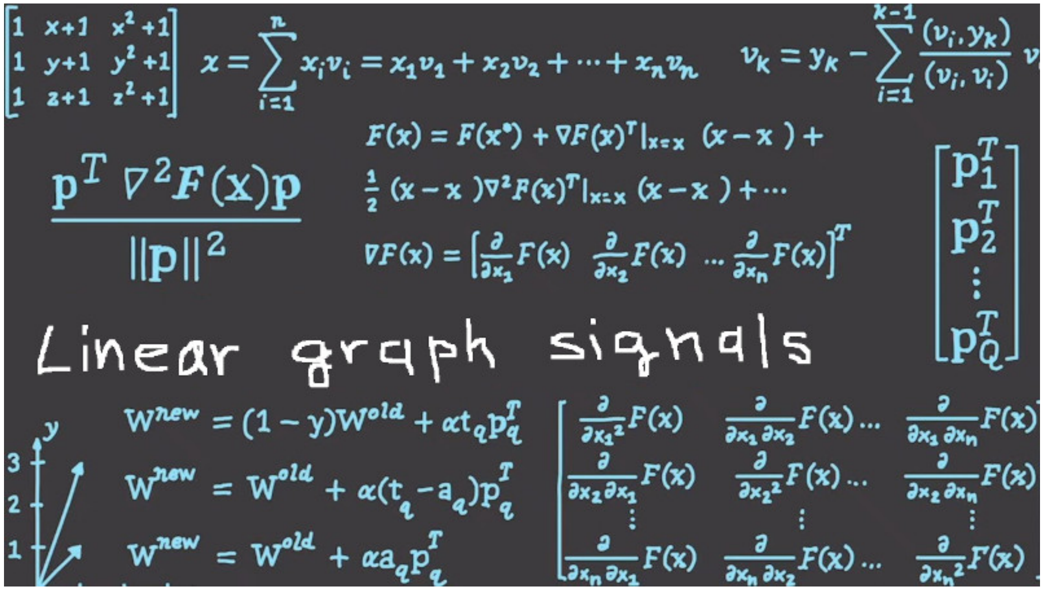
# Processing Linear Graph Signals




**Pavel Loskot**
*pavelloskot@intl.zju.edu.cn*

ZJU-UIUC INSTITUTE
Zhejiang University-University of Illinois at Urbana-Champaign Institute
浙江大学伊利诺伊大学厄巴纳香槟校区联合学院

The Seventh International Conference on Advances in Signal, Image and Video Processing
SIGNAL 2022
May 22, 2022 to May 26, 2022 - Venice, Italy
IARIA

# About Me

Pavel Loskot joined the ZJU-UIUC Institute as Associate Professor in January 2021. He received PhD in Wireless Communications from University of Alberta, Canada, and MSc and BSc in Radioelectronics and Biomedical Electronics, respectively, from the Czech Technical University of Prague. He is Senior Member of the IEEE, Fellow of the HEA, and the Recognized Research Supervisor of the UKCGE in the UK.

In the past 25 years, he was involved in numerous industrial and academic collaborative projects in the Czech Republic, Finland, Canada, UK, Turkey, and China. These projects concerned mainly wireless and optical telecommunication networks, but also genetic circuits, air transport services, and renewable energy systems. This experience allowed him to truly understand the interdisciplinary workings, and crossing the disciplines boundaries.

His current research focuses on statistical signal processing, classical machine learning, and importing methods from Telecommunication Engineering and Computer Science to model and analyze systems more efficiently and with greater information power.

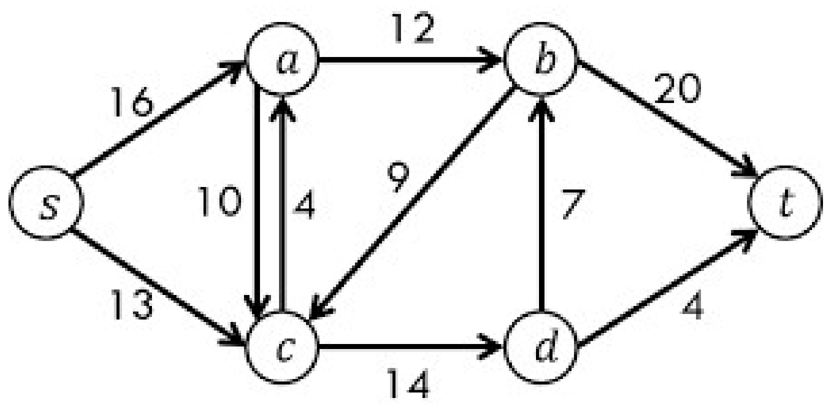
## Objective

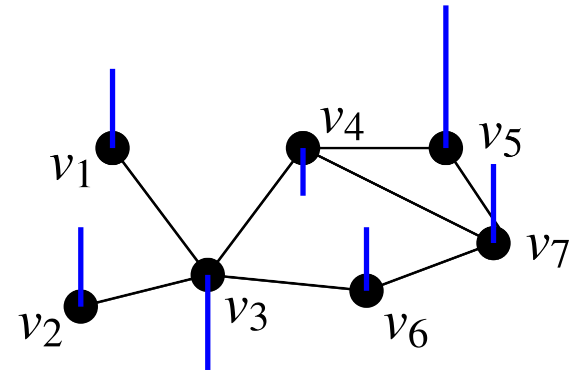
Define linear graph signals, and outline typical signal processing problems where these signals appear.

## Topics

- graph signals and systems
- linear graph signals
- difference equation
- digital filters
- ARMA signal model
- linear structural causal model
- graph inference
- compressive sensing
- graph signal compression/decompression
- linear statistical learning
- gamma process generation

# Graph Signals and Systems




## Graph systems

- flow-based modeling: networks of nodes with links
- flows have direction, constrained by transport capacity
- multiple-inputs (sources), multi-outputs (sinks)
- scenarios: epidemics, information spreading, Internet, computing etc.

## Graph signals*

- relationships among random variables: graphs of vertexes with edges
- set of instantaneous and long-term features/attributes/interactions
- features/attributes/interactions can be zero or not available
- scenarios: multivariate time-series, computer vision, biological signals

---

*P. Loskot, "A Generative Model for Correlated Graph Signals," Mathematics, 9(23), Nov. 2021.

