Graph Signal Processing - Part II: Processing and Analyzing Signals on Graphs

McGill

Jiaqi Zhu
Oct. 26th

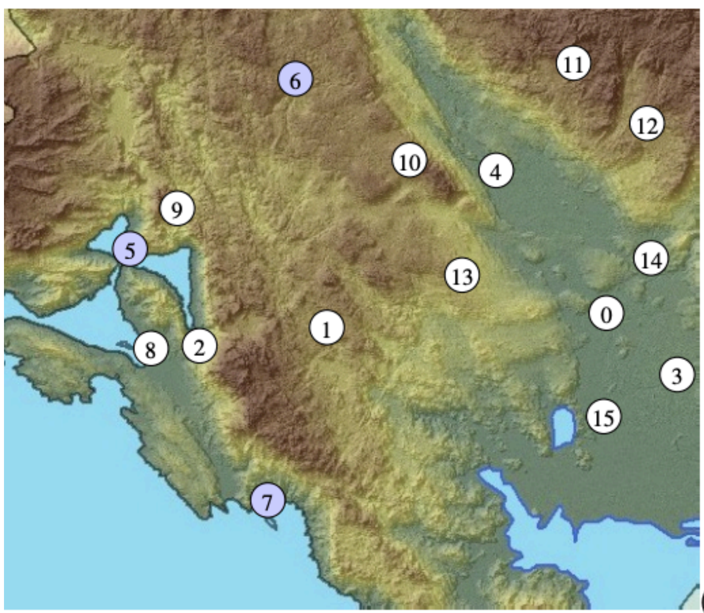
Outline

- Motivation: Case Study
- Signals and Systems on Graphs

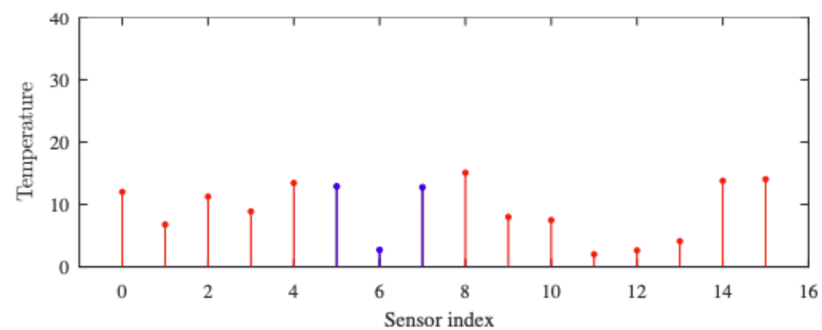
Motivation: Case Study

Problem Setup

- Consider a multi-sensor setup for measuring a temperature field in a region of interest.
- The temperature sensing locations are chosen according to the significance of a particular geographic area to local users (with N = 16 sensing points in total, as shown in figure (a))
- The temperature field is denoted by $\{x(n)\}$, with n as the sensor index, while a snapshot of its values is given in figure (b).
- Each measured sensor signal can then be mathematically expressed as $x(n) = s(n) + \varepsilon(n)$, $n = 0, 1, \ldots, 15$, where $s(n)$ is the true temperature that would have been obtained in ideal measuring conditions, and $\varepsilon(n)$ is the noise, where $\varepsilon(n) \in \mathcal{N}(0, 4)$. This yields the signal-to-noise ratio in in $x(n)$ of $SNR_{in} = 14.2$ dB



(a)



(b)

Motivation: Case Study

Goal:

- Denoise and increase SNR
  - **5 dB to 10 dB**: is below the minimum level to establish a connection, due to the noise level being nearly indistinguishable from the desired signal (useful information).
  - **10 dB to 15 dB**: is the accepted minimum to establish an unreliable connection.
  - **15 dB to 25 dB**: is typically considered the minimally acceptable level to establish poor connectivity.
  - **25 dB to 40 dB**: is deemed to be good.
  - **41 dB or higher**: is considered to be excellent.

(https://resources.pcb.cadence.com/blog/2020-what-is-signal-to-noise-ratio-and-how-to-calculate-it)

Motivation: Case Study

$\frac{x(n)}{|N_n|}$

Noise Reduction Scheme:

- "Situation-Aware": local signal average operator

Cumulative Temperature:

$$y(n) = \sum_{m \text{ at and around } n} x(m)$$

Example:

$S(3)$

n = 3 (low land)
n = 6(mountains)
n = 8 (coast)

y(3) = x(3) + x(0) + x(14) + x(15)
y(6) = x(6) + x(9) + x(10)

