Structure learning in Infrastructure Networks

Rajasekhar Anguluri

Department of Computer Science and Electrical Engineering

University of Maryland, Baltimore County (UMBC)

rajangul@umbc.edu

slides template inspiration: Joel Tropp, Caltech

My Research Interests

- **inverse problems**: parameter or signal estimation problems in networks and dynamical systems
- **matrices in action**: matrices arising in applications (e.g., ML, data science, controls)
- **power systems**: formulating (and solving) problems that have linear algebraic or systems theory flavor

https://rajanguluri.github.io

Bienvenue au cours: apercu*

- **Motto:** introduce to interesting problems in the intersection of networks, optimization, and statistical learning
  - **Day 1:** learning problem in infrastructure networks
  - **Day 2:** sparse estimation: theory and algorithms
  - **Day 3:** learning structure in sparse infrastructure networks

*merci beacucoup, ChatGPT

fig sources: internet

After three days, you should be skillful in

- using conservation laws to model infrastructure networks
- developing modern statistical methods for parameter estimation problems in networks
- implementing alternating direction method of multipliers

Day 1: Overview

1. “critical” infrastructure networks
2. equilibrium (conservation) equations
3. measurement models
4. structure learning: linear model
5. structure learning: covariance model
6. wrap up

Word choices, alternatives, and explanations

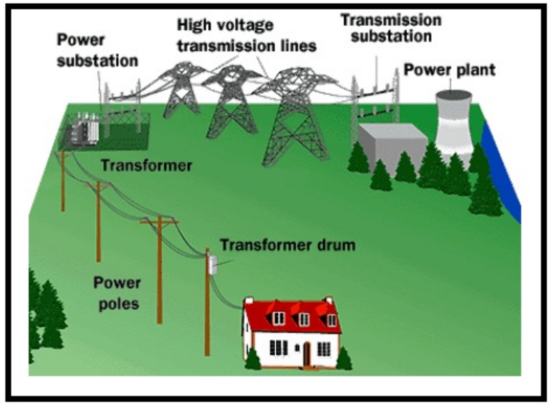
- **network:** graph
- **nodes:** buses (power system terminology)
- **structure:** topology or Laplacian matrix
- **learning:** estimation or identification
- **latent:** hidden or un-observed or partially-observed
- **complex-network:** network with complex-valued edge weights
- **currents:** flows; **voltages:** potentials
- **conservation:** equilibrium or balance

Critical Infrastructure Network Systems

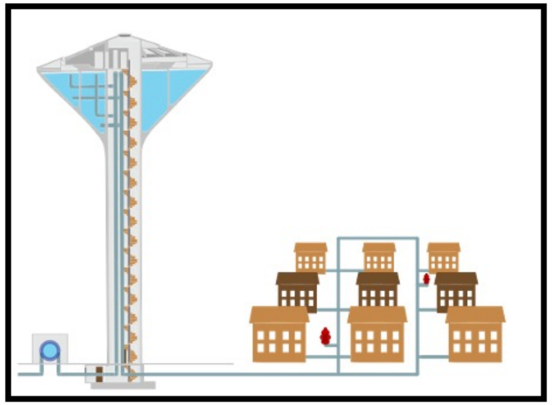
- systems that keep society running
- DHS (dept. of homeland security) identifies 16 critical sectors:

  chemical, food, energy, financial, defense, healthcare, communications, etc.

- energy systems relies on energy to meet our needs:



power system

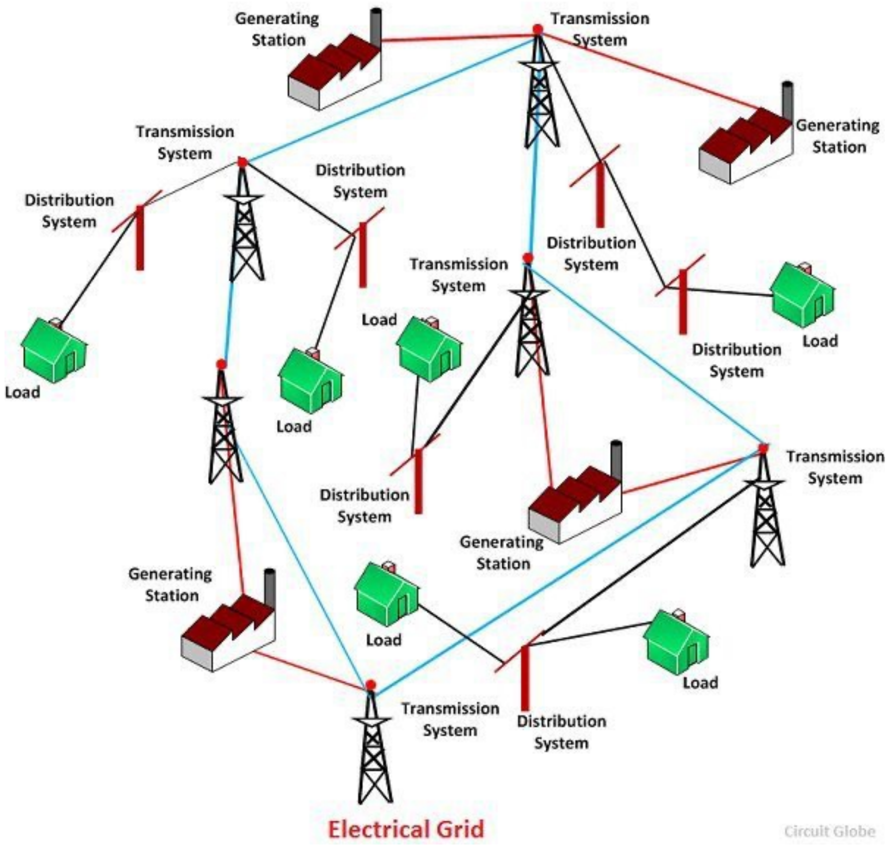


water supply system

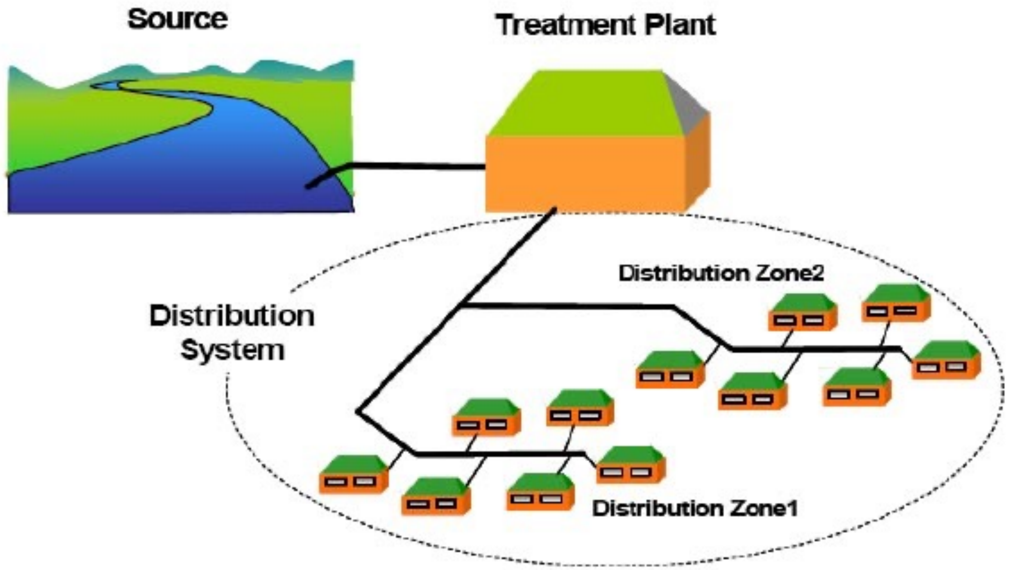
gas/oil system

Critical Infrastructure Network Systems

critical infrastructure systems are realized through ”network” of interconnected systems