# Graph Signal Processing

## Mainstream approach*

- 1D signal filtering

$$s_{t-1} = z^{-1} s_t, \qquad h(z) = h_0 z^0 + h_1 z^{-1} + \ldots + h_{N-1} z^{-(N-1)}$$

- graph filtering

$$\boldsymbol{s}_{t-1} = \boldsymbol{A}^{-1} \boldsymbol{s}_t, \qquad h(\boldsymbol{A}) = h_0 \boldsymbol{A}^0 + h_1 \boldsymbol{A}^{-1} + \ldots + h_{N-1} \boldsymbol{A}^{-(N-1)}$$

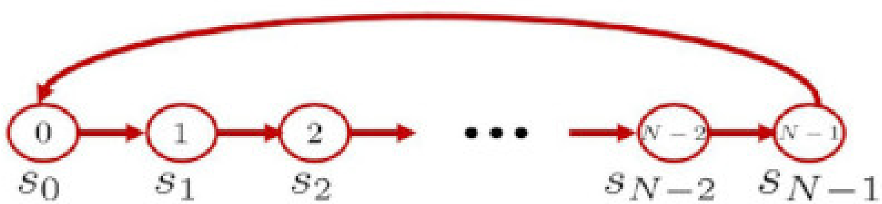
- shift-invariance

$$h'(z) \to z\, h(z), \qquad h'(\boldsymbol{A}) \to \boldsymbol{A}\, h(\boldsymbol{A})$$

- graph Fourier transform

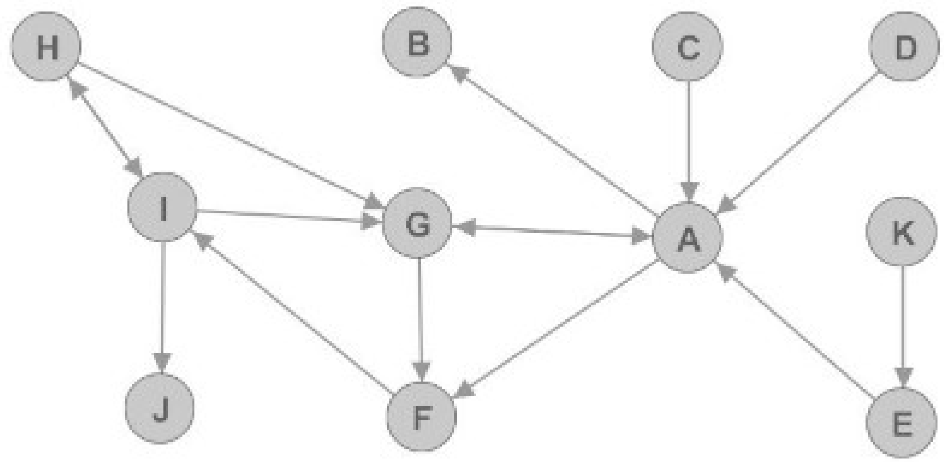
$$\boldsymbol{s}_{\text{out}} = \boldsymbol{A}\, \boldsymbol{s}_{\text{in}}, \quad \boldsymbol{A} = \boldsymbol{U}^{-1} \boldsymbol{\Lambda} \underbrace{\boldsymbol{U}}_{\text{GFT}}$$

$$\boldsymbol{U} \boldsymbol{s}_{\text{out}} = h(\boldsymbol{\Lambda})\, \boldsymbol{U}\, \boldsymbol{s}_{\text{in}}$$




*P. Loskot, "Current Approaches to Graph Signal Processing," Tutorial, Signal 2021.

# Linear Graph Signals




## Definition

- linearly combine neighbor node values

$$s_t(i) = \sum_{j \in \mathcal{N}(i)} \boldsymbol{A}_{i,j}\, s_{t-1}(i) \quad \Rightarrow \quad \boldsymbol{s}_t = \boldsymbol{s}_{t-1} + \boldsymbol{A}'\boldsymbol{s}_{t-1} = (\boldsymbol{I} + \boldsymbol{A}')\,\boldsymbol{s}_{t-1} = \boldsymbol{A}\,\boldsymbol{s}_{t-1}$$

- some nodes act as sinks or sources

$$\boldsymbol{s}_t = \boldsymbol{A}\,\boldsymbol{s}_{t-1} + \boldsymbol{u}_t$$

- linearly combine node values from neighbors at most $K$-hops away

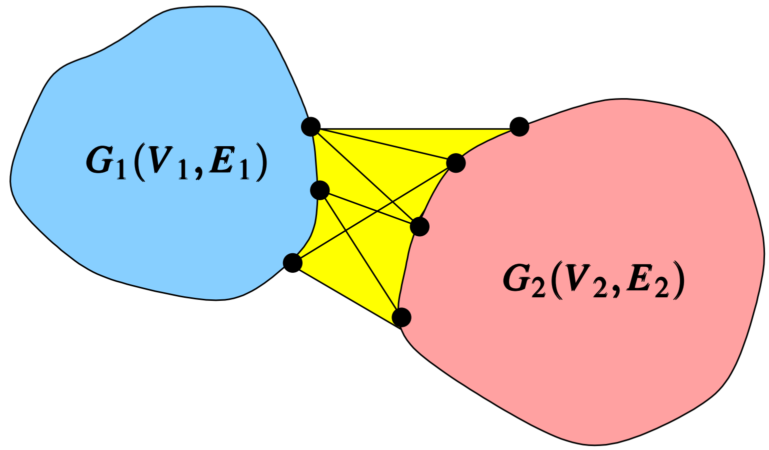
$$\boldsymbol{s}_t = \sum_{k=1}^{K} \boldsymbol{A}^k \boldsymbol{s}_{t-k} + \boldsymbol{u}_t \qquad (1)$$

  note that $\sum_{k=1}^{K} \boldsymbol{A}^k$ is the number of walks of length at most $K$

- eq. (1) is a stationary MIMO system with memory

# Linear Graph Signals (2)

## Super-linear graph signals




- graphs can be merged or split

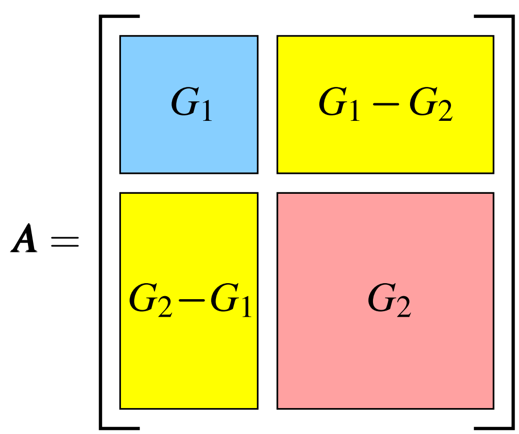
$$G = G_1 \circ G_2$$

- graphs can be embedded and form hierarchical structures

$$V_{\text{subset}} \rightarrow G_{\text{subset}} \quad \text{or} \quad G_{\text{subset}} \rightarrow V_{\text{subset}}$$

## Adjacency matrix

- $\boldsymbol{A}$ is symmetric ($\boldsymbol{A} = \boldsymbol{A}^T$) for undirected graphs
- $\boldsymbol{A}$ is asymmetric for directed graphs



# Linear Graph Signals (3)

## Graph inference

- given observations $\boldsymbol{s}_t$, $t = 1, \ldots, T$, find $\boldsymbol{A}$
  → statistical learning
- let $\boldsymbol{S}_{T-1} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{T-1}]$ and $\boldsymbol{S}_T = [\boldsymbol{s}_2, \ldots, \boldsymbol{s}_T]$, then

$$\boldsymbol{S}_T = \boldsymbol{A}\boldsymbol{S}_{T-1} \quad \Rightarrow \quad \hat{\boldsymbol{A}} = \boldsymbol{S}_T \boldsymbol{S}_{T-1}^{+}$$

- issues:
  → what if $T \gg N$ (number of nodes)?
  → other norms e.g. $l_1$ or $l_\infty$ or $\boldsymbol{A}$ to be binary?
  → add regularization or prior knowledge?
  → noisy measurements with sub-sampling (missing data)?