⬇

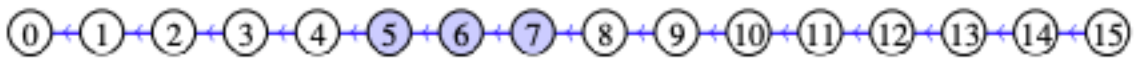
y = x + Ax
A is the *adjacency matrix*

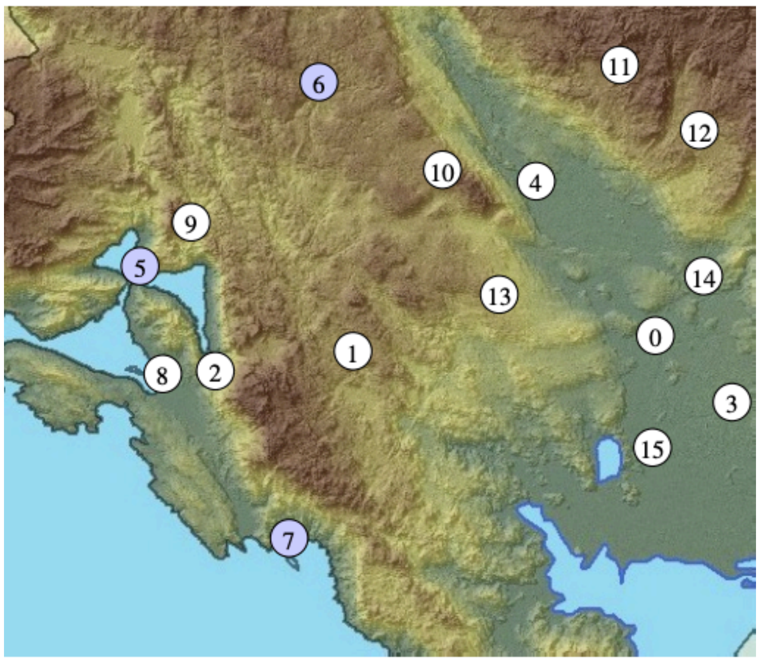
McGill

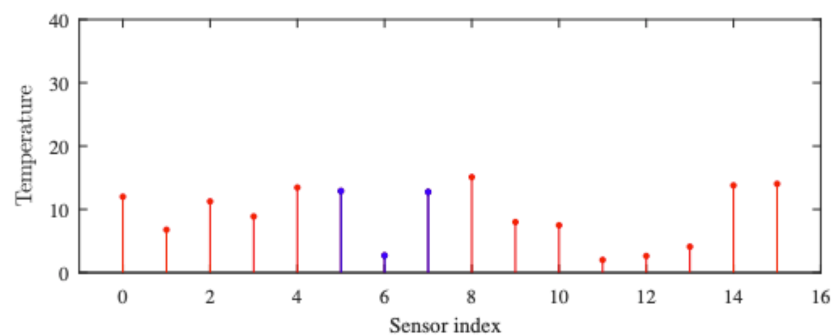
Motivation: Case Study

Classical Representation

- Linear structure







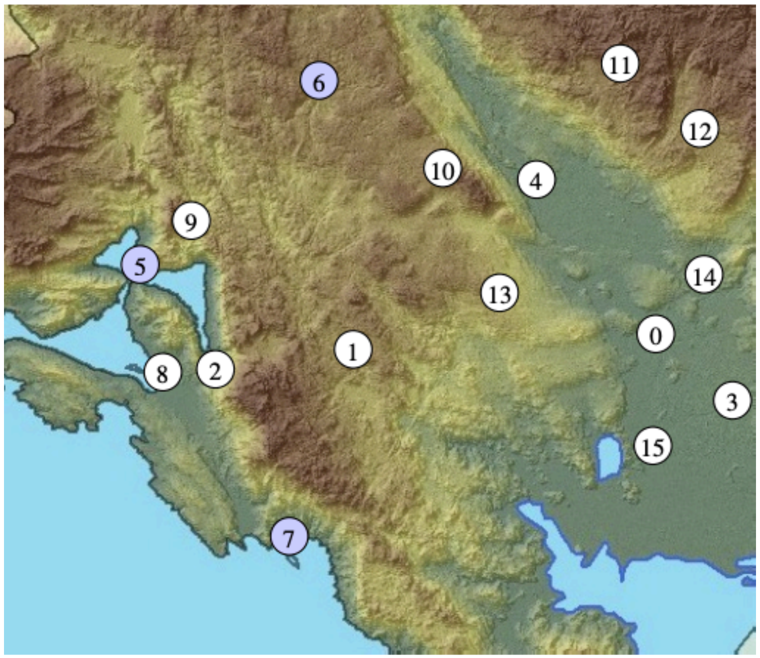
Motivation: Case Study

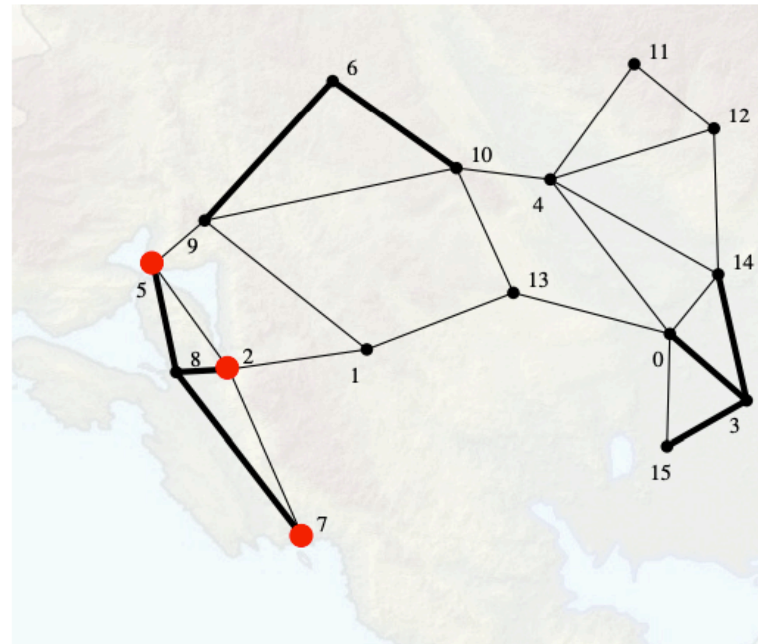
Graph Representation:

- Construct a graph
  - Vertices: sensing points where the signal is measured
  - Edges: Vertex-to-vertex lines which indicate physically meaningful connectivity among the sensing points

  We will use graph (rather than a standard vector of sensing points) to analyze and process data, since it exhibits both spatial and physical domain awareness

Motivation: Case Study

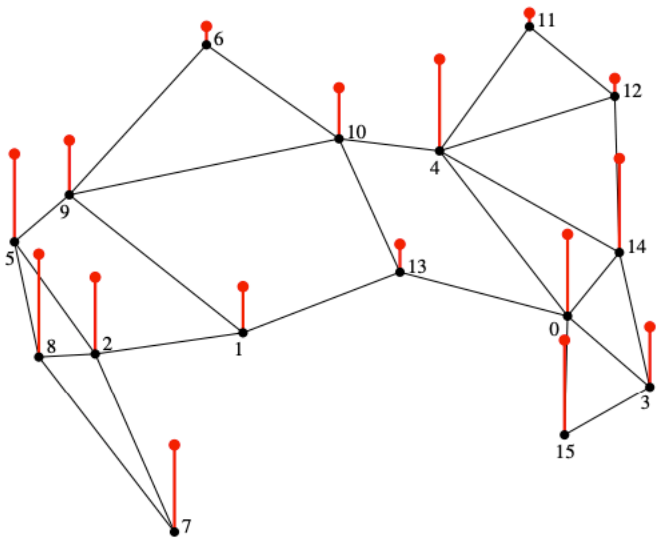


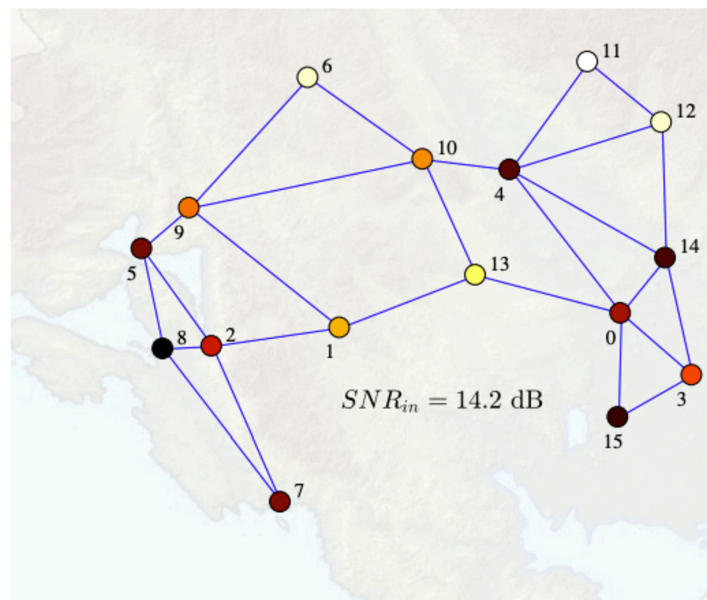


Motivation: Case Study

Graph Representation:

- Signal on Graph
    - The measured temperatures are now interpreted as signal samples on graph
    - Similar to traditional signal processing, this new graph signal may have many realizations on the same graph and may comprise noise





McGill

Motivation: Case Study

Graph Representation:

- System on Graph
  - **y = x + Ax** defines a simple system on a graph for local signal averaging
  - To model sensor relevance, we can also adapt a weighting scheme

$$y(n) = x(n) + \sum_{m \neq n} W_{nm} x(m)$$

where W_{nm} are the elements of the weighting matrix, W

  - Three ways to define graph edges and their corresponding weights (will be introduced in the next series of lectures)
    - Already physically well defined edges and weights
    - **Definition of edges and weights based on the geometry of vertex positions**
    - Data similarity based methods for learning the underlying graph topology

$$W_{mn} = e^{-\alpha r_{mn} - \beta h_{mn}}$$

where α and β are suitable constants.

