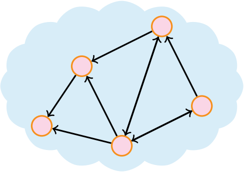


Outline

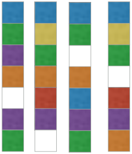
Cornell University

Graph Signal Processing Modeling

- Associate a *graph* $G = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$ to the coordinate system of the data set
- Graph signals are vectors defined on nodes of G, i.e., $\boldsymbol{s} \in \mathbb{R}^N$.
- Generalize DSP extending the notion of shift operator $z^{-1} \rightarrow \ \mathbf{S} = \mathbf{\Phi}(\mathbf{A})$



Filter & Generate $\longrightarrow$



GSP tools

Various DSP techniques have been extended to graphs — graph Fourier analysis, interpolation, sampling, multi-scale filtering,

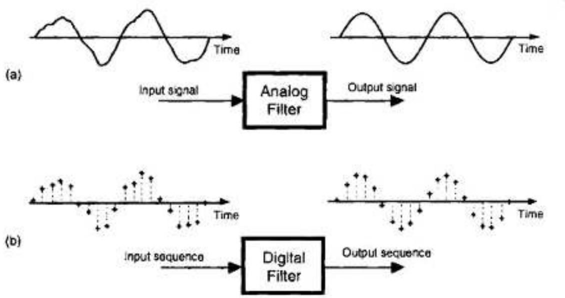
Outline

Cornell University

Outline

Cornell University

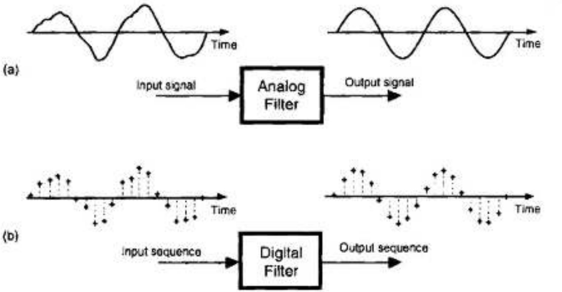
The notion of filter



- Digital Signal Processing (DSP) focuses on time series $x_t, \ t \in \mathbb{Z}$ and linear operators called filters
- The input of the filter is a time series $s_t \ t \in \mathbb{Z}$ and the output is a linear combination of the inputs

The notion of filter

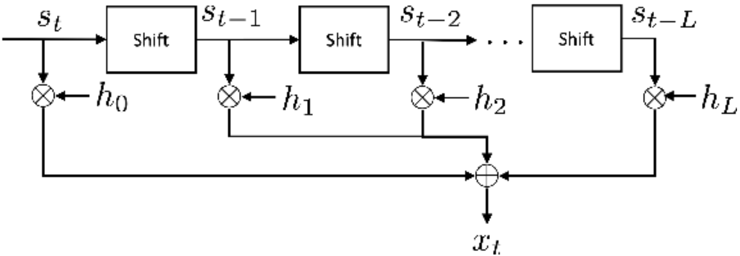
Cornell University



- Digital Signal Processing (DSP) focuses on time series $x_t,\ t \in \mathbb{Z}$ and linear operators called filters
- The input of the filter is a time series s_t $t \in \mathbb{Z}$ and the output is a linear combination of the inputs

The notion of filter

Cornell University



- For a Linear Time Invariant (LTI) the **input-output** relationship of a *finite memory* and *causal filter*

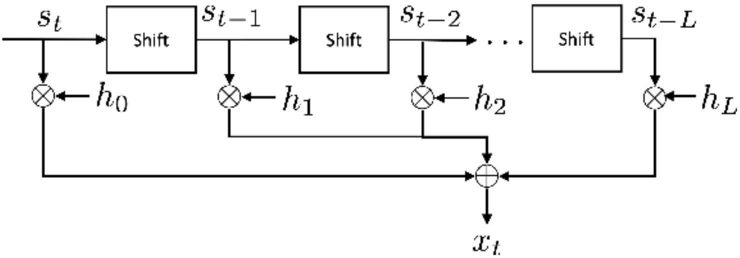
$$x_t = h_0 s_t + h_1 s_{t-1} + \ldots + h_L s_{t-L}$$

where L is the memory of the filter.

- The coefficients $h_0, \ldots, h_L$ are called, the impulse response. The remain the same irrespective of t (thus, LTI filter).

The notion of filter

Cornell University



- For a Linear Time Invariant (LTI) the **input-output** relationship of a *finite memory* and *causal filter*

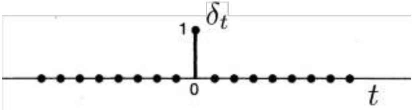
$$x_t = h_0 s_t + h_1 s_{t-1} + \ldots + h_L s_{t-L}$$

  where L is the memory of the filter.
- The coefficients $h_0, \ldots, h_L$ are called, the impulse response. The remain the same irrespective of t (thus, LTI filter).

Filtering in general

Cornell University

- In general a filter can combine an arbitrary number of samples in the sequence:
$$x_t = \sum_{\ell=-\infty}^{+\infty} h_\ell s_{t-\ell}$$
this operation is called *convolution* $x_t := h_t \star s_t$.
- Let us consider the impulse signal



- If the input is $s_t = \delta_t$, then the output is h_t (hence the name **impulse response** for h_t):

$$x_t = \sum_{\ell=-\infty}^{+\infty} h_k s_{t-\ell} = \sum_{\ell=-\infty}^{+\infty} h_k \delta_{t-\ell} \equiv h_t$$

Shift invariance

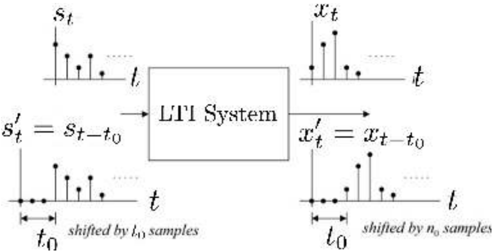
Cornell University

- Linear Time Invariant (LTI) refers to the fact thzat the **filter** is **a shift invariant operator**

Shift invariance

A linear system is a Linear Time Invariant (LTI) (or homogeneous) filter is shifting the input shifts the output by the same amount. Mathematically:

$$x_t = h_t \star s_t \quad \Leftrightarrow \quad x_{t-d} \equiv h_t \star s_{t-d}$$



Autoregressive and Moving average filters

Cornell University

- A finite memory filter has the feedforward structure we showed before, and is called **Moving Average (MA)**:

$$x_t = \sum_{\ell=0}^{r} b_\ell s_{t-\ell}$$

- A filter that generate an infinite impulse response is teh **Autoregressive (AR)** model, which is a *feedback filter*:

$$x_t = \sum_{\ell=0}^{q} a_\ell x_{t-\ell} + b_0 s_t$$

Autoregressive and Moving average filters

Cornell University

- A finite memory filter has the feedforward structure we showed before, and is called **Moving Average (MA)**:

$$x_t = \sum_{\ell=0}^{r} b_\ell s_{t-\ell}$$

- A filter that generate an infinite impulse response is teh **Autoregressive (AR)** model, which is a *feedback filter*:

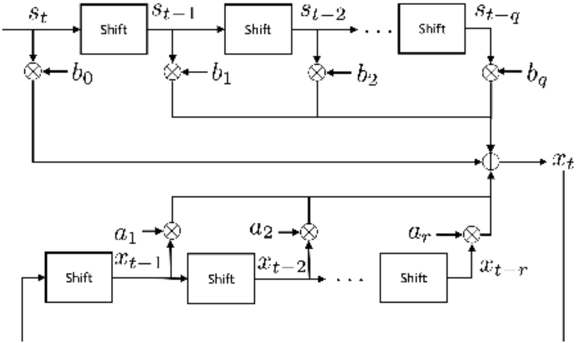
$$x_t = \sum_{\ell=0}^{q} a_\ell x_{t-\ell} + b_0 s_t$$