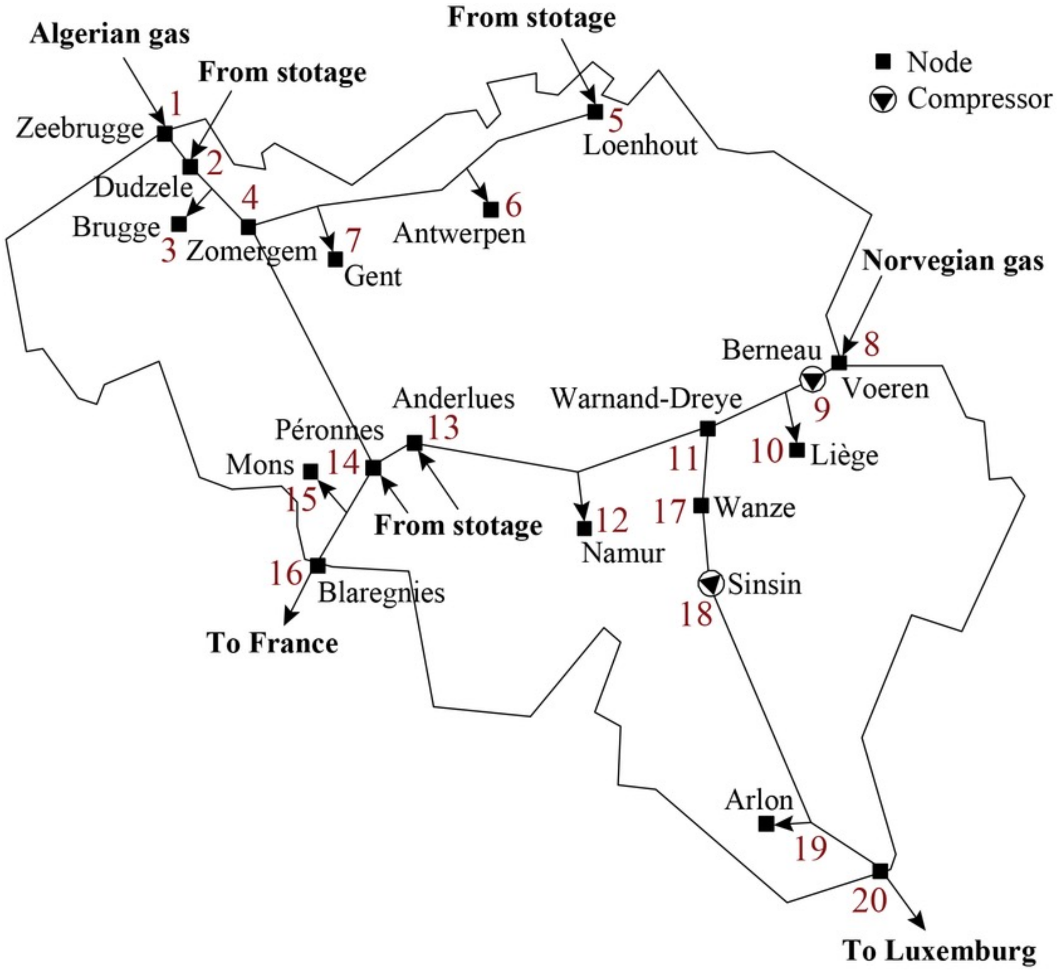


schematic of power network



schematic of water distribution network