## Graph design

- design $\boldsymbol{A}$ without measurements
  → model based design
  → statistical signal processing
- constraints:
  → model structure (e.g. sparsity)
  → input/output statistics (e.g. distributions, moments, independence)

*Many problems can be formulated this way.*

# Vector Difference Equation

$$s_t = \sum_{d=1}^{D} A_d\, s_{t-d} + b = A_t \otimes s_t + b$$

$D$ : order

$A_d, b$ : deterministic matrices, vector

First-order case $(D = 1)$

- homogeneous: $b = 0$

$$s_t = A^t s_0 = \left(U \Lambda U^T\right)^t s_0 = U \Lambda^t U^T s_0$$

- non-homogeneous: $b \neq 0$

$$s_t = s_{t-1} = (I - A)^{-1} b \equiv s^* \quad \text{(steady-state)}$$

$$s_t = s^* + A(s_{t-1} - s^*)$$

stable, if $\lim_{t\to\infty} s_t = s^*$

General case $(D > 1)$

- using roots of characteristic matrix polynomial of a homogeneous system
- using element-wise (vector) Z-transform

$$\tilde{s}(z) = \mathscr{Z}\{s_t\} = \sum_{t=0}^{\infty} s_t z^{-t}, \qquad s_t = \mathscr{Z}^{-1}\{\tilde{s}(z)\} = \frac{1}{2\pi \mathrm{j}} \oint_C \tilde{s}(z) z^{t-1}\, \mathrm{d}z$$

# Matrix Difference Equation

## First-order case

$$\boldsymbol{S}_{t+1} = \boldsymbol{A}\boldsymbol{S}_t + \boldsymbol{B}, \quad t = 0, 1, 2, \ldots$$

- deterministic system, initial value $\boldsymbol{S}_0$ and $\boldsymbol{A}$ and $\boldsymbol{B}$ are known
- steady state

$$\bar{\boldsymbol{S}} = \boldsymbol{A}\bar{\boldsymbol{S}} + \boldsymbol{B} \quad \Rightarrow \quad \bar{\boldsymbol{S}} = \frac{\boldsymbol{B}}{\boldsymbol{I}-\boldsymbol{A}}$$

- if $\boldsymbol{A} \neq \boldsymbol{I}$, then

$$\boldsymbol{S}_t = \boldsymbol{A}^t\boldsymbol{S}_0 + \left(\frac{\boldsymbol{I}-\boldsymbol{A}^t}{\boldsymbol{I}-\boldsymbol{A}}\right)\boldsymbol{B} = \boldsymbol{A}^t\Big(\boldsymbol{S}_0 - \underbrace{\frac{\boldsymbol{B}}{\boldsymbol{I}-\boldsymbol{A}}}_{\bar{\boldsymbol{S}}}\Big) + \frac{\boldsymbol{B}}{\boldsymbol{I}-\boldsymbol{A}}$$

- alternatively,

$$\boldsymbol{S}_t = c\boldsymbol{A}^t + \underbrace{\frac{\boldsymbol{B}}{\boldsymbol{I}-\boldsymbol{A}}}_{\bar{\boldsymbol{S}}}, \quad c = \boldsymbol{S}_0 - \frac{\boldsymbol{B}}{\boldsymbol{I}-\boldsymbol{A}} \quad \text{(same solution as above)}$$

- if $\boldsymbol{A} = \boldsymbol{I}$, then

$$\boldsymbol{S}_t = \boldsymbol{S}_0 + \boldsymbol{B}t$$

# Digital Filters

## Infinite size, LTI SISO system

- convolution, $\boldsymbol{y} = \boldsymbol{h} \otimes \boldsymbol{x}$, $\boldsymbol{h} = [h_0, h_1, \ldots, h_K]$ is finite impulse response
- matrix form with $N$ input time-series, $\boldsymbol{H}$ is Toeplitz
  $\rightarrow$ infinite $\boldsymbol{H}$ also allows for IIR filters

$$\underbrace{\begin{bmatrix} y_{01} & \cdots & y_{0N} \\ y_{11} & \cdots & y_{1N} \\ \vdots & & \vdots \end{bmatrix}}_{\boldsymbol{Y}} = \underbrace{\begin{bmatrix} h_0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ h_1 & h_0 & 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \ddots & & \ddots & & \cdots \\ 0 & 0 & h_K & \cdots & h_1 & h_0 & \cdots \\ \vdots & \vdots & \vdots & & \ddots & \ddots & \ddots \end{bmatrix}}_{\boldsymbol{H}} \cdot \underbrace{\begin{bmatrix} x_{01} & \cdots & x_{0N} \\ x_{11} & \cdots & x_{1N} \\ \vdots & & \vdots \end{bmatrix}}_{\boldsymbol{X}}$$

- estimating $\boldsymbol{h}$ (system identification)

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{h} \otimes \boldsymbol{x} = \boldsymbol{H} \cdot \boldsymbol{x} \\ \boldsymbol{y} &= \boldsymbol{x} \otimes \boldsymbol{h} = \boldsymbol{X} \cdot \boldsymbol{h} \end{aligned} \quad \Rightarrow \quad \hat{\boldsymbol{h}} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{h}$$

## Finite size, LTI SISO system

- cyclic convolution, $\boldsymbol{y} = \mathrm{DFT}^{-1}\{\mathrm{DFT}\{\boldsymbol{h}\} \cdot \mathrm{DFT}\{\boldsymbol{x}\}\}$
- $\boldsymbol{H}$ is cyclic and invertible (in general, inverse of FIR filter is IIR filter)

$$\boldsymbol{Y} = \begin{bmatrix} h_0 & 0 & \cdots & 0 & h_K & \cdots & h_2 & h_1 \\ h_1 & h_0 & 0 & \cdots & 0 & h_K & \cdots & h_2 \\ \vdots & \vdots & \ddots & & & \ddots & \ddots & \end{bmatrix} \cdot \boldsymbol{X}$$

# Digital Filters (2)

## MIMO IIR system

- state-space enables compact representation
- $\boldsymbol{p}$ is a (vector) state

$$\begin{aligned} \boldsymbol{p}(t+1) &= \boldsymbol{A}\boldsymbol{p}(t)+\boldsymbol{B}\boldsymbol{x}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{p}(t)+\boldsymbol{D}\boldsymbol{x}(t) \end{aligned}$$

- $\boldsymbol{A}$ is state transition matrix
  $\rightarrow$ determines dynamics (e.g., resonant modes)
- in control engineering, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are time-varying

## Impulse (matrix) response

- $\boldsymbol{h}(t)=\boldsymbol{y}(t)$, for $\boldsymbol{x}(t)=\delta(t)$

$$\boldsymbol{h}(t)=\begin{cases} \boldsymbol{D} & t=0 \\ \boldsymbol{C}\boldsymbol{A}^{t-1}\boldsymbol{B} & t>0 \end{cases}$$