ARMA filter model

Cornell University

- The combination of the two is called **Auto-regressive Moving Average (ARMA)** filter:

$$x_t = \sum_{\ell=0}^{q} a_\ell x_{t-\ell} + \sum_{\ell=0}^{r} b_\ell s_{t-\ell}$$



ARMA models for random time series

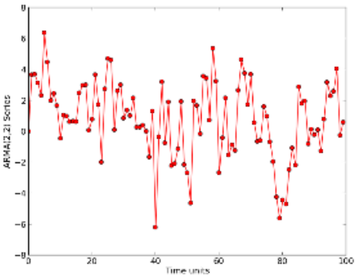
Cornell University

- What these filters abstractions are used for is primarily for modeling a time series

$$x_t = \sum_{\ell=0}^{q} a_\ell x_{t-\ell} + \sum_{\ell=0}^{r} b_\ell s_{t-\ell}$$

where $s_t \sim p(s_t)$ is assumed to be random i.i.d. in ARMA random processes.

- The idea is to say that there is an **excitation signal** s_t, that has independent samples, and that the observed signal x_t



Inference for ARMA

Cornell University

- Note that ARMA series are Markov processes - x_t depends on the following past history $\boldsymbol{p}_t = [x_{t-1}, \ldots, x_{t-q}, s_{t-1}, \ldots, s_{t-r}]$

$$p(x_t|\text{all history prior to time } t) = p(x_t|\boldsymbol{p}_t)$$

- If the filter parameters $[a_1, \ldots, a_q[, [b_0, \ldots, b_r]$ are known, we can forecast the future samples:

$$\hat{x}_t = \overbrace{\sum_{\ell=0}^{q} a_\ell x_{t-\ell} + \sum_{\ell=1}^{r} b_\ell s_{t-\ell}}^{[a_1,\ldots,a_q,b_1,\ldots,b_r]\boldsymbol{p}_t^\top} + b_0 \mathbb{E}[s_t]$$

- With GSP filter models we want to perform similar tasks but across different nodes in the graph

Inference for ARMA

Cornell University

- Note that ARMA series are Markov processes - x_t depends on the following past history $\boldsymbol{p}_t = [x_{t-1}, \ldots, x_{t-q}, s_{t-1}, \ldots, s_{t-r}]$

$$p(x_t|\text{all history prior to time } t) = p(x_t|\boldsymbol{p}_t)$$

- If the filter parameters $[a_1, \ldots, a_q[, [b_0, \ldots, b_r]$ are known, we can forecast the future samples:

$$\hat{x}_t = \overbrace{\sum_{\ell=0}^{q} a_\ell x_{t-\ell} + \sum_{\ell=1}^{r} b_\ell s_{t-\ell}}^{[a_1,\ldots,a_q,b_1,\ldots,b_r]\boldsymbol{p}_t^\top} + b_0 \mathbb{E}[s_t]$$

- With GSP filter models we want to perform similar tasks but across different nodes in the graph

Time graph

Cornell University

- The time index can be seen as a directed graph and the notion of neighborhood in time series is proximity in time ($v = t$, $\mathcal{N}_t = \{t-1, t+1\}$)

- Any *neighbohood* operation can involve only samples in $\mathcal{N}_t = \{t-1, t+1\}$

Time graph: linear neighborhood operations

- In general linear neighborhood operations have the form:

$$x_t = w_0 s_t + w_{-1} s_{t-1} + w_1 s_{t+1}$$

- A *time-shift* operation moves a signal forward in time (what was at time 0 is now at time 1 etc.) is a linear neighborhood operation

$$s_t \to x_t = s_{t-1}$$

  and it is an LTI filter with IR $h_t = \delta_{t-1}$.

- Another example is a local derivative:

$$\text{causal } x_t = s_t - s_{t-1}, \quad \text{non-causal } x_t = s_t - \frac{1}{2} s_{t-1} - \frac{1}{2} s_{t+1}$$

- Any filtering operation $h_t \star s_t$ can be interpreted as the composition of these elementary neighborhood operations

Outline

Cornell University

3 Review of Digital Signal Processing
- The notion of filter
- Fourier Transform

Complex exponential inputs

Cornell University

- When the input is an exponential $s_t \propto z^t$ (where $z \in \mathbb{C}$) the filters' shift invariant property is such that the output $x_t \propto z^t$
- Note that an exponential z^t is such that, if shifted, $z^{t-d} = z^t z^{-d}$. Let $s_t = z^t$ and calculate $x_t = h_t \star s_t = h_t \star z^t$

$$x_t = \sum_\ell h_\ell s_{t-\ell} = \sum_\ell h_\ell z^{t-\ell} = \sum_\ell h_\ell z^t z^{-\ell} = \overbrace{\left(\sum_\ell h_\ell z^{-\ell}\right)}^{\tilde{h}(z)} \overbrace{z^t}^{s_t}$$

- This proves when the input of an LTI filter is an exponential, the output is also an exponential proportional to the input

$$x_t = h_t \star z^t = \tilde{h}(z) \overbrace{z^t}^{s_t}$$

Complex exponential inputs

Cornell University

- When the input is an exponential $s_t \propto z^t$ (where $z \in \mathbb{C}$) the filters' shift invariant property is such that the output $x_t \propto z^t$
- Note that an exponential z^t is such that, if shifted, $z^{t-d} = z^t z^{-d}$. Let $s_t = z^t$ and calculate $x_t = h_t \star s_t = h_t \star z^t$

$$x_t = \sum_\ell h_\ell s_{t-\ell} = \sum_\ell h_\ell z^{t-\ell} = \sum_\ell h_\ell z^t z^{-\ell} = \overbrace{\left(\sum_\ell h_\ell z^{-\ell}\right)}^{\tilde{h}(z)} \overbrace{z^t}^{s_t}$$

- This proves when the input of an LTI filter is an exponential, the output is also an exponential proportional to the input

$$x_t = h_t \star z^t = \tilde{h}(z) \overbrace{z^t}^{s_t}$$

Complex exponential inputs

- When the input is an exponential $s_t \propto z^t$ (where $z \in \mathbb{C}$) the filters' shift invariant property is such that the output $x_t \propto z^t$
- Note that an exponential z^t is such that, if shifted, $z^{t-d} = z^t z^{-d}$. Let $s_t = z^t$ and calculate $x_t = h_t \star s_t = h_t \star z^t$

$$x_t = \sum_\ell h_\ell s_{t-\ell} = \sum_\ell h_\ell z^{t-\ell} = \sum_\ell h_\ell z^t z^{-\ell} = \overbrace{\left(\sum_\ell h_\ell z^{-\ell}\right)}^{\tilde{h}(z)} \overbrace{z^t}^{s_t}$$

- This proves when the input of an LTI filter is an exponential, the output is also an exponential proportional to the input

$$x_t = h_t \star z^t = \tilde{h}(z) \overbrace{z^t}^{s_t}$$

Eigenvectors of LTI systems

Cornell University

- Since, for an eigenvector input and output of a linear system are proportional another way of seeing this fact is to say that the signals $u_t = z^t$ are eigenvectors for LTI filters (a specific form of linear operators) because,

$$h_t \star z^t = \tilde{h}(z) z^t \quad \Leftrightarrow \quad h_t \star u_t \propto u_t$$

- The eigenvalues corresponding to $u_t = z^t$ are

$$\tilde{h}(z) = \sum_{\ell} h_\ell z^{-\ell}$$

which is also called the z-transform of h_t, or *transfer function* for the filter