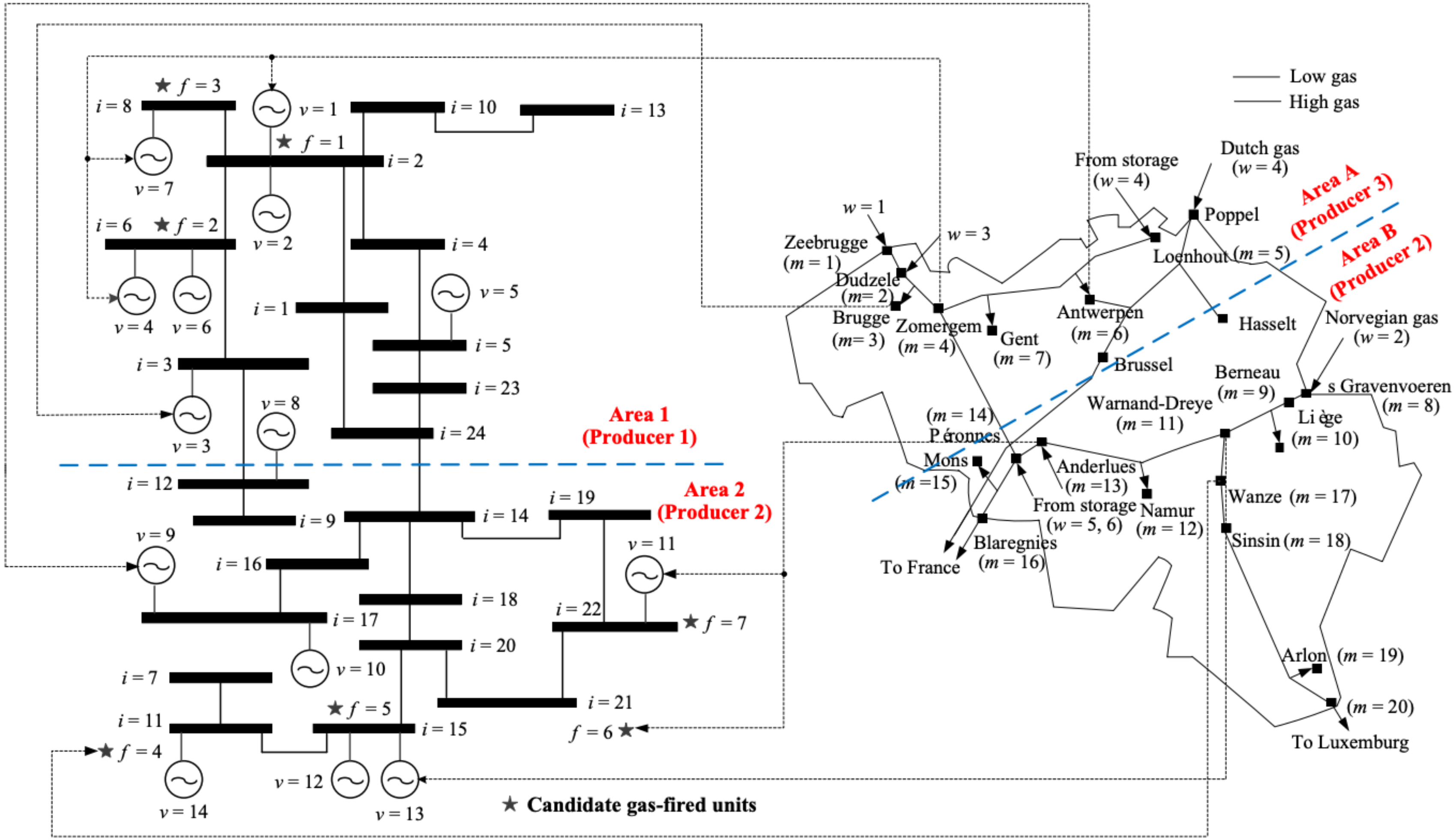
Infrastructure Networks are Sparsely Connected