Motivation: Case Study

Graph Representation:

- System on Graph
  - Weighted graph signal estimator

$$\mathbf{y} = \mathbf{x} + \mathbf{W}\mathbf{x}$$

  - Normalized estimator: weighting coefficients sum to 1 for each y(n)

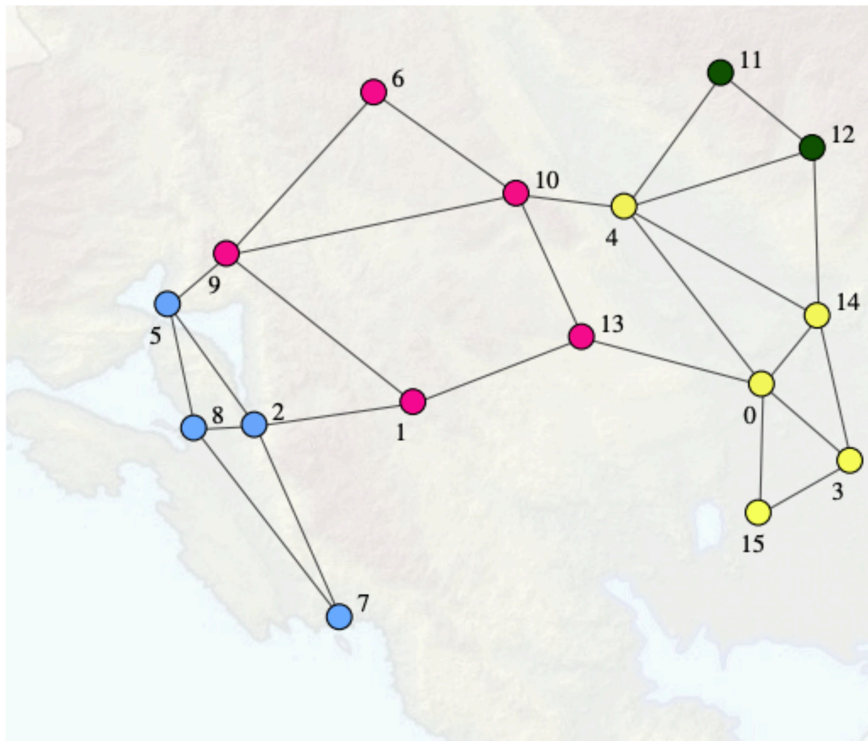
$$\mathbf{y} = \frac{1}{2}(\mathbf{x} + \mathbf{D}^{-1}\mathbf{W}\mathbf{x})$$

where D is the degree matrix, $D_{nn} = \sum_m W_{nm}$

Motivation: Case Study

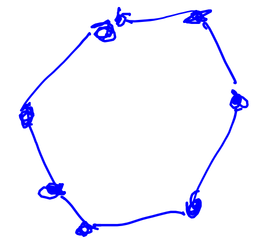
Graph Representation:

- Vertices Clustering



Signals and Systems on Graphs

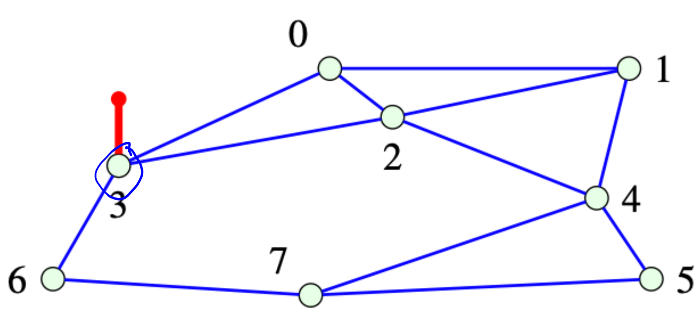
- Basic Concepts:
  - Signals on Graphs:



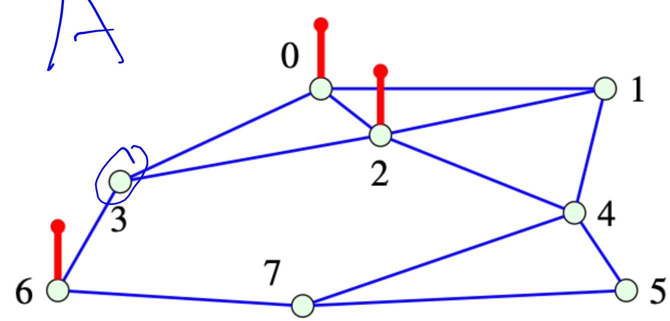
$$\mathbf{x} = [x(0), x(1), \ldots, x(N-1)]^T$$

  - Graph Signal Shift:
    - Walk with length 1: $\mathbf{x}_1 = \mathbf{A}\mathbf{x}$
    - Walk with length 2: $\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 = \mathbf{A}(\mathbf{A}\,\mathbf{x}) = \mathbf{A}^2\,\mathbf{x}$
    - Walk with length m: $\mathbf{x}_m = \mathbf{A}\mathbf{x}_{m-1} = \mathbf{A}^m\,\mathbf{x}$
  - Graph shift operator: a local operator that replaces a signal value at each node of a graph with a linear combinations of the signal values at the neighborhood of that value.

A



(a)



(b)

Signals and Systems on Graphs

- Basic Concepts
  - Systems on Graphs with Shift Operator as Adjacency Matrix:

$$\mathbf{y} = h_0\mathbf{A}^0\mathbf{x} + h_1\mathbf{A}^1\mathbf{x} + \cdots + h_{M-1}\mathbf{A}^{M-1}\mathbf{x} = \sum_{m=0}^{M-1} h_m \mathbf{A}^m \mathbf{x}$$

where $A^0 = I$, by definition, and $h_0, h_1, \ldots, h_{M-1}$ are the system coefficients

I

    - Remark: a physically meaningful system order $(M - 1)$ should be lower than the number of vertices N, that is is, $M \leq N$

  - General System of Graphs (in vertex domain):

$$\mathbf{y} = H(\mathbf{A})\mathbf{x}$$

where $H(A)$ is a vertex domain system (filter) function

    - Linearity: $H(\mathbf{A})(a_1\mathbf{x}_1 + a_2\mathbf{x}_2) = a_1\mathbf{y}_1 + a_2\mathbf{y}_2$
    - Shift Invariance: $H(\mathbf{A})[\mathbf{A}\mathbf{x}] = \mathbf{A}[H(\mathbf{A})\mathbf{x}] = \mathbf{A}\mathbf{y}$

Signals and Systems on Graphs

- Spectral Domain of the Adjacency Matrix

$A = U \Lambda U^T$

  - Graph discrete Fourier transform (GDFT)

$$\mathbf{X} = \mathbf{U}^{-1}\mathbf{x}$$

where X denotes a vector of the GDFT coefficients, and U is a matrix whose columns represent the eigenvectors of the adjacency matrix

    - For undirected graph: $U^{-1} = U^{T}$

signal

$$X(k) = \sum_{n=0}^{N-1} x(n) u_k(n)$$

A

  - Inverse graph discrete Fourier transform (IGDFT)

$$\mathbf{x} = \mathbf{U}\,\mathbf{X}$$

$$x(n) = \sum_{k=0}^{N-1} X(k) u_k(n)$$

Signals and Systems on Graphs

- System on a graph in the GDFT domain

$$\mathbf{y} = h_0\mathbf{A}^0\mathbf{x} + h_1\mathbf{A}^1\mathbf{x} + \cdots + h_{M-1}\mathbf{A}^{M-1}\mathbf{x} = \sum_{m=0}^{M-1} h_m\mathbf{A}^m\mathbf{x}$$