Eigenvectors of LTI systems

Cornell University

- Since, for an eigenvector input and output of a linear system are proportional another way of seeing this fact is to say that the signals $u_t = z^t$ are eigenvectors for LTI filters (a specific form of linear operators) because,

$$h_t \star z^t = \tilde{h}(z) z^t \quad \Leftrightarrow \quad h_t \star u_t \propto u_t$$

- The eigenvalues corresponding to $u_t = z^t$ are

$$\tilde{h}(z) = \sum_{\ell} h_\ell z^{-\ell}$$

which is also called the z-transform of h_t, or *transfer function* for the filter

z-Transform and Fourier Analysis

Cornell University

- The Discrete Time Fourier Transform (DTFT) is the z-transform evaluated at $z = e^{-j\omega}$ (i.e. the unit circle)

$$\tilde{h}(e^{j\omega}) = \sum_{\ell} h_\ell e^{-j\omega\ell}$$

- The z and Fourier transform are defined obviously not only for the filters but for the signals themselves:

$$\tilde{x}(z) = \sum_t x_t z^{-t}, \quad \tilde{s}(z) = \sum_t s_t z^{-t}$$

z-Transform and Fourier Analysis

Cornell University

- The Discrete Time Fourier Transform (DTFT) is the z-transform evaluated at $z = e^{-j\omega}$ (i.e. the unit circle)

$$\tilde{h}(e^{j\omega}) = \sum_{\ell} h_\ell e^{-j\omega\ell}$$

- The z and Fourier transform are defined obviously not only for the filters but for the signals themselves:

$$\tilde{x}(z) = \sum_t x_t z^{-t}, \quad \tilde{s}(z) = \sum_t s_t z^{-t}$$

Multiplication property

Cornell University

- The other important relationship is as follows

$$x_t = h_t \star s_t \quad \Leftrightarrow \quad \tilde{x}(z) = \tilde{h}(z)\tilde{s}(z)$$

Proof:

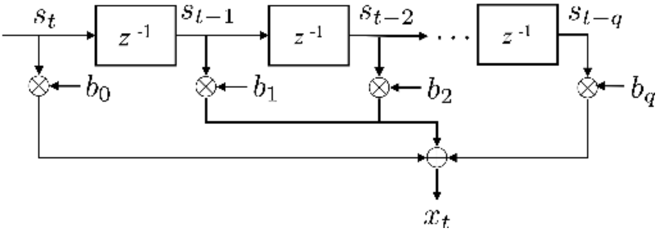
$$\tilde{x}(z) = \sum_t x_t z^{-t} = \sum_t \Big(\sum_\ell h_\ell s_{t-\ell}\Big) z^{-t} = \sum_t \sum_\ell h_\ell s_{t-\ell} z^{-(t-\ell+\ell)}$$

$$= \sum_t \overbrace{\Big(\sum_\ell h_\ell z^{-\ell}\Big)}^{\tilde{h}(z)} s_{t-\ell} z^{-(t-\ell)}$$

$$= \tilde{h}(z) \sum_t s_{t-\ell} z^{-(t-\ell)} = \tilde{h}(z) \overbrace{\sum_{t'} s_{t'} z^{-t'}}^{\tilde{s}(z)}$$

Filters representation with the shift operator

Cornell University

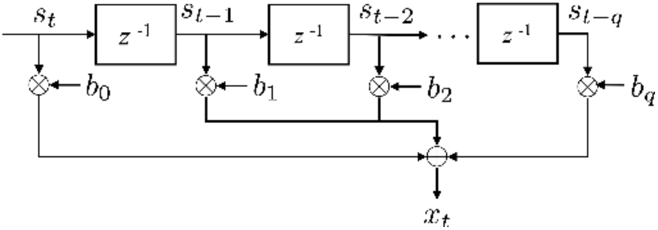


- Conventionally, in the filter block diagrams, we use z^{-1} to represent the time shift
- Shift invariance is obvious, since

$$\tilde{x}(z) = \tilde{h}(z)\tilde{x}(z) \quad \Leftrightarrow \quad \tilde{x}(z)z^{-d} = \tilde{h}(z)\tilde{s}(z)z^{-d}$$

Filters representation with the shift operator

Cornell University



- Conventionally, in the filter block diagrams, we use z^{-1} to represent the time shift
- Shift invariance is obvious, since

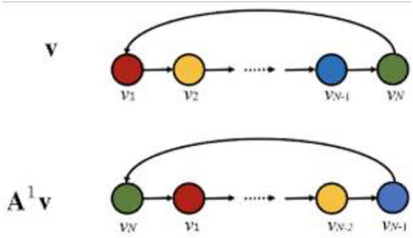
$$\tilde{x}(z) = \tilde{h}(z)\tilde{x}(z) \quad \Leftrightarrow \quad \tilde{x}(z)z^{-d} = \tilde{h}(z)\tilde{s}(z)z^{-d}$$

Finite sequences

Cornell University

- Time series abstractions last forever $-\infty \leq t \leq +\infty$
- In reality we only have finite sequences for DSP. The same is true for graphs: they may be large but not infinite!
- If we have N values, we can assume that the rest of the samples are zero, i.e. $x_t = 0$ for $t < 0$ and $t > N-1$
- In DPS is algorithmically convenient to manipulate finite signals considering their *periodic extension* x_t^{PE} defined as follows

$$x_t^{\text{PE}} = \sum_{p=-\infty}^{+\infty} x_{t-pN}$$

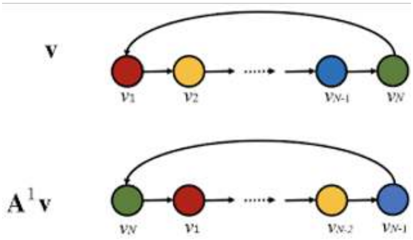


Finite sequences

Cornell University

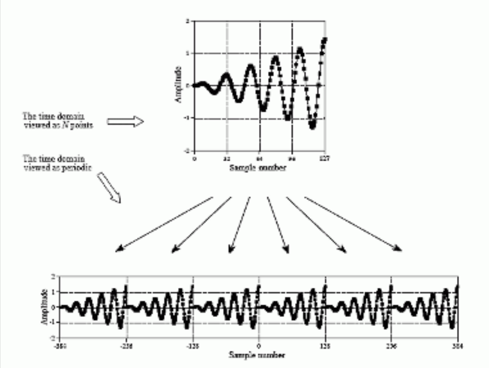
- Time series abstractions last forever $-\infty \leq t \leq +\infty$
- In reality we only have finite sequences for DSP. The same is true for graphs: they may be large but not infinite!
- If we have N values, we can assume that the rest of the samples are zero, i.e. $x_t = 0$ for $t < 0$ and $t > N-1$
- In DPS is algorithmically convenient to manipulate finite signals considering their *periodic extension* x_t^{PE} defined as follows

$$x_t^{\mathsf{PE}} = \sum_{p=-\infty}^{+\infty} x_{t-pN}$$



Periodic Extension

Cornell University



- The reason is that the Fourier Analysis of the signal can be has associated the discrete frequency components $u_t^{(n)} = e^{j2\pi \frac{nt}{N}}$

$$\tilde{x}_n = \frac{1}{N}\sum_{t=0}^{N-1} x_t e^{-j2\pi \frac{nt}{N}} \quad \Rightarrow \quad x_t^{\text{PE}} = \sum_{n=0}^{N-1} \tilde{x}_n e^{j2\pi \frac{nt}{N}}$$

Periodic Extension

Cornell University

- The following operation is called the Discrete Time Fourier Transform (DFT) and it is equal to the DTFT evaluated at $\omega = 2\pi\frac{nt}{N}$, i.e.