## Overall output response

$$\boldsymbol{y}(t) = \underbrace{\boldsymbol{C}\boldsymbol{A}^{t-1}\boldsymbol{p}(0)}_{\text{initial state } \boldsymbol{p}(0)} + \boldsymbol{h}(t)\otimes\boldsymbol{x}(t)=\boldsymbol{C}\boldsymbol{A}^{t-1}\boldsymbol{p}(0) + \begin{cases} \boldsymbol{D}\boldsymbol{x}(0) & t=0 \\ \sum_{\tau=0}^{t}\boldsymbol{C}\boldsymbol{A}^{\tau-1}\boldsymbol{B}\,\boldsymbol{x}(t-\tau) & t>0 \end{cases}$$

# Digital Filters (3)

## Transfer function

- (matrix) Z-transform

$$\tilde{\boldsymbol{h}}(z) = \sum_{t=0}^{\infty} \boldsymbol{h}(t)\, z^{-t} = \boldsymbol{D} + \sum_{t=1}^{\infty} (\boldsymbol{C}\boldsymbol{A}^{n-1}\boldsymbol{B}) z^{-n} = \boldsymbol{D} + \boldsymbol{C}\, \underbrace{z^{-1} \sum_{t=0}^{\infty} (z^{-1}\boldsymbol{A})^t}_{(z\boldsymbol{I}-\boldsymbol{A})^{-1}}\, \boldsymbol{B}$$

- MIMO system in Z-domain

$$\tilde{\boldsymbol{y}}(z) = \tilde{\boldsymbol{h}}(z) \odot \tilde{\boldsymbol{x}}(z)$$

- all components of $\tilde{\boldsymbol{h}}(z)$ have the same poles, but different zeros
  $\rightarrow$ different controllability from particular inputs
  $\rightarrow$ different observability at particular outputs

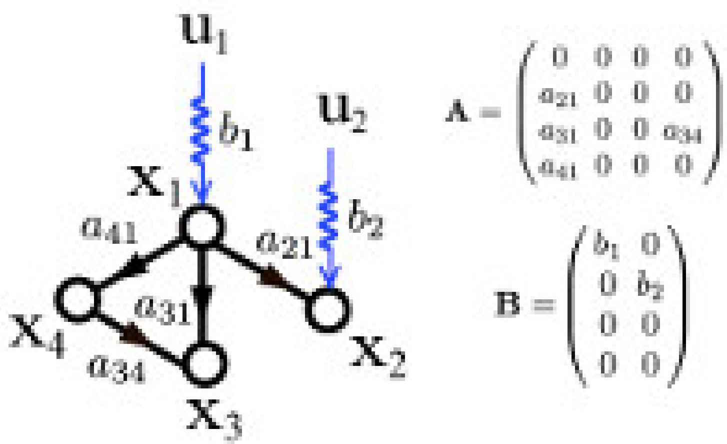
## Similarity transform

- $\boldsymbol{x}'(t) = \boldsymbol{S}\, \boldsymbol{x}(t)$, does not change $\tilde{\boldsymbol{h}}(z)$, and so also does not change $\boldsymbol{h}(t)$

## Diagonal $\boldsymbol{A}$

- if transition matrix, $\boldsymbol{A} \equiv \boldsymbol{\Lambda}$, and $\boldsymbol{S}$ are eigenvectors of $\boldsymbol{A}$, then
  $\rightarrow$ the system is a set of decoupled parallel one-pole SISO systems
  $\rightarrow$ it allows partial fraction expansion of $\boldsymbol{H}(z)$, e.g. for stability analysis

# Digital Filters (4)




$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$$

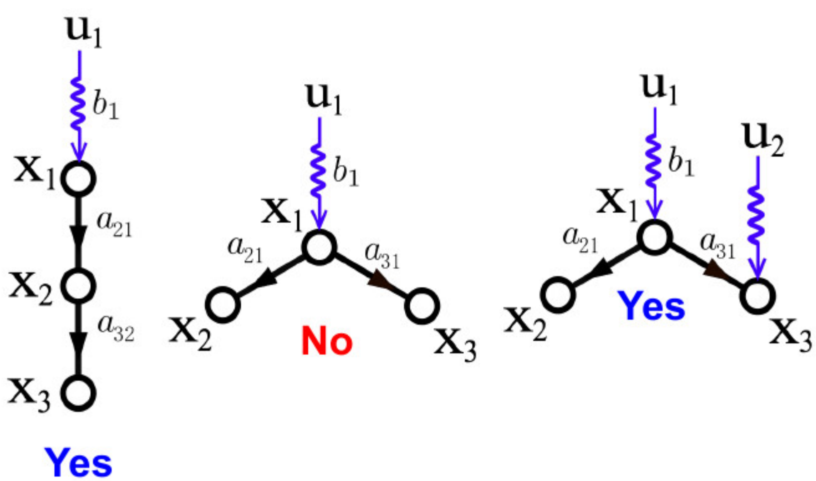
## Observability

- state $\boldsymbol{x}(t)$ can be inferred from finite observations (subset of nodes)
  $\rightarrow$ can find trajectory $\boldsymbol{x}(t)$ from any initial state $\boldsymbol{x}(0)$ to current state
  $\rightarrow$ $[\boldsymbol{C}, \boldsymbol{C}\boldsymbol{A}, \ldots, \boldsymbol{C}\boldsymbol{A}^{n-1}]^T$ must have rank $n$
- strongly connected components (there is a path between any two nodes)
  $\rightarrow$ have to observe at least one node from each SCC
- for linear model, can use maximum matching to find minimum # sensors
  $\rightarrow$ often much larger than # SCC (due to model symmetries)
- surprisingly, for non-linear model, sensors predicted by SCCs are necessary as well as sufficient (since model symmetries are rare)

---

*YY Liu, JJ Slotine, AL Barabási, Observability of complex systems, PNAS 2013.*

# DIGITAL FILTERS (5)




$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$$

## Controllability

- can drive the system between any given states
  $\rightarrow$ $[\boldsymbol{B}, \boldsymbol{A}\boldsymbol{B}, \boldsymbol{A}^2\boldsymbol{B}, \ldots, \boldsymbol{A}^{n-1}\boldsymbol{B}]$ must be full rank (Kalman condition)
- surprisingly, driver nodes tend to avoid hubs
  $\rightarrow$ average degree of driver nodes is smaller than average degree of a graph
  $\rightarrow$ required # of driver nodes mainly determined by graph degree distribution
- sparse and heterogeneous networks are harder to control than dense and homogeneous networks

---

*YY Liu, JJ Slotine, AL Barabási, Controllability of complex networks, Nature 2011.*