$A = U\Lambda U^{-1}$

$$\Rightarrow \mathbf{y} = H(\mathbf{A})\mathbf{x} = \left(h_0\mathbf{A}^0 + h_1\mathbf{A}^1 + \cdots + h_{M-1}\mathbf{A}^{M-1}\right)\mathbf{x}$$

$$\Rightarrow \mathbf{y} = \left(h_0\mathbf{U}\boldsymbol{\Lambda}^0\mathbf{U}^{-1} + h_1\mathbf{U}\boldsymbol{\Lambda}^1\mathbf{U}^{-1} + \cdots + h_{M-1}\mathbf{U}\boldsymbol{\Lambda}^{M-1}\mathbf{U}^{-1}\right)\mathbf{x}$$

$$= \mathbf{U}(h_0\boldsymbol{\Lambda}^0 + h_1\boldsymbol{\Lambda}^1 + \cdots + h_{M-1}\boldsymbol{\Lambda}^{M-1})\mathbf{U}^{-1}\mathbf{x}$$

$$= \mathbf{U}\,H(\boldsymbol{\Lambda})\mathbf{U}^{-1}\mathbf{x}$$

with $H(\boldsymbol{\Lambda}) = h_0\boldsymbol{\Lambda}^0 + h_1\boldsymbol{\Lambda}^1 + \cdots + h_{M-1}\boldsymbol{\Lambda}^{M-1}$

$$\Rightarrow \mathbf{U}^{-1}\mathbf{y} = H(\boldsymbol{\Lambda})\mathbf{U}^{-1}\mathbf{x} \Leftrightarrow \mathbf{Y} = H(\boldsymbol{\Lambda})\,\mathbf{X}$$

$$Y(k) = (h_0 + h_1\lambda_k + \cdots + h_{M-1}\lambda_k^{M-1})X(k)$$

GDFT of output signal y

The output graph signal in the vertex domain can then be calculated as

$$\mathbf{y} = H(\mathbf{A})\mathbf{x} = \mathrm{IGDFT}\{H(\boldsymbol{\Lambda})\,\mathbf{X}\}$$

X =

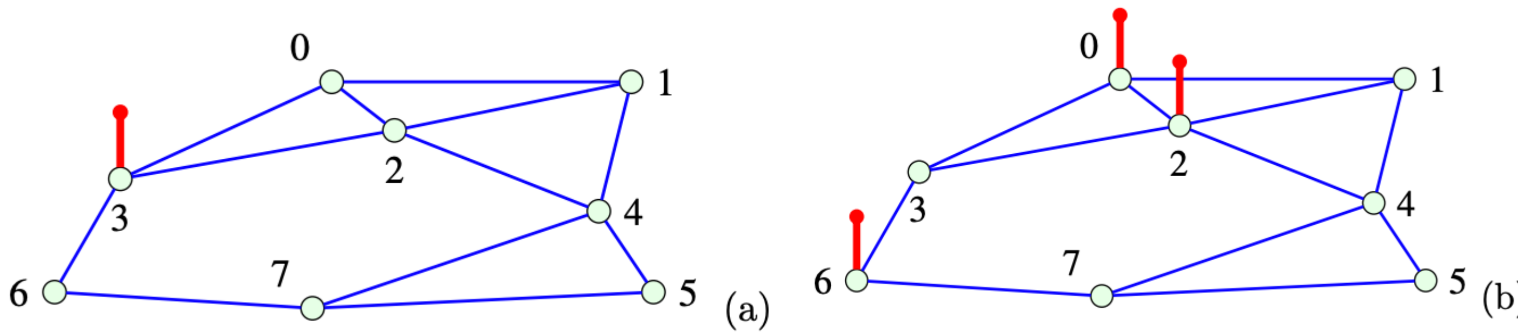
Transfer function of a system on a graph:

$$H(\lambda_k) = \frac{Y(k)}{X(k)} = h_0 + h_1\lambda_k + \cdots + h_{M-1}\lambda_k^{M-1}$$

Signals and Systems on Graphs

- Graph signal filtering
  - Energy of a graph shifted signal

$$\|\mathbf{x}_1\|_2^2 = \|\mathbf{A}\mathbf{x}\|_2^2$$



In general $\|\mathbf{A}\mathbf{x}\|_2^2 \neq \|\mathbf{x}\|_2^2$

$$\max\{\frac{\|\mathbf{A}\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2}\} = \max\{\frac{\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}}{\|\mathbf{x}\|_2^2}\} = \lambda_{\max}^2$$, where $\lambda_{\max} = \max_k |\lambda_k|,\ k = 0, 1, \ldots, N-1.$

Signals and Systems on Graphs

- Graph signal filtering
  - Normalization of the Adjacency of Matrix

$$\mathbf{A}_{norm} = \frac{1}{\lambda_{\max}}\mathbf{A} \quad \Longrightarrow \quad \|\mathbf{A}_{norm}\mathbf{x}\|_2^2 \le \|\mathbf{x}\|_2^2$$

  - Spectral Ordering: low-varying and high-varying eigenvectors
    - First graph difference

$$\mathbf{\Delta x} = \mathbf{x} - \mathbf{x}_1 = \mathbf{x} - \mathbf{A}_{norm}\mathbf{x}.$$

    - Energy of signal change

$$E_{\Delta x} = \|\mathbf{x} - \mathbf{A}_{norm}\mathbf{x}\|_2^2 = \left\|\mathbf{x} - \frac{1}{\lambda_{\max}}\mathbf{A}\mathbf{x}\right\|_2^2$$

$$\mathrm{x} = \mathrm{u} \quad \Downarrow$$

$$E_{\Delta u} = \left\|\mathbf{u} - \frac{1}{\lambda_{\max}}\lambda\mathbf{u}\right\|_2^2 = \left|1 - \frac{\lambda}{\lambda_{\max}}\right|^2$$

$\lambda_{\max}$

two-norm total variation of a basis function/eigenvector.