$$\tilde{x}_n = \frac{1}{N}\sum_{t=0}^{N-1} x_t e^{-j2\pi\frac{nt}{N}}, \quad \Rightarrow \quad \tilde{x}(e^{j\omega t})\Big|_{\omega=2\pi\frac{nt}{N}} \equiv \tilde{x}_n$$

- The Periodic Extension is the Inverse Discrete Fourier Transform (IDFT):

$$x_t^{\mathsf{PE}} = \sum_{n=0}^{N-1} \tilde{x}_n e^{j2\pi\frac{nt}{N}} \qquad \Rightarrow \quad x_t^{\mathsf{PE}} \equiv x_t, \quad t = 0, \ldots, N-1$$

Periodic Extension

Cornell University

- The following operation is called the Discrete Time Fourier Transform (DFT) and it is equal to the DTFT evaluated at $\omega = 2\pi\frac{nt}{N}$, i.e.

$$\tilde{x}_n = \frac{1}{N}\sum_{t=0}^{N-1} x_t e^{-j2\pi\frac{nt}{N}}, \quad \Rightarrow \quad \tilde{x}(e^{j\omega t})\Big|_{\omega=2\pi\frac{nt}{N}} \equiv \tilde{x}_n$$

- The Periodic Extension is the Inverse Discrete Fourier Transform (IDFT):

$$x_t^{\mathsf{PE}} = \sum_{n=0}^{N-1} \tilde{x}_n e^{j2\pi\frac{nt}{N}} \qquad \Rightarrow \quad x_t^{\mathsf{PE}} \equiv x_t, \quad t = 0, \ldots, N-1$$

Circular convolution

Cornell University

- When the input of an LTI filter is periodic the output is also periodic. In fact, it can be shown that, if s_t is periodic

$$x_t = h_t \star s_t = \sum_{n=0}^{N-1} \overbrace{(\tilde{h}_n \tilde{s}_n)}^{\tilde{x}_n} e^{j2\pi \frac{nt}{N}}$$

$$\tilde{h}_n = \frac{1}{N} \sum_{t=0}^{N-1} h_t e^{-j2\pi \frac{nt}{N}}, \quad \tilde{x}_n = \frac{1}{N} \sum_{t=0}^{N-1} s_t e^{-j2\pi \frac{nt}{N}}$$

- However, if $x_t = h_t \star s_t$ has a general finite duration input s_t, if we convolve with h_t the periodic extension s_t^{PE} of s_t with h_t we do not get the periodic extension of the output x_t

$$h_t \star s_t^{\mathsf{PE}} \neq x_t^{\mathsf{PE}}$$

what we have is a circular convolution

Circular convolution

Cornell University

- When the input of an LTI filter is periodic the output is also periodic. In fact, it can be shown that, if s_t is periodic

$$x_t = h_t \star s_t = \sum_{n=0}^{N-1} \overbrace{(\tilde{h}_n \tilde{s}_n)}^{\tilde{x}_n} e^{j2\pi \frac{nt}{N}}$$

$$\tilde{h}_n = \frac{1}{N} \sum_{t=0}^{N-1} h_t e^{-j2\pi \frac{nt}{N}}, \quad \tilde{x}_n = \frac{1}{N} \sum_{t=0}^{N-1} s_t e^{-j2\pi \frac{nt}{N}}$$

- However, if $x_t = h_t \star s_t$ has a general finite duration input s_t, if we convolve with h_t the periodic extension s_t^{PE} of s_t with h_t we do not get the periodic extension of the output x_t

$$h_t \star s_t^{\mathsf{PE}} \neq x_t^{\mathsf{PE}}$$

what we have is a circular convolution

Outline

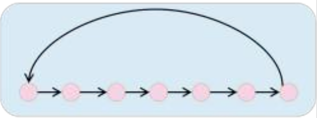
Cornell University

The Graph Shift Operator (GSO)

Cornell University

Point of view: algebraic SP, introduced in [1]

- For a (periodic/time-limited) signal the canonical coordinate vectors $\boldsymbol{x}$ form a cyclic graph G



$$\rightarrow \quad \boldsymbol{A} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \end{pmatrix}$$

- The graph shift operator (GSO) $\mathbf{S}$ can only mix elements adjacent in G;
- A filter $\mathcal{H}(\mathbf{S})$ is a linear operator invariant relative to the GSO $\mathbf{S}$

$$\boldsymbol{x} = \mathcal{H}(\mathbf{S})\mathbf{s} \quad \rightarrow \quad \mathbf{S}\mathbf{x} = \mathbf{S}\mathcal{H}(\mathbf{S})\mathbf{s} \equiv \mathcal{H}(\mathbf{S})\mathbf{S}\mathbf{s} \tag{1}$$

Shift invariance

$$\mathbf{S}\mathcal{H}(\mathbf{S}) \equiv \mathcal{H}(\mathbf{S})\mathbf{S} \tag{2}$$

Graph Filters

Cornell University

- To be invariant w.r.t. the GSO a **graph filter** must be a matrix polynomial:

$$\mathcal{H}(\mathbf{S}) \triangleq \sum_{\ell=-\infty}^{+\infty} h_\ell \mathbf{S}^\ell. \tag{3}$$

- Graph/network data are **filtered** graph signals, i.e.,

$$\boldsymbol{x} = \mathcal{H}(\mathbf{S})\boldsymbol{s}. \tag{4}$$

- The signal/observation is $\boldsymbol{x}$ while $\boldsymbol{s}$ is viewed as the **excitation**.

Graph Filters Examples

Cornell University

- Examples of infinite order filters — $\mathcal{H}(\mathbf{S}) = (\mathbb{I} - \alpha\mathbf{S})^{-\mathbf{1}}$, $\mathcal{H}(\mathbf{S}) = \mathbf{e}^{\alpha\mathbf{S}}$.

Special cases for the GSO

The GSO in [2] is the adjacency operator $\mathbf{S} = \mathbf{A}$; in e.g.[3] the GSO is the Laplacian operator $\mathbf{S} = \mathrm{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A}$, but there are other generalizations.

Graph Filters Examples

- Examples of infinite order filters — $\mathcal{H}(\mathbf{S}) = (\mathbb{I} - \alpha\mathbf{S})^{-\mathbf{1}}$, $\mathcal{H}(\mathbf{S}) = \mathbf{e}^{\alpha\mathbf{S}}$.

Special cases for the GSO

The GSO in [2] is the adjacency operator $\mathbf{S} = \mathbf{A}$; in e.g.[3] the GSO is the Laplacian operator $\mathbf{S} = \mathrm{diag}(\mathbf{A1}) - \mathbf{A}$, but there are other generalizations.