# ARMA Model

## LTI SISO system

- described by a difference equation of order $(m, n)$

$$s(t) = \sum_{k=1}^{m} a_k\, s(t-k) + \sum_{k=0}^{M} b_k\, u(t-k)$$

- can introduce a vector of $m$ states to obtain the first-order difference equation

$$\underbrace{\begin{bmatrix} p_1(t) \\ p_2(t) \\ \vdots \\ p_{m-1}(t) \\ p_m(t) \end{bmatrix}}_{\boldsymbol{p}(t)} = \underbrace{\begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -a_{m-1} & 0 & 0 & \ddots & 1 \\ -a_m & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\boldsymbol{A}} \times \underbrace{\begin{bmatrix} p_1(t-1) \\ p_2(t-1) \\ \vdots \\ p_{m-1}(t-1) \\ p_m(t-1) \end{bmatrix}}_{\boldsymbol{p}(t-1)} + \underbrace{\begin{bmatrix} b_0 \\ \vdots \\ b_M \\ \vdots \\ 0 \end{bmatrix} u(t)}_{\boldsymbol{q}(t)}$$

- process noise

$\rightarrow$ $\mathrm{E}[\boldsymbol{q}(t)] = \boldsymbol{0}$ (zero-mean), $\mathrm{E}\left[\boldsymbol{q}^T(t)\boldsymbol{q}(\tau)\right] = \begin{cases} \boldsymbol{C}_{\boldsymbol{q}}(t) & t = \tau \\ \boldsymbol{0} & t \neq \tau \end{cases}$ (white, but non-stationary)

# ARMA Model (2)

## MIMO LTI system

- described by a difference equation of order $(m,n)$

$$\boldsymbol{s}(t) = \sum_{k=1}^{m} \boldsymbol{A}_k \boldsymbol{s}(t-k) + \sum_{k=0}^{M} \boldsymbol{B}_k \boldsymbol{u}(t-k) \qquad (1)$$

- if $\boldsymbol{A}$ and $\boldsymbol{B}$ are $N \times N$ matrices, (1) can be rewritten as SISO ARMA model of order $(mN, nN)$
  $\rightarrow$ then again rewritten as the first-order difference model

$$\boldsymbol{p}(t) = \boldsymbol{A}\boldsymbol{p}(t-1) + \boldsymbol{q}(t)$$

## Estimating the model graph

- stacking up $T$ measurements followed by the least-square fitting

$$\boldsymbol{P}_T = \boldsymbol{A}\boldsymbol{P}_{T-1} + \boldsymbol{Q} \quad \Rightarrow \quad \hat{\boldsymbol{A}} = \boldsymbol{P}_T \boldsymbol{P}_{T-1}^T (\boldsymbol{P}_{T-1} \boldsymbol{P}_{T-1}^T)^{-1}$$

- but numerical issues as the dimension $\propto N$

# Linear Structural Causal Models

## Linear SCM

$$
\begin{aligned}
Y_i &= \textstyle\sum_{k \in \mathrm{PA}_j} \beta_{jk} X_k + U_i \\
\boldsymbol{Y} &= \boldsymbol{\beta} \boldsymbol{X} + \boldsymbol{U}
\end{aligned}
$$

$\beta_{jk}$ : structural coefficients

$X_k$ : direct cause of $Y_i$

$U_i$ : exogenous unobserved variables/effects (effects outside the model)

## SCM rules

- $X_i$ are normalized ($\mathrm{E}[X_i] = 0$, $\mathrm{E}\left[X_i^2\right] = 1$)

$$X \xrightarrow{\ \beta\ } Y$$

$$\mathrm{E}[Y|\mathrm{do}(X = x)] = \mathrm{E}[\beta x + U] = \beta x$$

$$\mathrm{E}[XY] = \mathrm{E}[X(\beta X + U)] = \beta \mathrm{E}\left[X^2\right] + \mathrm{E}[X]\,\mathrm{E}[U] = \beta$$

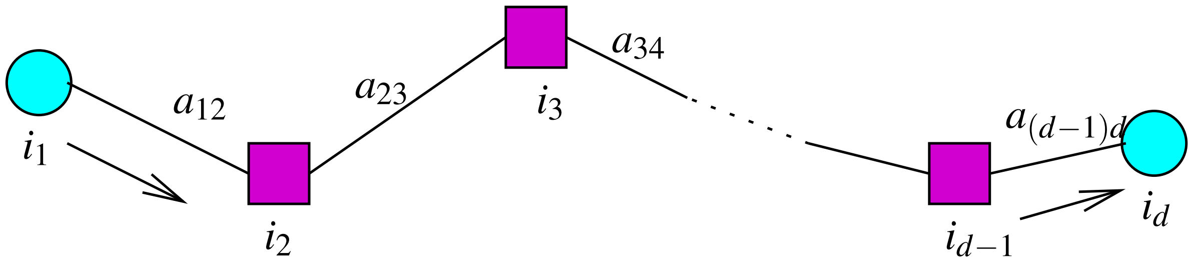
$$\overset{u_{XY}}{\longleftrightarrow} \qquad X \xrightarrow{\ \beta\ } Y$$

$$\mathrm{E}[XY] = \mathrm{E}[X(\beta X + U_Y)] = \beta + \mathrm{E}[X U_Y] = \beta + u_{XY}$$

Theorem: Linear SCM with Gaussian exogenous effects is fully identifiable from the observations, if the exogenous variables have equal or known variances.

- run independence tests to identify (multiple) candidate SCMs
  $\rightarrow$ select the model with the best score/likelihood

# Network Tomography




## Network observations

- network with weights $a_{ij}$ between nodes $i$ and $j$
- path $\mathcal{P} = \{(i_1 i_2), (i_2 i_3), \ldots, (i_{d-1} i_d)\}$ from node $i_1$ to node $i_d$
  → accumulated weight along the path $\mathcal{P}$ is $y = \sum_{(ij)\in\mathcal{P}} a_{ij}$
- paths as rows in binary matrix $\boldsymbol{P} \in \{0,1\}^{m\times n}$ measured by probes $\boldsymbol{x}$, and $\boldsymbol{a}$ represents an adjacency weight vector

$$\boldsymbol{y} = \boldsymbol{P} \cdot \boldsymbol{a} + \boldsymbol{x}$$

## Task

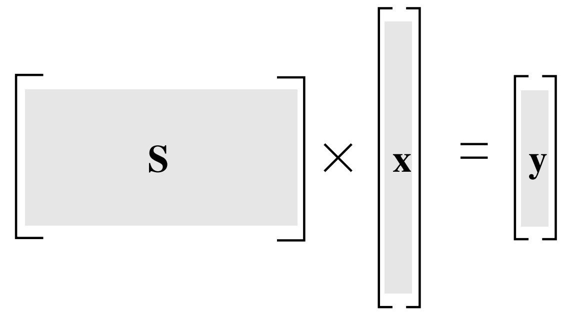
- find minimal $\boldsymbol{P}$ to efficiently estimate $\boldsymbol{a}$ from $\boldsymbol{y}$ and $\boldsymbol{x}$
  → only some nodes may be suitable as inputs $\boldsymbol{x}$ and outputs $\boldsymbol{y}$
- requires over-determined system of equations
  → if $\boldsymbol{a}$ is sparse, may design $\boldsymbol{P}$ to achieve compressive sensing
- alternatively, determine (sparse) change $\Delta\boldsymbol{a}$ (anomaly), if nominal $\boldsymbol{a}$ known

$$\boldsymbol{y} = \boldsymbol{P} \cdot (\boldsymbol{a} + \Delta\boldsymbol{a}) + \boldsymbol{x} = \underbrace{\boldsymbol{P} \cdot \boldsymbol{a}}_{\text{known}} + \boldsymbol{P} \cdot \Delta\boldsymbol{a} + \boldsymbol{x}$$