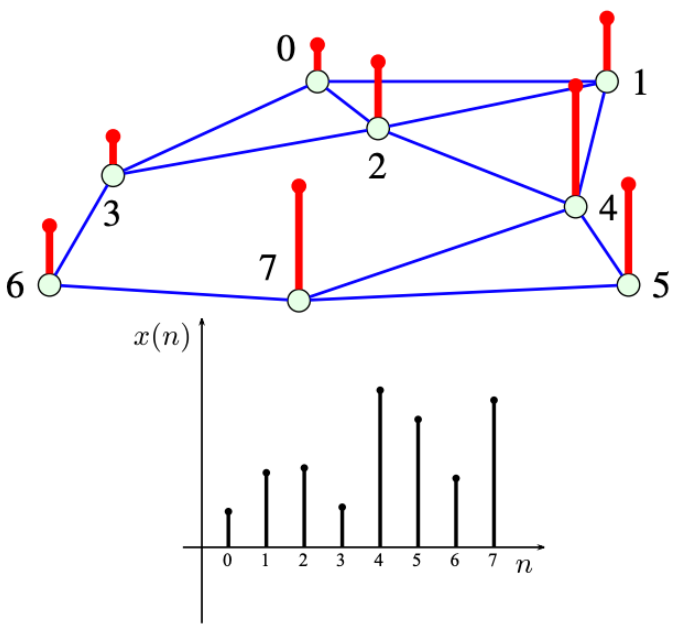
The eigenvectors corresponding to large λ_k correspond to the low-pass part of a graph signal

Signals and Systems on Graphs

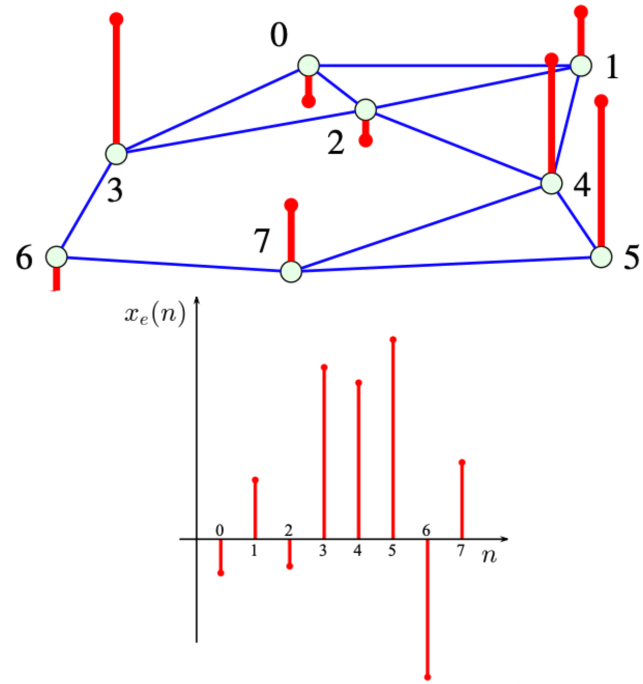
- Graph signal filtering
  - Ideal low-pass filter on a graph

cutoff eigen value

$$f(\lambda) = \begin{cases} 1, & \text{for } \lambda > \lambda_c, \\ 0, & \text{for other } \lambda. \end{cases}$$



(a) original signal, $\mathbf{x} = 3.2\mathbf{u}_7 + 2\mathbf{u}_6$



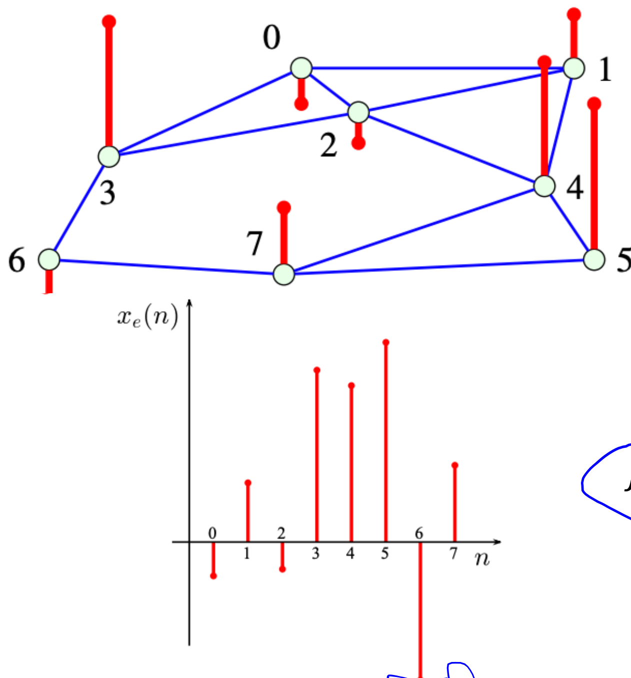
(b) noisy signal, $\mathbf{x}_\varepsilon = \mathbf{x} + \boldsymbol{\varepsilon}$

Signals and Systems on Graphs

$\text{SNR}_{\text{in}} = 2.7\text{dB}$

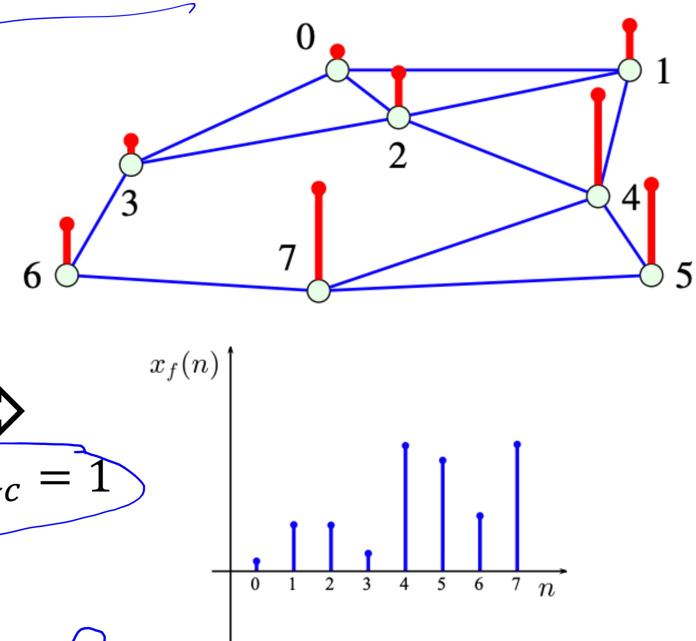
$$SNR = \frac{P_{signal}}{P_{noise}}$$

$\text{SNR}_{\text{in}} = 18.8\text{dB}$



(b) noisy signal, $\mathbf{x}_\varepsilon = \mathbf{x} + \boldsymbol{\varepsilon}$

$f(\lambda)$ with $\lambda_c = 1$



(c) filtered signal

$$\frac{\|s\|_2^2}{\delta^2}$$

$$\varepsilon \sim N(0, \sigma^2)$$

$$\frac{\|s\|_2^2}{N\sigma^2}$$

\# of vertices

Signals and Systems on Graphs

- Spectral domain filter design
  - Let $G(\Lambda)$ denote desired graph transfer function of a system defined on a graph. A system with this transfer function can be implemented either in the spectral domain or in the vertex domain
  - Spectral domain implementation steps:

    1. Calculate the GDFT of the input graph signal, $\mathbf{X} = \mathbf{U}^{-1}\mathbf{x}$,
    2. Multiply the GDFT of the input graph signal by the graph transfer function, $G(\mathbf{\Lambda})$, to obtain the output spectral form, $\mathbf{Y} = G(\mathbf{\Lambda})\mathbf{X}$, and
    3. Calculate the output graph signal as the inverse GDFT of $\mathbf{Y}$ in Step 2, that is, $\mathbf{y} = \mathbf{U}\mathbf{Y}$.