Graph Laplacian

Cornell University

- The Laplacian is popular choice for the GSO
  - A differential (not shift operator) - GSO is a misnomer used nonetheless
- Imperfect for models with a a directed graph support
- The Laplacian of a (weighted) graph is $\mathbf{S} = \boldsymbol{B}^T \mathrm{diag}(\boldsymbol{w}) \boldsymbol{B}$
- Note: the Laplacian has always a zero eigenvector equal to $\mathbf{1}$, since $\boldsymbol{B}^T \mathrm{diag}(\boldsymbol{w}) \boldsymbol{B}\mathbf{1} = 0$

Graph Laplacian

Cornell University

- The Laplacian is popular choice for the GSO
  - A differential (not shift operator) - GSO is a misnomer used nonetheless
- Imperfect for models with a a directed graph support
- The Laplacian of a (weighted) graph is $\mathbf{S} = \boldsymbol{B}^T \mathrm{diag}(\boldsymbol{w}) \boldsymbol{B}$
- Note: the Laplacian has always a zero eigenvector equal to $\mathbf{1}$, since $\boldsymbol{B}^T \mathrm{diag}(\boldsymbol{w}) \boldsymbol{B} \mathbf{1} = 0$

Graph Laplacian

Cornell University

- Using the Laplacian, with $0 \leq \gamma \leq \max_i [\mathbf{S}]_{ii}$ the operation

$$(\boldsymbol{x} - \gamma \mathbf{S} \boldsymbol{x}) \mapsto \boldsymbol{x} \qquad \text{or} \qquad \dot{\boldsymbol{x}} = -\mathbf{S}\boldsymbol{x}$$

  reallocates all the edge differences, adding them where an edge enters and subtracting them where an edge leaves the node

- A graph signal subject to this action is progressively *smoothened*

Graph Laplacian

Cornell University

- Using the Laplacian, with $0 \leq \gamma \leq \max_i [\mathbf{S}]_{ii}$ the operation

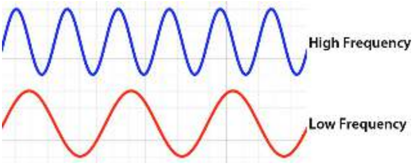
$$(\boldsymbol{x} - \gamma \mathbf{S}\boldsymbol{x}) \mapsto \boldsymbol{x} \qquad \text{or} \qquad \dot{\boldsymbol{x}} = -\mathbf{S}\boldsymbol{x}$$

  reallocates all the edge differences, adding them where an edge enters and subtracting them where an edge leaves the node
- A graph signal subject to this action is progressively *smoothened*

What are low, versus high frequency?

Cornell University

- The idea is to measure the *variation* in adjacent entries of the GFT basis: the larger the variation the higher the frequency.



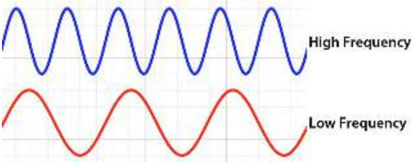
- The total variation based on the squared Frobenius norm of graph derivative of the GS $\boldsymbol{x}$:

$$\mathrm{S}_2(\boldsymbol{x}) := \frac{1}{2}\|\nabla \boldsymbol{x}\|_{\mathrm{F}}^2 = \boldsymbol{x}^\top \mathbf{S}\boldsymbol{x} = \sum_{i,j} S_{ij}(x_i - x_j)^2. \qquad (5)$$

What are low, versus high frequency?

Cornell University

- The idea is to measure the *variation* in adjacent entries of the GFT basis: the larger the variation the higher the frequency.



- The total variation based on the squared Frobenius norm of graph derivative of the GS $\boldsymbol{x}$:

$$\mathrm{S}_2(\boldsymbol{x}) := \frac{1}{2}\|\nabla \boldsymbol{x}\|_{\mathrm{F}}^2 = \boldsymbol{x}^\top \mathbf{S}\boldsymbol{x} = \sum_{i,j} S_{ij}(x_i - x_j)^2. \tag{5}$$

GFT basis construction

Cornell University

- The GS is *smooth* if $\mathrm{S}_2(\boldsymbol{x})/\|\boldsymbol{x}\|_2$ is small.
- Note that the eigenvectors are minima of the ratio $\mathrm{S}_2(\boldsymbol{x})/\|\boldsymbol{x}\|_2$ which is called also Rayleigh quotient for the GSO $\mathbf{S}$:

$$\lambda_1 = \mathrm{S}_2(\boldsymbol{u}_1)/\|\boldsymbol{u}_1\|_2 = 0 = \min_{\boldsymbol{x}} \mathrm{S}_2(\boldsymbol{x})/\|\boldsymbol{x}\|_2$$

$$\lambda_i = \mathrm{S}_2(\boldsymbol{u}_i)/\|\boldsymbol{u}_i\|_2 = \min_{\boldsymbol{x}:\boldsymbol{u}_j\boldsymbol{x}^\top=0, j=1,\ldots,i-1} \mathrm{S}_2(\boldsymbol{x})/\|\boldsymbol{x}\|_2$$

Hence, the eigenvectors $\boldsymbol{U} = (\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_n)$ with $\mathbf{u}_i \in \mathbb{R}^n$ being the eigenvector for λ_i in order $0 = \lambda_1 \leq \cdots \leq \lambda_N$ are the right basis and λ_i is a good parallel of the notion of frequency, going from low to high.

GFT basis construction

Cornell University

- The GS is *smooth* if $\mathrm{S}_2(\boldsymbol{x})/\|\boldsymbol{x}\|_2$ is small.
- Note that the eigenvectors are minima of the ratio $\mathrm{S}_2(\boldsymbol{x})/\|\boldsymbol{x}\|_2$ which is called also Rayleigh quotient for the GSO $\mathbf{S}$:

$$\lambda_1 = \mathrm{S}_2(\boldsymbol{u}_1)/\|\boldsymbol{u}_1\|_2 = 0 = \min_{\boldsymbol{x}} \mathrm{S}_2(\boldsymbol{x})/\|\boldsymbol{x}\|_2$$

$$\lambda_i = \mathrm{S}_2(\boldsymbol{u}_i)/\|\boldsymbol{u}_i\|_2 = \min_{\boldsymbol{x}:\boldsymbol{u}_j\boldsymbol{x}^\top=0, j=1,\ldots,i-1} \mathrm{S}_2(\boldsymbol{x})/\|\boldsymbol{x}\|_2$$

Hence, the eigenvectors $\boldsymbol{U} = (\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_n)$ with $\mathbf{u}_i \in \mathbb{R}^n$ being the eigenvector for λ_i in order $0 = \lambda_1 \leq \cdots \leq \lambda_N$ are the right basis and λ_i is a good parallel of the notion of frequency, going from low to high.

Graph Fourier Transform (GFT)

- The Discrete-time Fourier Transform (DFT) basis are the eigen- vectors of the GSO for the cyclic graph $\rightarrow$ shift invariant signals
- $\mathbf{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1}$. The columns of $\boldsymbol{U}$ are the signals that are invariant w.r.t. the GSO $\mathbf{S}$, in fact $\lambda_i \boldsymbol{u}_i = \mathbf{S}\boldsymbol{u}_i$

$$\tilde{\boldsymbol{x}} = \overbrace{\boldsymbol{U}^{-1}}^{GFT} \boldsymbol{x} \tag{6}$$

- The transfer function of the GF is:

$$\tilde{\mathcal{H}}(\boldsymbol{\Lambda}) = \mathrm{diag}(\tilde{\boldsymbol{h}}) \quad \text{where} \quad [\tilde{\boldsymbol{h}}]_i = \tilde{h}_i = \sum_{\ell} h_\ell \lambda_i^\ell \tag{7}$$

Graph Fourier Transform (GFT)

Cornell University

- The Discrete-time Fourier Transform (DFT) basis are the eigen- vectors of the GSO for the cyclic graph $\rightarrow$ shift invariant signals
- $\mathbf{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1}$. The columns of $\boldsymbol{U}$ are the signals that are invariant w.r.t. the GSO $\mathbf{S}$, in fact $\lambda_i \boldsymbol{u}_i = \mathbf{S}\boldsymbol{u}_i$

$$\tilde{\boldsymbol{x}} = \overbrace{\boldsymbol{U}^{-1}}^{GFT} \boldsymbol{x} \tag{6}$$

- The transfer function of the GF is:

$$\tilde{\mathcal{H}}(\boldsymbol{\Lambda}) = \mathrm{diag}(\tilde{\boldsymbol{h}}) \quad \text{where} \quad [\tilde{\boldsymbol{h}}]_i = \tilde{h}_i = \sum_{\ell} h_\ell \lambda_i^\ell \tag{7}$$

Graph Fourier Transform (GFT)