# Compressive Sensing

## Linear sensing

$$\boldsymbol{y} = \boldsymbol{S} \cdot \boldsymbol{x} + \boldsymbol{w}$$




- $\boldsymbol{y}$ : compressed/observed signal
- $\boldsymbol{S}$ : sensing matrix
- $\boldsymbol{x}$ : desired signal
- $\boldsymbol{w}$ : observation noise

## Task

- given $\boldsymbol{y}$, $\boldsymbol{S}$, and norms $l_p$ and $l_q$, approximate $\boldsymbol{x}$ by $\boldsymbol{x}^*$, so that

$$||\boldsymbol{x}^* - \boldsymbol{x}||_p \leq C(k) \min_{\tilde{\boldsymbol{x}}} ||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_q$$

- best $\boldsymbol{x}^*$ contains $k$ largest (abs) values of $\boldsymbol{x}$, so if $\boldsymbol{x}$ is also $k$-sparse, then $\boldsymbol{x}^* = \boldsymbol{x}$, and the recovery is exact
- typically

$$\begin{aligned} ||\boldsymbol{x}^* - \boldsymbol{x}||_2 &\leq C \min_{\tilde{\boldsymbol{x}}} ||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_2 \\ ||\boldsymbol{x}^* - \boldsymbol{x}||_1 &\leq C \min_{\tilde{\boldsymbol{x}}} ||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_1 \\ ||\boldsymbol{x}^* - \boldsymbol{x}||_2 &\leq (C/\sqrt{k}) \min_{\tilde{\boldsymbol{x}}} ||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_1 \end{aligned}$$

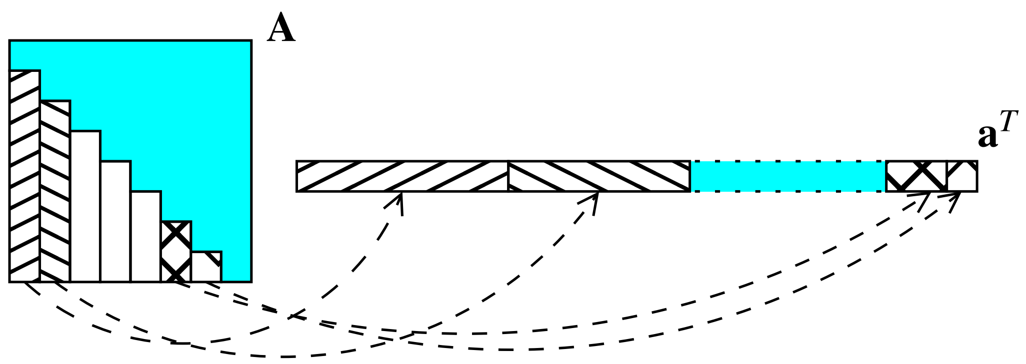
## Questions

- what is minimum # of observations (maximum compression)?
- how to find $\boldsymbol{S}$? by assuming, $||\boldsymbol{x}||_2 \approx ||\boldsymbol{A}\boldsymbol{x}||_2$

# Graph Compression

## Adjacency vector

- undirected graph $G(V,E)$ has at most $n = \frac{1}{2}(|V|^2 - |V|)$ edges




## Graph compression

- if graph $G(V,E)$ is sparse, adjacency vector $\boldsymbol{a}$ is $k$-sparse, and $k = |E| \ll n$
- define compression matrix $\boldsymbol{S}$ for a known $k$-sparse $\boldsymbol{a}$
  $\rightarrow$ dense random $\boldsymbol{S}$ may be a good choice

## Graph decompression

$$\text{no noise:} \quad \min \|\boldsymbol{a}^*\|_1 \quad \text{s.t.} \quad \boldsymbol{S}\,\boldsymbol{a}^* = \boldsymbol{S}\,\boldsymbol{a}$$

$$\text{with noise:} \quad \min \|\boldsymbol{a}^*\|_1 \quad \text{s.t.} \quad \|\boldsymbol{S}\,\boldsymbol{a}^* - \boldsymbol{y}\|_2 \leq \epsilon$$

# LINEAR STATISTICAL LEARNING*

## General model

$$\boldsymbol{Y} = \boldsymbol{L} + \boldsymbol{A}\boldsymbol{S} + \boldsymbol{W}$$

$\boldsymbol{Y}$ : observations
$\boldsymbol{L}$ : background
$\boldsymbol{A}$ : dictionary
$\boldsymbol{S}$ : sparse signal
$\boldsymbol{W}$ : additive noise

## Problems

| | |
|---|---|
| $\boldsymbol{L} = \boldsymbol{0}$, $\boldsymbol{A}$ known | compressive sampling/sensing |
| $\boldsymbol{L} = \boldsymbol{0}$ | dictionary learning, matrix factorization |
| $\boldsymbol{A} = \boldsymbol{I}$ | principal component pursuit (PCP) |
| $\boldsymbol{S} = \boldsymbol{0}$, $\boldsymbol{L}$ low-rank | principal component analysis (PCA) |

## Solutions

- minimize reconstruction error
- stochastic approximations
- off-line and online strategies
- linear algebra with random matrices

## Applications

- graph visualization
- graph compression/decompression
- clustering data
- detecting outliers and anomalies

*G. B. Giannakis, Challenges and SP Tools for Big Data Analytics, IEEE SPS Summer School, Vancouver, Canada, July 2014.

# Linear Statistical Learning (2)

## Graph and signal learning

- observations $\boldsymbol{y}_t = \boldsymbol{A}\boldsymbol{s}_t + \boldsymbol{w}_t$, such that $\boldsymbol{A} \geq 0$ is $N \times N$ and sparse
- batch of measurements, $t = 1, \ldots, T \gg N$

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{W}$$

- for Gaussian $\boldsymbol{W}$, alternate minimization of the convex loss functions

$$\begin{aligned} \boldsymbol{A}_{k+1} &= \operatorname{argmin}_{\boldsymbol{A}} \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{S}_k\|_F^2 + \lambda \sum_{n=1}^{N} \|\boldsymbol{a}_n\|_1 \quad \text{(Lasso)} \\ \boldsymbol{S}_{k+1} &= \operatorname{argmin}_{\boldsymbol{S}} \|\boldsymbol{Y} - \boldsymbol{A}_{k+1}\boldsymbol{S}\|_F^2 \end{aligned}$$