    - Computationally demanding for large graphs
    - More convenient to implement the desired filter (or its close approximation) directly in the vertex domain

assume λ_i are known

want to avoid

$y = \boxed{G(A)x}$

Signals and Systems on Graphs

- Spectral domain filter design
  - Let $G(\Lambda)$ denote desired graph transfer function of a system defined on a graph. A system with this transfer function can be implemented either in the spectral domain or in the vertex domain
  - Vertex domain implementation: find the coefficients (cf. standard impulse response) $h_0, h_1, \ldots, h_{M-1}$ in

$$\mathbf{y} = h_0 \mathbf{A}^0 \mathbf{x} + h_1 \mathbf{A}^1 \mathbf{x} + \cdots + h_{M-1} \mathbf{A}^{M-1} \mathbf{x} = \sum_{m=0}^{M-1} h_m \mathbf{A}^m \mathbf{x}$$

such that their spectral representation, $H(\Lambda)$, is equal to the $G(\Lambda)$

i.e.

$$H(\lambda_k) = \frac{Y(k)}{X(k)} = h_0 + h_1 \lambda_k + \cdots + h_{M-1} \lambda_k^{M-1} = G(\lambda_k)$$

for $k = 0, 1, \ldots, N-1$

Signals and Systems on Graphs

- Spectral domain filter design ~~Monomial~~ polynomial approximation
  - Vertex domain implementation (cont'd)

~~N equation~~

~~M unknowns~~

$$h_0 + h_1\lambda_0^1 + \cdots h_{M-1}\lambda_0^{M-1} = G(\lambda_0)$$

$$h_0 + h_1\lambda_1^1 + \cdots + h_{M-1}\lambda_1^{M-1} = G(\lambda_1)$$

$$\vdots$$

$$h_0 + h_1\lambda_{N-1}^1 + \cdots + h_{M-1}\lambda_{N-1}^{M-1} = G(\lambda_{N-1})$$

N equation

M unknowns.

$$\Downarrow$$

$$\mathbf{V}_\lambda\,\mathbf{h} = \mathbf{g}\ (1) \quad \text{with} \quad \mathbf{g} = [G(\lambda_0), G(\lambda_1), \ldots, G(\lambda_{N-1})]^T = \mathrm{diag}(G(\boldsymbol{\Lambda}))$$

$$\mathbf{h} = [h_0, h_1, \ldots, h_{M-1}]^T$$

$$\mathbf{V}_\lambda = \begin{bmatrix} 1 & \lambda_0^1 & \cdots & \lambda_0^{M-1} \\ 1 & \lambda_1^1 & \cdots & \lambda_1^{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_{N-1}^1 & \cdots & \lambda_{N-1}^{M-1} \end{bmatrix}$$

monomial base matrix

Signals and Systems on Graphs

- Spectral domain filter design
  - Vertex domain implementation (cont'd)

    Solution of (1):

    - Consider the case with N vertices and with all distinct eigenvalues of the adjacency
      - a) If the filter order, M, is such that M = N, then the solution to (1) is unique
      - b) If the filter order, M, is such that M < N, then the system in (1) is overdetermined. Therefore, the solution to (1) can only be obtained using least square approximation

$V_\lambda h = g$

$\prod (\lambda_i - \lambda_j)$

$0 \le i < j \le M$

N^2

= non-zero entries in A

$$y = P_{M-1}(A)x$$

$$P_{M-1}(A)x = c_0 + c_1 A + \dots + c_m A^m$$

A^m

2MNA

$$A^2 x = A(Ax)$$

$$A^m x = A(A^{m-1}x)$$

Signals and Systems on Graphs

of unique λ

$N > N_m$

- Spectral domain filter design
  - Vertex domain implementation (cont'd)

    Solution of (1): $\leftarrow V_\lambda h = g$

    - If some of the eigenvalues are of a degree higher than one, the system in (1) can be reduced into a system of N_m linear equations
      a) If the filter order, M, is such that $N_m < M \leq N$, the system in (1) is underdetermined, $(M - N_m)$ filter coefficients are free variables. The system has an infinite number of solutions
      b) If the filter order is such that $M = N_m$, the solution to the system in (1) is unique
      c) If the filter order is such that M < Nm, the system in (1) is overdetermined and the solution is obtained in the least squares sense

McGill

Signals and Systems on Graphs

- Spectral domain filter design
  - Vertex domain implementation (cont'd)
    - Exact Solution ($M = N$ or $M = N_m$)

$$\mathbf{h} = \mathbf{V}_\lambda^{-1}\mathbf{g}$$

    - Least-squares solution ($M < N_m$)

polynomial approximation with Vandermonde matrix

We want to minimize $e = \|\mathbf{V}_\lambda \mathbf{h} - \mathbf{g}\|_2^2$

From $\partial e/\partial \mathbf{h}^T = \mathbf{0}$ we then have

$$\hat{\mathbf{h}} = (\mathbf{V}_\lambda^T \mathbf{V}_\lambda)^{-1}\mathbf{V}_\lambda^T \mathbf{g} = \mathrm{pinv}(\mathbf{V}_\lambda)\mathbf{g}.$$

Since this solution may not satisfy $V_\lambda h = g$, the designed coefficient vector, $\hat{g}$, obtained from

$$\mathbf{V}_\lambda \hat{\mathbf{h}} = \hat{\mathbf{g}}$$

in general, differs from the desired system coefficients, g

Signals and Systems on Graphs

- Spectral domain filter design
  - Polynomial approximation of the system on a graph transfer function:
    - Why: Eigenanalysis is computationally expensive
    - How

$G(\lambda)$

      - » Without loss of generality, it can be considered that the desired transfer function, $g = [G(\lambda_0), G(\lambda_1), \ldots, G(\lambda_{N-1})]^T$, consists of samples taken from a continuous function of λ within the interval $\lambda_{min} \leq \lambda \leq \lambda_{max}$, where λ_{min} and λ_{max} denote the minimum and maximum value of $\{\lambda_0, \lambda_1, \ldots, \lambda_{N-1}\}$. The system on graph uses only the values at discrete points $\lambda \in \{\lambda_0, \lambda_1, \ldots, \lambda_{N-1}\}$
      - » We find to find an approximating polynomial $P(\lambda)$, which has the smallest maximum absolute error from the desired function value, i.e error at the points within $\lambda_{min} \leq \lambda \leq \lambda_{max}$, is bounded and sufficiently small => minmax approximation

Signals and Systems on Graphs

$y = G(A)x$

$A^n x$

$y(x) = (x)x \quad N_A \cdot n$

- Spectral domain filter design
  - Polynomial approximation of the system on a graph transfer function (cont'd)
    - Chebyshev approximation

$y = P_{M-1}(A)x$

$y = c_0 + c_1 A + c_2 A^2 + \cdots$

truncated chebyshev polynomials

(x)

$G(A) \rightarrow$

$$P_{M-1}(z) = \frac{c_0}{2} + \sum_{m=1}^{M-1} c_m T_m(z)$$

A^3

bases

Chebyshev polynomials

$$T_0(z) = 1,$$
$$T_1(z) = z,$$
$$T_2(z) = 2z^2 - 1,$$
$$T_3(z) = 4z^3 - 3z,$$
$$\vdots$$
$$T_m(z) = 2zT_{m-1}(z) - T_{m-2}(z),$$

Chebyshev coefficients

$$c_m = \frac{2}{\pi}\int_{-1}^{1} G(z)T_m(z)\frac{dz}{\sqrt{1-z^2}}$$
$$= \frac{2}{\pi}\int_0^{\pi} cos(m\theta)G(cos(\theta))d\theta.$$

Normalized Input $z \in [-1, 1]$

$$z = \frac{2\lambda - (\lambda_{\max} + \lambda_{\min})}{\lambda_{\max} - \lambda_{\min}}$$

$$\lambda = \frac{1}{2}\Big(z(\lambda_{\max} - \lambda_{\min}) + \lambda_{\max} + \lambda_{\min}\Big)$$