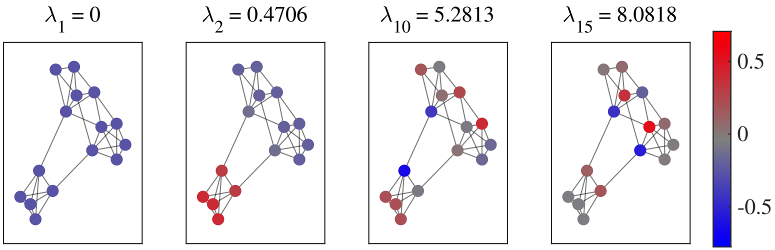
- The Discrete-time Fourier Transform (DFT) basis are the eigen- vectors of the GSO for the cyclic graph $\rightarrow$ shift invariant signals
- $\mathbf{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1}$. The columns of $\boldsymbol{U}$ are the signals that are invariant w.r.t. the GSO $\mathbf{S}$, in fact $\lambda_i \boldsymbol{u}_i = \mathbf{S}\boldsymbol{u}_i$

$$\tilde{\boldsymbol{x}} = \overbrace{\boldsymbol{U}^{-1}}^{GFT} \boldsymbol{x} \tag{6}$$

- The transfer function of the GF is:

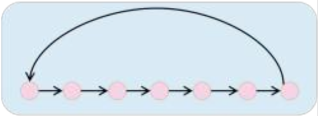
$$\tilde{\mathcal{H}}(\boldsymbol{\Lambda}) = \mathrm{diag}(\tilde{\boldsymbol{h}}) \quad \text{where} \quad [\tilde{\boldsymbol{h}}]_i = \tilde{h}_i = \sum_{\ell} h_\ell \lambda_i^\ell \tag{7}$$

The GFT basis behavior



The GFT basis, $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_{10}, \mathbf{u}_{15}$, associated to the graph Laplacian of an undirected, unweighted graph with 15 nodes

Consistency with DSP



$$\rightarrow \quad \boldsymbol{A} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \end{pmatrix}$$

- This choice is consistent with the DFT for the cycle graph.
- In fact for the cycle graph the Laplacian is a circulant matrix:

$$\mathbf{S} = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

and the eigenvectors and eigenvalues of $\mathbf{S}$, indexed by $k = 0, \dots, N-1$:

$$\boldsymbol{u}_k = (1, \rho_k, \dots, \rho_k^{N-1})^\top, \quad \rho_k = \frac{1}{\sqrt{N}} e^{-j\frac{2\pi k}{N}}, \quad \lambda_k = 2(1 - \cos(2\pi k/N))$$

GFT filters input/output relatioship

Cornell University

- We said the transfer function of the GF $\mathcal{H}(\mathbf{S})$ is:

$$\tilde{\mathcal{H}}(\mathbf{\Lambda}) = \mathrm{diag}(\tilde{\boldsymbol{h}}) \quad \text{where} \quad [\tilde{\boldsymbol{h}}]_i = \tilde{h}_i = \sum_k h_k \lambda_i^k \tag{8}$$

- In fact, we recover the familiar relationship between the input and output of a filter GFTs, as the product with the filter transfer function:

$$\mathcal{H}(\mathbf{S}) = \sum_k h_k \mathbf{S}^k = \boldsymbol{U} \overbrace{\left(\sum_k h_k \mathbf{\Lambda}^k\right)}^{\tilde{\mathcal{H}}(\mathbf{\Lambda}) = \mathrm{Diag}(\tilde{\boldsymbol{h}})} \boldsymbol{U}^{-1}$$

$$\boldsymbol{x} = \mathcal{H}(\mathbf{S})\boldsymbol{s} \rightarrow \overbrace{\boldsymbol{U}^{-1}\boldsymbol{x}}^{\tilde{\boldsymbol{x}}} = \overbrace{\boldsymbol{U}^{-1}\boldsymbol{U}}^{\boldsymbol{I}} \mathrm{Diag}(\tilde{\boldsymbol{h}}) \overbrace{\boldsymbol{U}^{-1}\boldsymbol{s}}^{\tilde{\boldsymbol{s}}}$$

$$\rightarrow \tilde{\boldsymbol{x}} = \mathrm{Diag}(\tilde{\boldsymbol{h}})\tilde{\boldsymbol{s}},$$

- As a convention, we assume $0 = \lambda_1 \leq \cdots \leq \lambda_N$.

GFT filters input/output relatioship

Cornell University

- We said the transfer function of the GF $\mathcal{H}(\mathbf{S})$ is:

$$\tilde{\mathcal{H}}(\mathbf{\Lambda}) = \mathrm{diag}(\tilde{\boldsymbol{h}}) \quad \text{where} \quad [\tilde{\boldsymbol{h}}]_i = \tilde{h}_i = \sum_k h_k \lambda_i^k \tag{8}$$

- In fact, we recover the familiar relationship between the input and output of a filter GFTs, as the product with the filter transfer function:

$$\mathcal{H}(\mathbf{S}) = \sum_k h_k \mathbf{S}^k = \boldsymbol{U} \overbrace{\left(\sum_k h_k \mathbf{\Lambda}^k\right)}^{\tilde{\mathcal{H}}(\mathbf{\Lambda}) = \mathrm{Diag}(\tilde{\boldsymbol{h}})} \boldsymbol{U}^{-1}$$

$$\boldsymbol{x} = \mathcal{H}(\mathbf{S})\boldsymbol{s} \;\rightarrow\; \overbrace{\boldsymbol{U}^{-1}\boldsymbol{x}}^{\tilde{\boldsymbol{x}}} = \overbrace{\boldsymbol{U}^{-1}\boldsymbol{U}}^{\boldsymbol{I}} \,\mathrm{Diag}(\tilde{\boldsymbol{h}})\, \overbrace{\boldsymbol{U}^{-1}\boldsymbol{s}}^{\tilde{\boldsymbol{s}}}$$

$$\rightarrow \tilde{\boldsymbol{x}} = \mathrm{Diag}(\tilde{\boldsymbol{h}})\tilde{\boldsymbol{s}},$$

- As a convention, we assume $0 = \lambda_1 \leq \cdots \leq \lambda_N$.

GFT filters input/output relatioship

- We said the transfer function of the GF $\mathcal{H}(\mathbf{S})$ is:

$$\tilde{\mathcal{H}}(\mathbf{\Lambda}) = \mathrm{diag}(\tilde{\boldsymbol{h}}) \quad \text{where} \quad [\tilde{\boldsymbol{h}}]_i = \tilde{h}_i = \sum_k h_k \lambda_i^k \tag{8}$$

- In fact, we recover the familiar relationship between the input and output of a filter GFTs, as the product with the filter transfer function:

$$\mathcal{H}(\mathbf{S}) = \sum_k h_k \mathbf{S}^k = \boldsymbol{U} \overbrace{\left(\sum_k h_k \mathbf{\Lambda}^k \right)}^{\tilde{\mathcal{H}}(\mathbf{\Lambda}) = \mathrm{Diag}(\tilde{\boldsymbol{h}})} \boldsymbol{U}^{-1}$$

$$\boldsymbol{x} = \mathcal{H}(\mathbf{S})\boldsymbol{s} \;\; \rightarrow \;\; \overbrace{\boldsymbol{U}^{-1}\boldsymbol{x}}^{\tilde{\boldsymbol{x}}} = \overbrace{\boldsymbol{U}^{-1}\boldsymbol{U}}^{\boldsymbol{I}} \mathrm{Diag}(\tilde{\boldsymbol{h}}) \overbrace{\boldsymbol{U}^{-1}\boldsymbol{s}}^{\tilde{\boldsymbol{s}}}$$
$$\rightarrow \tilde{\boldsymbol{x}} = \mathrm{Diag}(\tilde{\boldsymbol{h}})\tilde{\boldsymbol{s}},$$

- As a convention, we assume $0 = \lambda_1 \leq \cdots \leq \lambda_N$.