Belgium natural gas network

fig source: internet

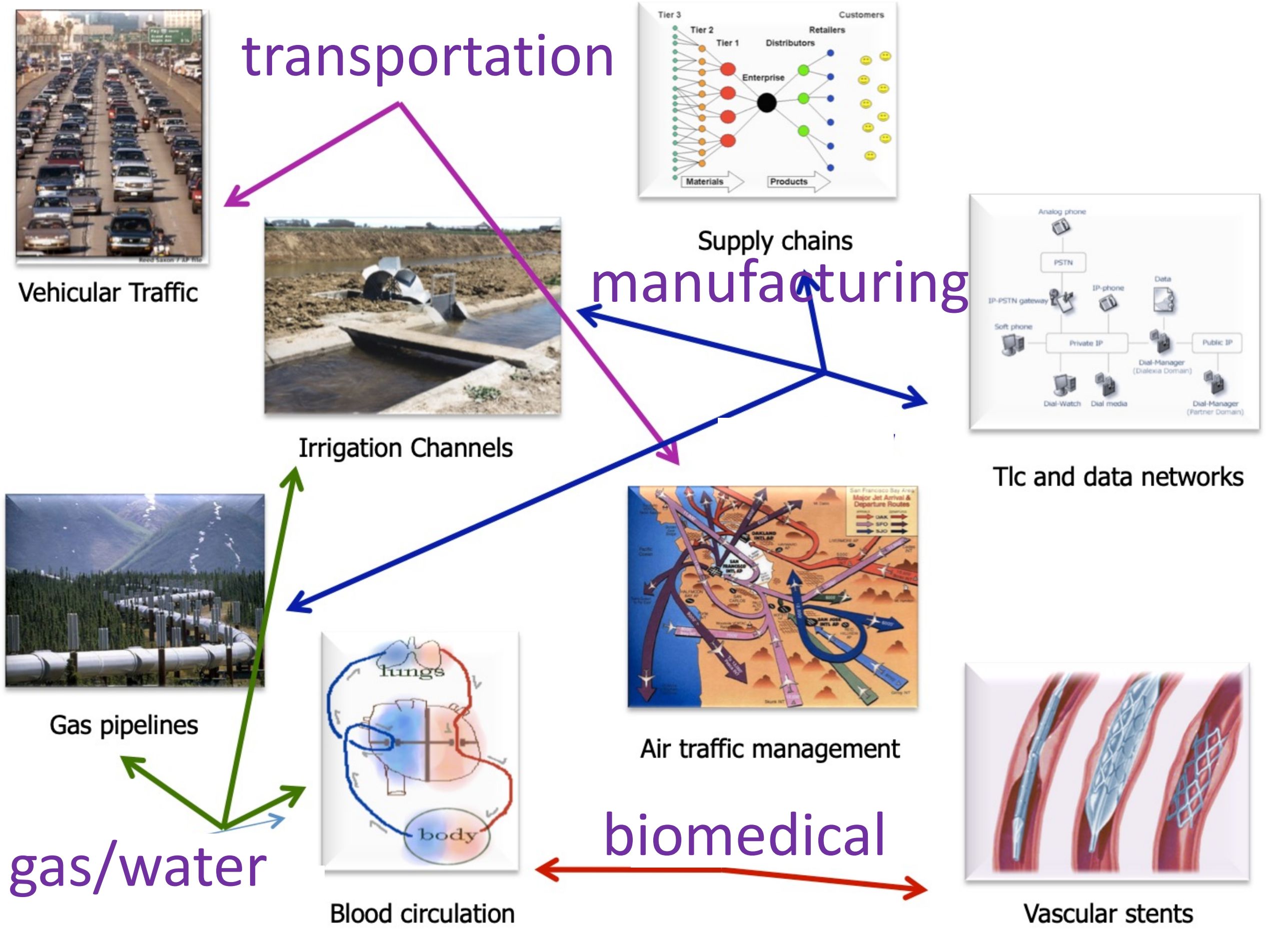
Infrastructure Networks are Sparsely Connected



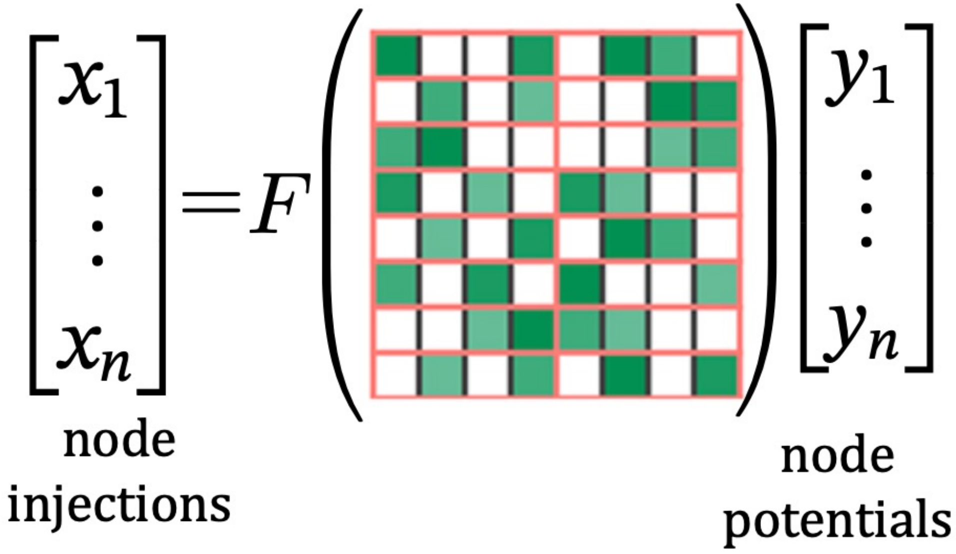
Belgium combined electric and natural gas network