Signals and Systems on Graphs

- Spectral Domain of the Laplacian Matrix
  - Graph discrete Fourier transform (GDFT)

$$\mathbf{X} = \mathbf{U}^{-1}\mathbf{x}$$

where X denotes a vector of the GDFT coefficients, and U is a matrix whose columns represent the eigenvectors of the Laplacian matrix

    - For undirected graph: $U^{-1} = U^T$

$$\text{Vector} \rightarrow X(k) = \sum_{n=0}^{N-1} x(n) u_k(n)$$

  - Inverse graph discrete Fourier transform (IGDFT)

$$\mathbf{x} = \mathbf{U}\,\mathbf{X}$$

$$x(n) = \sum_{k=0}^{N-1} X(k) u_k(n)$$

Signals and Systems on Graphs

- System on a graph in the GDFT domain

$$\mathbf{y} = h_0\mathbf{L}^0\mathbf{x} + h_1\mathbf{L}^1\mathbf{x} + \cdots + h_{M-1}\mathbf{L}^{M-1}\mathbf{x}$$
$$= \sum_{m=0}^{M-1} h_m \mathbf{L}^m \mathbf{x}.$$

$\mathcal{L} = U\Lambda U^T$

$$\Rightarrow \quad \mathbf{y} = \mathbf{U}\mathbf{Y} = \sum_{m=0}^{M-1} h_m \mathbf{L}^m \mathbf{x} = H(\mathbf{L})\mathbf{x}$$
$$= \mathbf{U}H(\boldsymbol{\Lambda})\mathbf{U}^T\mathbf{x} = \mathbf{U}H(\boldsymbol{\Lambda})\mathbf{X}$$

with $H(\boldsymbol{\Lambda}) = \sum_{m=0}^{M-1} h_m \boldsymbol{\Lambda}^m$

$$\Rightarrow \quad \mathbf{Y} = H(\boldsymbol{\Lambda})\mathbf{X} \qquad \underline{Y(k)} = (h_0 + h_1\lambda_k + \cdots + h_{M-1}\lambda_k^{M-1})X(k)$$

Signals and Systems on Graphs

The n[th] element of the output signal, y = UH(Λ)U^T x, of a system on a graph is given by

$$y(n) = \sum_{k=0}^{N-1} \sum_{i=0}^{N-1} x(i) u_k(i) H(\lambda_k) u_k(n) = \sum_{i=0}^{N-1} x(i) h_n(i),$$

where $h_n(i) = \sum_{k=0}^{N-1} H(\lambda_k) u_k(n) u_k(i) = \mathcal{T}_n\{h(i)\}$. (graph impulse response)

Transfer function of a system on a graph:

$$H(\lambda_k) = h_0 + h_1 \lambda_k + \cdots + h_{M-1} \lambda_k^{M-1}$$

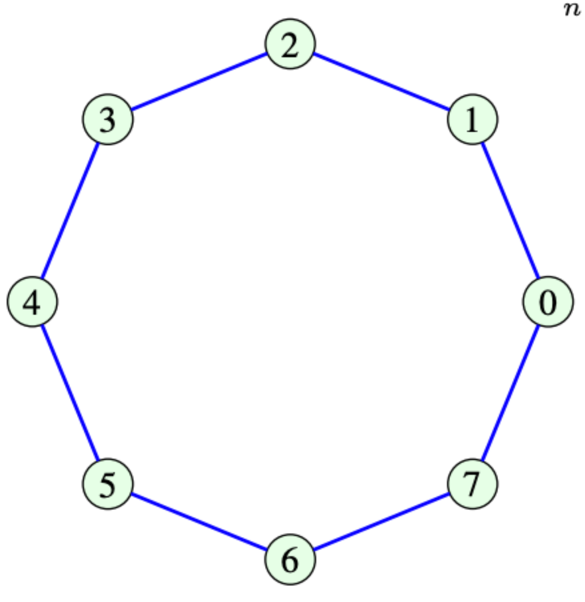
Signals and Systems on Graphs

- Graph signal filtering
  - Spectral Ordering: low-varying and high-varying eigenvectors
    - Second order finite difference on circular graph

$$y(n) = -u(n-1) + 2u(n) - u(n+1) = \lambda u = Lu$$

    - Energy of signal change on circular graph

$$E_u = \sum_n \left[\Big(u(n) - u(n-1)\Big)^2 + \Big(u(n) - u(n+1)\Big)^2 \right] / 2$$



$$\Updownarrow$$

$$E_u = u^T L u$$

$$\Updownarrow$$

$$\mathbf{u}^T \mathbf{L} \mathbf{u} = \lambda \mathbf{u}^T \mathbf{u} = \lambda = E_u$$

Signals and Systems on Graphs

- Graph signal filtering
  - Spectral Ordering (cont'd)

$$\mathbf{u}_k^T \mathbf{L} \mathbf{u}_k = \lambda_k = \frac{1}{2} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} W_{nm} \Big(u_k(n) - u_k(m) \Big)^2 \geq 0$$

$$\Downarrow$$

a small $u_k^T L u_k = \lambda_k$ corresponds to slow eigenvector variation $W_{nm}(u_k(n) - u_k(m))^2$ within the neighboring vertices

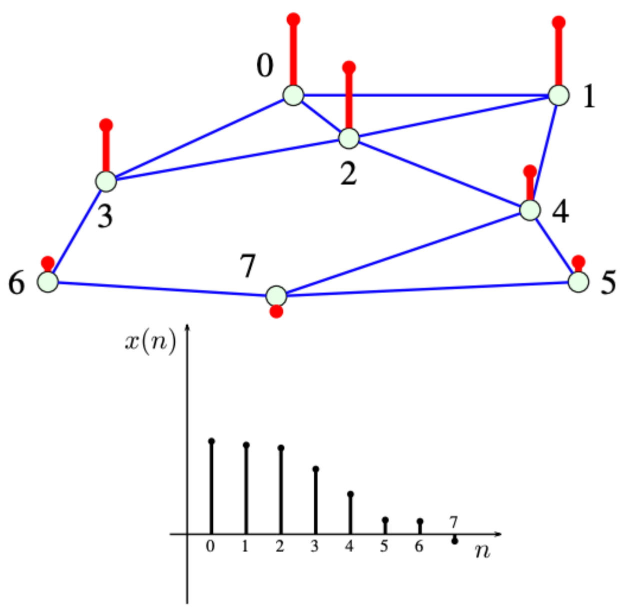
$$\Downarrow$$

the eigenvectors corresponding to small λ_k correspond to the low-pass part of a graph signal i.e. smoother.