$\rightarrow$ $\lambda$ controls sparsity of $\boldsymbol{A}$

- online learning for streaming data

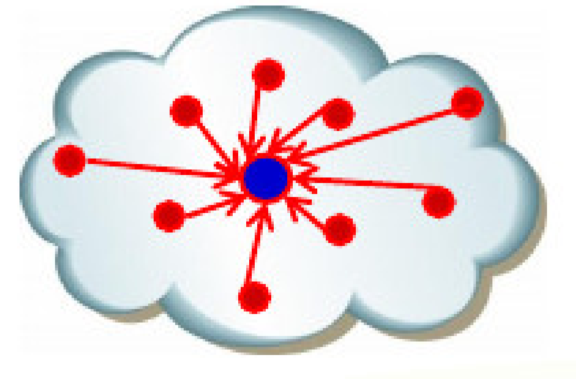
$$\begin{aligned} \boldsymbol{A}_{k+1} &= \operatorname{argmin}_{\boldsymbol{A}} \sum_{k=t-\tau}^{t} \left\|\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{s}_k\right\|_F^2 + \lambda \sum_{n=1}^{N} \|\boldsymbol{a}_n\|_1 \\ \boldsymbol{s}_{k+1} &= \operatorname{argmin}_{\boldsymbol{s}} \left\|\boldsymbol{y}_t - \boldsymbol{A}_{k+1}\boldsymbol{s}\right\|_F^2 \end{aligned}$$

- other constraints, e.g. $\boldsymbol{S}^T\boldsymbol{S} = \boldsymbol{I}$ (orthogonal signals)
- other problems, e.g. $\boldsymbol{Y} = \boldsymbol{A}(\boldsymbol{S} + \boldsymbol{S}_a) + \boldsymbol{W}$, where $\boldsymbol{S}_a$ represents anomaly

# Linear Statistical Learning (3)

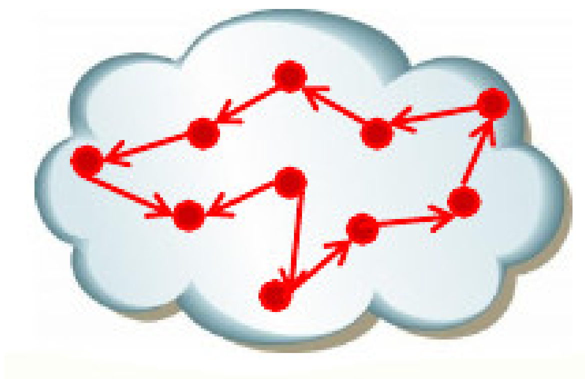
## Learning over networks: A fusion center

- difficult to scale, single point of failure
- communication bottleneck (real-time apps)
- high energy demand at the center



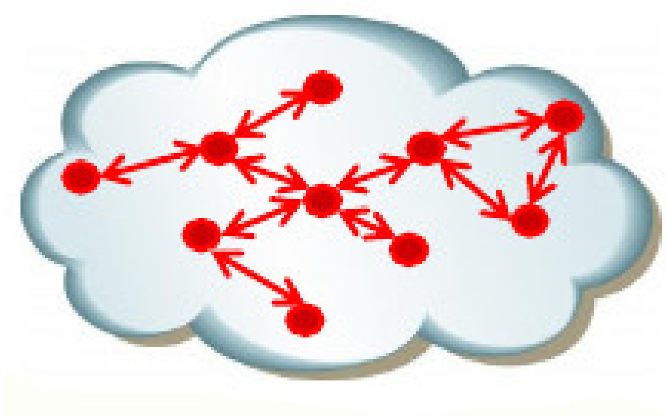
## Learning over networks: sequentially distributed

- a ring does not scale, node failures
- easy to design for static nodes
- slow learning (real-time apps)



## Learning over networks: concurrent

- difficult to design and maintain
- convergence issues
- may not use resources efficiently and fairly
- but learning can be robust and fast



# Linear Statistical Filtering

## Parameters

$$\boldsymbol{p}(t) = \boldsymbol{A}(t)\boldsymbol{p}(t-1) + \boldsymbol{b}(t) + \boldsymbol{q}(t)$$

| | | |
|---|---|---|
| $\boldsymbol{A}(t)$ | : | known transition matrix (a graph) |
| $\boldsymbol{b}(t)$ | : | known deterministic vector signal |
| $\boldsymbol{q}(t)$ | : | parameter uncertainty |

## Measurements

$$\boldsymbol{x}(t) = \boldsymbol{D}(t)\boldsymbol{p}(t) + \boldsymbol{r}(t) + \boldsymbol{w}(t)$$

| | | |
|---|---|---|
| $\boldsymbol{D}(t)$ | : | known measurement transform |
| $\boldsymbol{r}(t)$ | : | known deterministic effects |
| $\boldsymbol{w}(t)$ | : | measurement noise |

## Noise and parameter uncertainty models

$$\mathrm{cov}\left[\boldsymbol{w}(t)\boldsymbol{w}^T(\tau)\right] = \begin{cases} \boldsymbol{0} & t \neq \tau \\ \boldsymbol{W}(t) & t = \tau \end{cases}, \quad \mathrm{E}[\boldsymbol{w}(t)] = \boldsymbol{0}, \quad \mathrm{cov}\left[\boldsymbol{w}(t)\boldsymbol{p}^T(\tau)\right] = \boldsymbol{0}$$

$$\mathrm{cov}\left[\boldsymbol{q}(t)\boldsymbol{q}^T(\tau)\right] = \begin{cases} \boldsymbol{0} & t \neq \tau \\ \boldsymbol{Q}(t) & t = \tau \end{cases}, \quad \mathrm{E}[\boldsymbol{q}(t)] = \boldsymbol{0}, \quad \mathrm{cov}\left[\boldsymbol{q}(t)\boldsymbol{w}^T(\tau)\right] = \boldsymbol{0}$$

# Linear Statistical Filtering (2)

## Kalman filter

- recursive linear MMSE or best linear unbiased (BLUE) estimator
- linear measurement model with white Gaussian noise
  $\rightarrow$ only need to track changes in conditional mean and covariance
- in steady-state, the matrices $\boldsymbol{A}$, $\boldsymbol{D}$, $\boldsymbol{Q}$ and $\boldsymbol{W}$ are constant
  $\rightarrow$ Kalman filter converges Wiener filter

## Filtering steps

1. Prediction step
   - extrapolate $\boldsymbol{p}(t-1)$ as $\boldsymbol{p}_e(t)$, and $\boldsymbol{x}(t-1)$ as $\boldsymbol{D}\boldsymbol{p}_e(t)+\boldsymbol{r}(t)$
2. Correction step
   - estimate $\hat{\boldsymbol{p}}(t)=\boldsymbol{p}_e(t)+\boldsymbol{K}(t)\left[\boldsymbol{x}(t)-\boldsymbol{D}\boldsymbol{p}_e(t)-\boldsymbol{r}(t)\right]$
   - Kalman gain $\boldsymbol{K}(t)$ is computed to recursively achieve MMSE

## Two vector parameters

- two independent first-order AR processes $\boldsymbol{p}_1(t)$ and $\boldsymbol{p}_2(t)$
- measurements: $\boldsymbol{x}(t)=\boldsymbol{D}_1(t)\boldsymbol{p}_1(t)+\boldsymbol{D}_2(t)\boldsymbol{p}_2(t)+\boldsymbol{r}(t)+\boldsymbol{w}(t)$
- Task 1: separate $\boldsymbol{p}_1(t)$ and $\boldsymbol{p}_2(t)$
- Task 2: eliminate $\boldsymbol{p}_2(t)$ representing non-white noise/interference