Outline

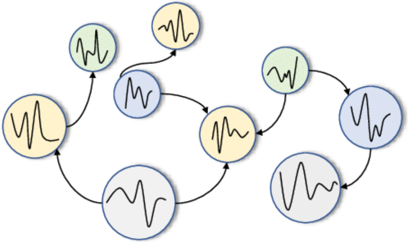
Cornell University

Graph Temporal Filters

Cornell University

- In many situations we observe time series of graph signals $\boldsymbol{x}_t$
- A graph-temporal filter with the impulse response:

$$\mathcal{H}(\mathbf{S}, t) := \sum_{\ell=-\infty}^{+\infty} h_{t,\ell}\mathbf{S}^{\ell},$$



- In the time domain:

$$\boldsymbol{x}_t = \sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S}, k)\boldsymbol{s}_{t-k} \equiv \sum_{k=-\infty}^{+\infty} \sum_{\ell=-\infty}^{+\infty} h_{k,\ell}\mathbf{S}^{\ell}\boldsymbol{s}_{t-k}$$

Graph temporal filters

Cornell University

- The spatio-temporal GSO is $\mathbf{S}z^{-1}$. If we take the z-transform only:

$$\boldsymbol{x}_t = \sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k)\boldsymbol{s}_{t-k}, \quad \mathcal{H}(\mathbf{S};k) = \sum_{\ell=0}^{L} h_{k,\ell}\mathbf{S}^{\ell}$$

$$\rightarrow \overbrace{\sum_t \boldsymbol{x}_t z^{-t}}^{\boldsymbol{x}(z)} = \sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k) \overbrace{\sum_t \boldsymbol{s}_{t-k} z^{-t}}^{\boldsymbol{s}(z)z^{-k}} \rightarrow \boldsymbol{x}(z) = \overbrace{\left(\sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k) z^{-k} \right)}^{\boldsymbol{h}(\mathbf{S};z)} \boldsymbol{s}(z)$$

$$\boldsymbol{h}(\mathbf{S};z) := \sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k) z^{-k} = \sum_{k=-\infty}^{+\infty} \sum_{\ell=-\infty}^{+\infty} h_{k,\ell} \boldsymbol{S}^{\ell} z^{-k} = \sum_{\ell=-\infty}^{+\infty} \boldsymbol{S}^{\ell} \overbrace{\left(\sum_{k=-\infty}^{+\infty} h_{k,\ell} z^{-k} \right)}^{h_\ell(z)}$$

Finally, in the z-domain, the graph filter response is:

$$\boldsymbol{x}(z) = \boldsymbol{h}(\mathbf{S};z)\boldsymbol{s}(z) = \sum_{\ell=-\infty}^{+\infty} h_\ell(z) \boldsymbol{S}^{\ell} \boldsymbol{s}(z)$$

Graph temporal filters

Cornell University

- The spatio-temporal GSO is $\mathbf{S}z^{-1}$. If we take the z-transform only:

$$\boldsymbol{x}_t = \sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k)\boldsymbol{s}_{t-k}, \quad \mathcal{H}(\mathbf{S};k) = \sum_{\ell=0}^{L} h_{k,\ell}\mathbf{S}^{\ell}$$

$$\rightarrow \overbrace{\sum_t \boldsymbol{x}_t z^{-t}}^{\boldsymbol{x}(z)} = \sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k) \overbrace{\sum_t \boldsymbol{s}_{t-k} z^{-t}}^{\boldsymbol{s}(z)z^{-k}} \rightarrow \boldsymbol{x}(z) = \overbrace{\left(\sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k) z^{-k}\right)}^{\boldsymbol{h}(\mathbf{S};z)} \boldsymbol{s}(z)$$

$$\boldsymbol{h}(\mathbf{S};z) := \sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k) z^{-k} = \sum_{k=-\infty}^{+\infty} \sum_{\ell=-\infty}^{+\infty} h_{k,\ell}\boldsymbol{S}^{\ell} z^{-k} = \sum_{\ell=-\infty}^{+\infty} \boldsymbol{S}^{\ell} \overbrace{\left(\sum_{k=-\infty}^{+\infty} h_{k,\ell} z^{-k}\right)}^{h_\ell(z)}$$

Finally, in the z-domain, the graph filter response is:

$$\boldsymbol{x}(z) = \boldsymbol{h}(\mathbf{S};z)\boldsymbol{s}(z) = \sum_{\ell=-\infty}^{+\infty} h_\ell(z)\boldsymbol{S}^{\ell}\boldsymbol{s}(z)$$

Graph temporal filters

Cornell University

- The spatio-temporal GSO is $\mathbf{S}z^{-1}$. If we take the z-transform only:

$$\boldsymbol{x}_t = \sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k)\boldsymbol{s}_{t-k}, \quad \mathcal{H}(\mathbf{S};k) = \sum_{\ell=0}^{L} h_{k,\ell}\mathbf{S}^{\ell}$$

$$\rightarrow \overbrace{\sum_t \boldsymbol{x}_t z^{-t}}^{\boldsymbol{x}(z)} = \sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k) \overbrace{\sum_t \boldsymbol{s}_{t-k} z^{-t}}^{\boldsymbol{s}(z)z^{-k}} \rightarrow \boldsymbol{x}(z) = \overbrace{\left(\sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k) z^{-k}\right)}^{\boldsymbol{h}(\mathbf{S};z)} \boldsymbol{s}(z)$$

$$\boldsymbol{h}(\mathbf{S};z) := \sum_{k=-\infty}^{+\infty} \mathcal{H}(\mathbf{S};k) z^{-k} = \sum_{k=-\infty}^{+\infty} \sum_{\ell=-\infty}^{+\infty} h_{k,\ell} \boldsymbol{S}^{\ell} z^{-k} = \sum_{\ell=-\infty}^{+\infty} \boldsymbol{S}^{\ell} \overbrace{\left(\sum_{k=-\infty}^{+\infty} h_{k,\ell} z^{-k}\right)}^{h_\ell(z)}$$

Finally, in the z-domain, the graph filter response is:

$$\boldsymbol{x}(z) = \boldsymbol{h}(\mathbf{S};z)\boldsymbol{s}(z) = \sum_{\ell=-\infty}^{+\infty} h_\ell(z) \boldsymbol{S}^{\ell} \boldsymbol{s}(z)$$

Graph-z transform

Cornell University

- Having define the z-transform of the Graph Temporal filter, we now can define the joint Graph-z transform.
- Let $U_{\mathcal{V}}$ be the GFT basis of the spatial GSO $\mathbf{S} = U_{\mathcal{V}} \Lambda U_{\mathcal{V}}^{-1}$, of the graph temporal filters. We have:

$$\widetilde{x}(z) = U_{\mathcal{V}}^{-1} x(z) = U_{\mathcal{V}}^{-1} h(\mathbf{S}; z) s(z) = \sum_{\ell=-\infty}^{+\infty} h_{\ell}(z) U_{\mathcal{V}}^{-1} S^{\ell} s(z)$$

$$= \sum_{\ell=-\infty}^{+\infty} h_{\ell}(z) U_{\mathcal{V}}^{-1} \overbrace{(U_{\mathcal{V}} \Lambda U_{\mathcal{V}}^{-1})^{\ell}}^{=U_{\mathcal{V}} \Lambda^{\ell} U_{\mathcal{V}}^{-1}} s(z) = \overbrace{\sum_{\ell=-\infty}^{+\infty} h_{\ell}(z) \Lambda^{\ell}}^{\widetilde{h}(z)} \overbrace{U_{\mathcal{V}}^{-1} s(z)}^{\tilde{s}(z)}$$

$$= \widetilde{h}(z) \odot \widetilde{s}(z)$$

and $\tilde{h}(z)$ is the graph-temporal joint transfer function:

$$[\tilde{h}(z)]_i = \mathbb{H}(\lambda_i, z), \quad \text{where} \quad \mathbb{H}(\lambda, z) := \sum_{t=0}^{\infty} \sum_{k=0}^{K-1} h_{k,t} \lambda^k z^{-t}$$