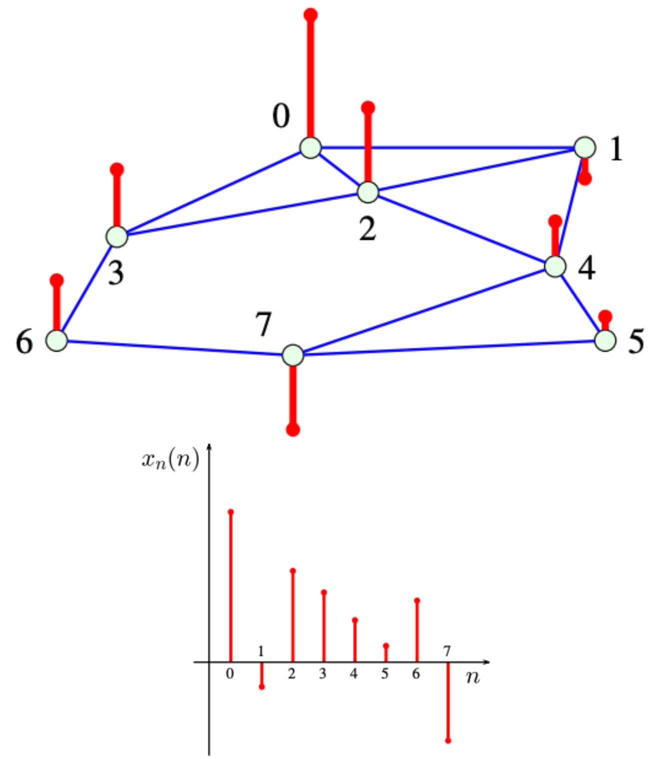
Signals and Systems on Graphs

- Graph signal filtering
  - Ideal low-pass filter on a graph

$$f(\lambda) = \begin{cases} 1, & \text{for } \lambda < \lambda_c \\ 0, & \end{cases}$$



(a) original signal, $x = 2u_0 + 1.5u_1$

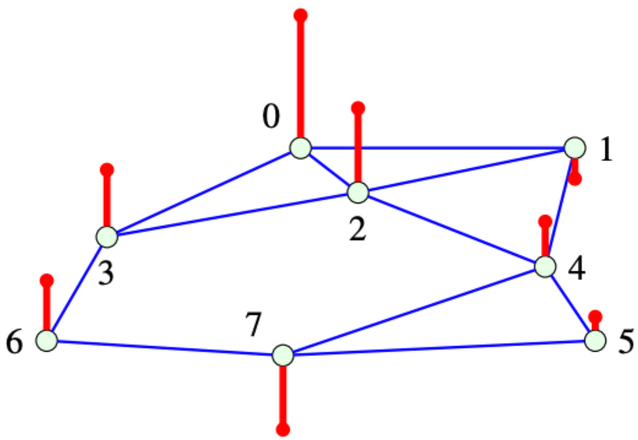


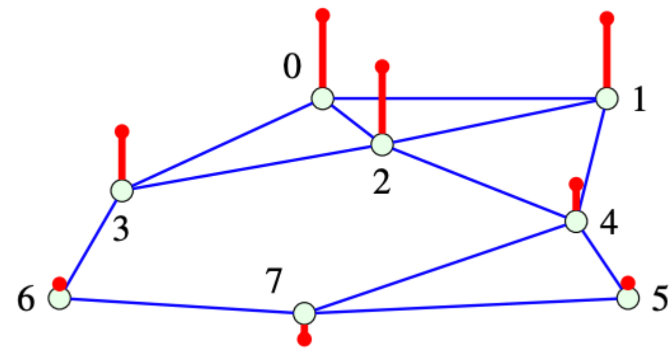
(b) noisy signal

Signals and Systems on Graphs

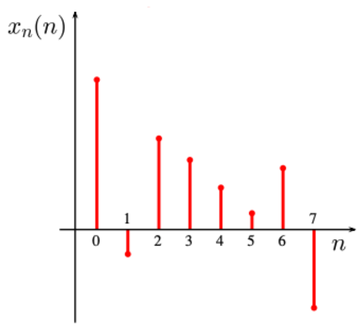
SNR$_{in}$ = -1.76 dB

SNR$_{in}$ = 21.29 dB

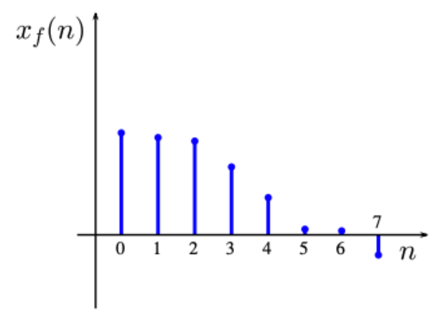




$f(\lambda)$ *with* $\lambda_c = 0.25$



(b) noisy signal



(c) filtered signal

Signals and Systems on Graphs

- Convolution of signals on a graph
  - Definition: Consider two graph signals, x(n) and h(n) A generalized convolution operator for these two signals on a graph is defined as:

$$\begin{aligned} y(n) &= x(n) * h(n) \\ &= \sum_{k=0}^{N-1} Y(k) u_k(n) = \sum_{k=0}^{N-1} X(k) H(k) u_k(n) \end{aligned}$$

$$\text{where} \quad H(k) = \sum_{n=0}^{N-1} h(n) u_k(n)$$

Based on the assumption that

$$Y(k) = X(k) H(k)$$

Signals and Systems on Graphs

- Shift on a graph
  - Consider the graph signal, h(n), and the delta function located at a vertex m,

$$\delta_m(n) = \begin{cases} 1, & \text{for } m = n \\ 0, & \text{for } m \neq n \end{cases}$$

GDFT: $$\Delta(k) = \sum_{n=0}^{N-1} \delta_m(n) u_k(n) = u_k(m)$$

  - The shifted signal is obtained as a convolution of the original signal and an appropriately shifted delta function

$$h(n) * \delta_m(n) \text{ with GDFT } H(k)u_k(m)$$

$h_m(n)$: the shifted version of the graph signal, h(n), "toward" a vertex m

$$h_m(n) = h(n) * \delta_m(n) = \sum_{k=0}^{N-1} H(k) u_k(m) u_k(n)$$

Signals and Systems on Graphs

- Shift on a graph
  - Output signal

$$
\begin{aligned}
y(n) &= \sum_{k=0}^{N-1} X(k)H(k)u_k(n) \\
&= \sum_{k=0}^{N-1} \sum_{m=0}^{N-1} x(m)u_k(m)H(k)u_k(n) \\
&= \sum_{m=0}^{N-1} x(m)h_m(n) = x(n) * h(n),
\end{aligned}
$$

Where, $$h_m(n) = \sum_{k=0}^{N-1} H(k)u_k(m)u_k(n) = T_m\{h(n)\}$$

Signals and Systems on Graphs

- Optimal Denoising
  - Problem: Consider a measurement, x, composed of a slow-varying graph signal, s, and a fast changing disturbance, ε: $x = s + \varepsilon$. The aim is to design a filter for disturbance suppression (denoising), the output is denoted by y = H(x)
  - Solution:

$$J = \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \alpha \mathbf{y}^T \mathbf{L} \mathbf{y}$$

$$\frac{\partial J}{\partial \mathbf{y}^T} = \mathbf{y} - \mathbf{x} + 2\alpha \mathbf{L} \mathbf{y} = \mathbf{0}$$

$$\Longrightarrow \quad \mathbf{y} = (\mathbf{I} + 2\alpha \mathbf{L})^{-1} \mathbf{x}.$$

$$\Longrightarrow \quad \mathbf{Y} = (\mathbf{I} + 2\alpha \mathbf{\Lambda})^{-1} \mathbf{X}.$$ (The Laplacian spectral domain form)

$$H(\lambda_k) = \frac{1}{1 + 2\alpha \lambda_k}$$ (graph filter transfer function)

Next Time:

- Subsampling, Compressed Sensing, and Re-construction
- Random Graph Singals