# Generating Gamma Process

## Definition

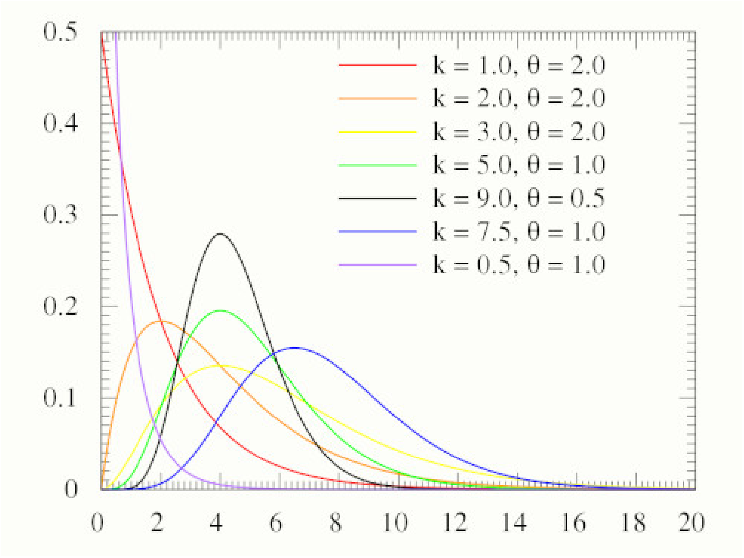
- samples are gamma distributed: $X(t) \sim \frac{\lambda^{\gamma t}}{\Gamma(\gamma t)} x^{\gamma t-1} \mathrm{e}^{-\lambda x}$, for $\forall t$
- auto-correlation: $\mathrm{corr}[X(\tau), X(t)] = \sqrt{\frac{\tau}{t}}$, $\tau < t$
- the increase per unit time has mean $\gamma/\lambda$ and variance $\gamma/\lambda^2$

## Gamma distribution

- $\mathcal{G}(X;k,\theta) \equiv \frac{1}{\Gamma(k)\theta^k} x^{k-1} \mathrm{e}^{-x/\theta}$, where $k > 0$ shape, $\theta > 0$ scale
- mean $\mu = k\theta$, variance $\nu = k\theta^2$, moments $\mathrm{E}[X^n] = \theta^n \frac{\Gamma(n+k)}{\Gamma(k)}$
- exponential and chi-square are special case of gamma distribution

## Sum of gamma random variables

- if $X_i$ are:
  1. independent
  2. gamma distributed
  3. have same scale $\theta$
- then, $Y = \sum_{i=1}^N X_i \sim \mathcal{G}(Y; \sum_{i=1}^N k_i, \theta)$
- note that, $\lim_{k\to\infty} \mathcal{G}(X;k,\theta) = \delta(X)$



# Generating Gamma Process (2)

## General strategy

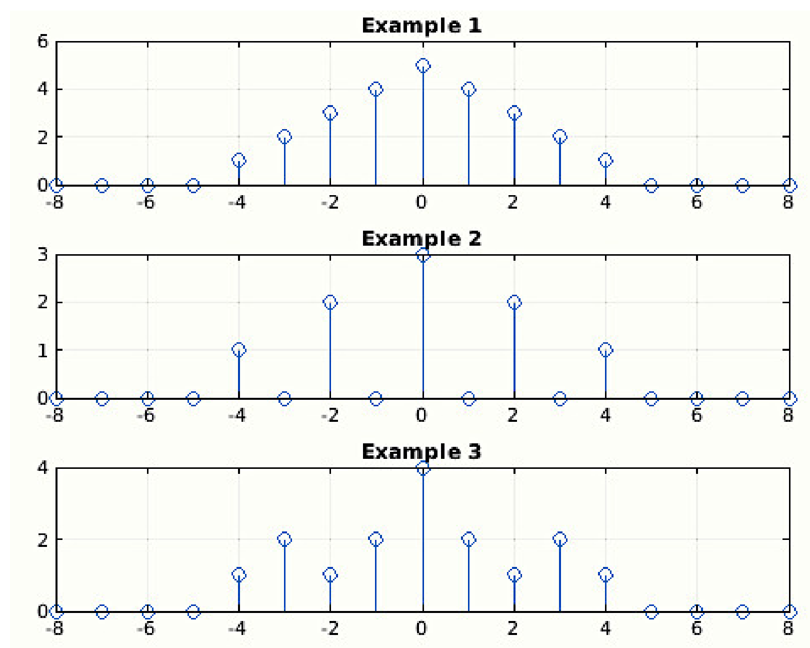
- linearly combine independent gamma distributed samples
  → available in most numerical packages (Matlab, Octave, R, Numpy, ...)

$$Y_t = h_t \otimes X_t$$

Theorem: The impulse response $h_t$ of an linear filter to linearly combine gamma samples $X_t$ must be finite, and $h_t \in \{0, c\}$ for $\forall t$ to generate a gamma process.

## Auto-covariance

- $\{X_t\}_t$ is white and $\theta = \text{const}$, and filter is LTI
- if $\{X_t\}_t$ is stationary, i.e., $k_t = \text{const}$:
  $C_Y(t) = k\theta^2 \cdot h_t \otimes h_{-t}$
- if $\{X_t\}_t$ is non-stationary, i.e., $k_t$ is *deterministic*: $C_Y(t,\tau) = \theta^2 \cdot k(t,\tau) \otimes k(-t,\tau)$




## Examples

1. $h_t = [1,1,1,1,1]$, $k_t = 1 = \text{const}$
2. $h_t = [1,0,1,0,1]$, $k_t = [1,1,0.1]$ (periodic)
3. $h_t = [1,1,0,1,1]$, $k_t = [1,2,0.5]$ (periodic)

# Take-Home Messages

1. Graph signals represent multiple stochastic processes with a defined set of pairwise constraints.

2. Linear graph signals are defined by an adjacency matrix $\boldsymbol{A}$ and linear algebra operations with or without a recursion.
   $\rightarrow$ $\boldsymbol{A}$ can be binary, sparse, or otherwise constrained

3. Many important and interesting problems involve linear graph signals.
   $\rightarrow$ Linear digital filtering
   $\rightarrow$ Linear structural causal models
   $\rightarrow$ ARMA processes and their processing
   $\rightarrow$ Graph inference/reconstruction from observations
   $\rightarrow$ Compressive sensing
   $\rightarrow$ Graph compression/decompression
   $\rightarrow$ Linear statistical learning, centralized and distributed
   $\rightarrow$ Generating graph signals

   *... and any combinations of the above*

# Other Research Topics

## More general graph signals

- non-linear, non-Gaussian, non-stationary
- time-varying and stochastic graphs
- signal-dependent noise and/or uncertainty
- discretizing random processes in high-dimensions
- causally constrained signals

## Other methods for graph signals

- generating multiple stochastic processes with pairwise constraints
- combining Network Science and signal processing methods
- implementing the methods in computing packages
- implementing distributed signal processing and associated protocols

*Thank you!*

*pavelloskot@intl.zju.edu.cn*