Graph-z transform

Cornell University

- Having define the z-transform of the Graph Temporal filter, we now can define the joint Graph-z transform.
- Let $\boldsymbol{U}_{\mathcal{V}}$ be the GFT basis of the spatial GSO $\mathbf{S} = \boldsymbol{U}_{\mathcal{V}} \boldsymbol{\Lambda} \boldsymbol{U}_{\mathcal{V}}^{-1}$, of the graph temporal filters. We have:

$$
\begin{aligned}
\widetilde{\boldsymbol{x}}(z) &= \boldsymbol{U}_{\mathcal{V}}^{-1} \boldsymbol{x}(z) = \boldsymbol{U}_{\mathcal{V}}^{-1} \boldsymbol{h}(\mathbf{S}; z) \boldsymbol{s}(z) = \sum_{\ell=-\infty}^{+\infty} h_\ell(z) \boldsymbol{U}_{\mathcal{V}}^{-1} \boldsymbol{S}^\ell \boldsymbol{s}(z) \\
&= \sum_{\ell=-\infty}^{+\infty} h_\ell(z) \boldsymbol{U}_{\mathcal{V}}^{-1} \overbrace{(\boldsymbol{U}_{\mathcal{V}} \boldsymbol{\Lambda} \boldsymbol{U}_{\mathcal{V}}^{-1})^\ell}^{= \boldsymbol{U}_{\mathcal{V}} \boldsymbol{\Lambda}^\ell \boldsymbol{U}_{\mathcal{V}}^{-1}} \boldsymbol{s}(z) = \overbrace{\sum_{\ell=-\infty}^{+\infty} h_\ell(z) \boldsymbol{\Lambda}^\ell}^{\widetilde{\boldsymbol{h}}(z)} \overbrace{\boldsymbol{U}_{\mathcal{V}}^{-1} \boldsymbol{s}(z)}^{\tilde{\boldsymbol{s}}(z)} \\
&= \widetilde{\boldsymbol{h}}(z) \odot \widetilde{\boldsymbol{s}}(z)
\end{aligned}
$$

and $\tilde{\boldsymbol{h}}(z)$ is the graph-temporal joint transfer function:

$$
[\tilde{\boldsymbol{h}}(z)]_i = \mathbb{H}(\lambda_i, z), \quad \text{where} \quad \mathbb{H}(\lambda, z) := \sum_{t=0}^{\infty} \sum_{k=0}^{K-1} h_{k,t} \lambda^k z^{-t}
$$

GF-ARMA (q,r) filters

Cornell University

- For graph signal processes a useful model is the GF-ARMA (q, r) filter
- Its input-output relation in time domain and z-GFT domain are:

$$\boldsymbol{x}_t - \mathcal{A}_1(\mathrm{S})\boldsymbol{x}_{t-1} \cdots - \mathcal{A}_q(\mathrm{S})\boldsymbol{x}_{t-q} = \mathcal{B}_0(\mathrm{S})\boldsymbol{s}_t + \cdots + \mathcal{B}_r(\mathrm{S})\boldsymbol{s}_{t-r},$$

$$\widetilde{\boldsymbol{a}}(z) \odot \widetilde{\boldsymbol{x}}(z) = \widetilde{\boldsymbol{b}}(z) \odot \widetilde{\boldsymbol{s}}(z),$$

Outline

Cornell University

Graph signals with sparse GFT

Cornell University

- Like LTI filters, a graph filter can be classified as either low-pass, band-pass, or high-pass, or having a sparse support, depending on its graph frequency response
- Like in parametric spectrum estimation, low-pass graph signals can be viewed as the output of a low-pass graph filter with a excitation signal
- While generally speaking all such graph signals generate low-rank sequences of samples, are worth focusing on because:
  1. they are common in practice;
  2. the subspace vectors have interesting properties tied to the community structure of the graph

Graph signals with sparse GFT

Cornell University

- Like LTI filters, a graph filter can be classified as either low-pass, band-pass, or high-pass, or having a sparse support, depending on its graph frequency response
- Like in parametric spectrum estimation, low-pass graph signals can be viewed as the output of a low-pass graph filter with a excitation signal
- While generally speaking all such graph signals generate low-rank sequences of samples, are worth focusing on because:
  1. they are common in practice;
  2. the subspace vectors have interesting properties tied to the community structure of the graph

Low Pass Graph Filter

Cornell University

- The definition of Ideal Low-Pass Graph Filter is rather obvious: for that $h(\lambda) = 1, \lambda \leq \lambda_k$ and 0 otherwise.
- Unfortunately, this is rather strict in practical models. Most GS models are amenable to the following looser definition:

Definition of Low-Pass Graph Signal

For any $1 \leq k \leq n-1$, define the ratio

$$\eta_k := \frac{\max\{|h(\lambda_{k+1})|, \ldots, |h(\lambda_n)|\}}{\min\{|h(\lambda_1)|, \ldots, |h(\lambda_k)|\}}. \tag{9}$$

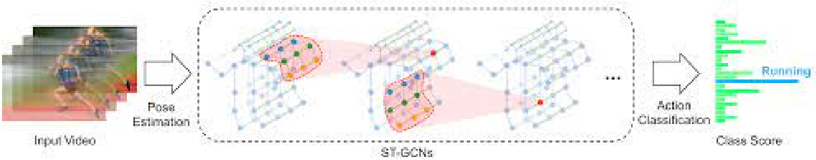
The graph filter $\mathcal{H}(\mathbf{S})$ is k-low-pass if and only if the low-pass ratio η_k satisfies $\eta_k \in [0, 1)$. Upon passing a graph signal through $\mathcal{H}(\mathbf{S})$, the high frequency components (above λ_k) are attenuated by a factor of less than or equal to η_k.

- The integer parameter k characterizes the *bandwidth*, or the cut-off frequency of the low-pass filter is at λ_k

Conclusions

Cornell University

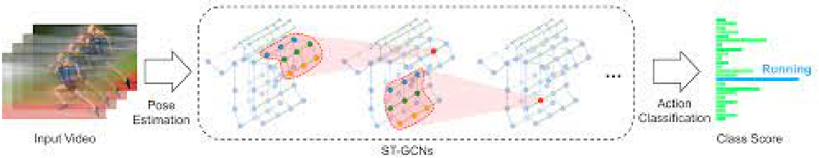
- We introduced GSP
- We mentioned how frequently data exhibit a low pass nature that is due to the inherent structure of the network diffusion process
- Overall, for ML the key point is to be able to do feature extraction and for graph signals this requires identifying the right GSO



Conclusions

Cornell University

- We introduced GSP
- We mentioned how frequently data exhibit a low pass nature that is due to the inherent structure of the network diffusion process
- Overall, for ML the key point is to be able to do feature extraction and for graph signals this requires identifying the right GSO



References

[1] Markus Püschel and José MF Moura.
The algebraic approach to the discrete cosine and sine transforms and their fast algorithms.
SIAM Journal on Computing, 32(5):1280–1316, 2003.

[2] Aliaksei Sandryhaila and José MF Moura.
Discrete signal processing on graphs.
IEEE transactions on signal processing, 61(7):1644–1656, 2013.

[3] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst.
The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains.
IEEE Signal Processing Magazine, 30(3):83–98, 2013.