fig source: internet

1. Modeling Infrastructure Networks: Conservation Laws

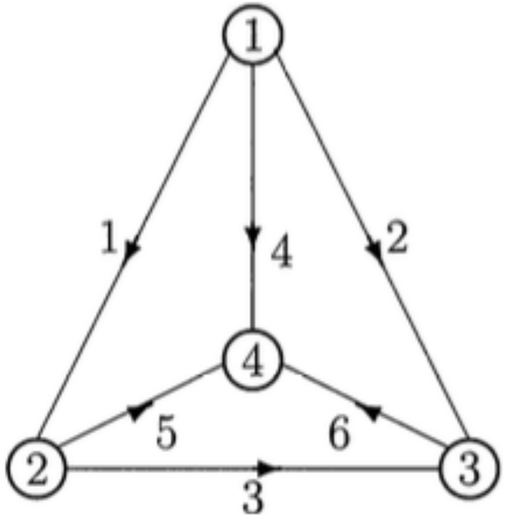


conservation law:
flow in = flow out



Graphs and Kirchoff’s and Ohm’s Laws

- **graph:** contains set of nodes (n) and set of edges (m) between them
- **incidence matrix:** signed binary matrix $A \in \{-1,0,1\}^{m \times n}$; gives connections



node

$$\begin{array}{c} \begin{matrix} \textcircled{1} & \textcircled{2} & \textcircled{3} & \textcircled{4} \end{matrix} \\ A = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} \end{array}$$

edges 1 to 6

Incidence matrix

complete graph with $m = 6$ edges and $n = 4$ nodes

Graphs and Kirchoff's and Ohm's Laws

- the incidence matrix acts as a *difference matrix:*

(differences across edges)

$$Au = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} u_2 - u_1 \\ u_3 - u_1 \\ u_3 - u_2 \\ u_4 - u_1 \\ u_4 - u_2 \\ u_4 - u_3 \end{bmatrix}$$

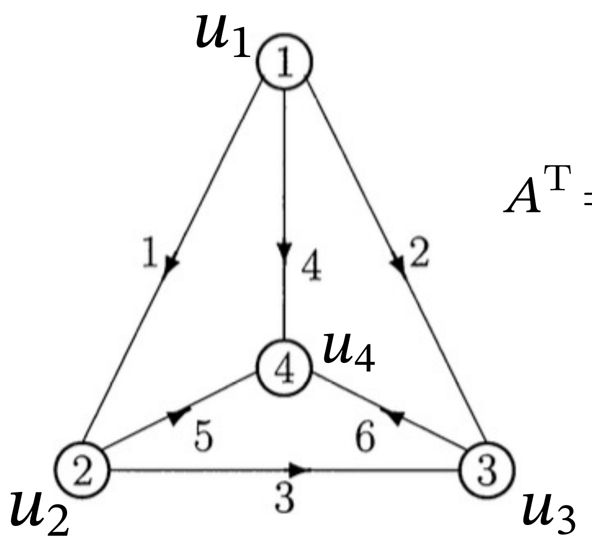
- the row vector u $= (u_1, \dots, u_n)$ is called the **node potential**

leveling (geodesy)	heights	height difference
water	pressures	pressure difference
power	voltages	voltage difference
general	potentials	**potential difference**

$$u = \begin{cases} \text{heights} \\ \text{pressures} \\ \text{voltages} \\ \text{potentials} \end{cases} \qquad Au = \begin{cases} \text{height difference} \\ \text{pressure difference} \\ \text{voltage difference} \\ \textbf{potential difference} \end{cases}$$

Graphs and Kirchoff's and Ohm's Laws

- **Kirchoff's Current Law (KCL)**: balances edge currents $w = [w_1, \dots, w_m]^T$



$$A^{\mathrm{T}} = \begin{bmatrix} -1 & -1 & 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad \begin{aligned} -w_1 - w_2 - w_4 &= 0 \quad \text{node 1} \\ w_1 - w_3 - w_5 &= 0 \quad \text{node 2} \\ w_2 + w_3 - w_6 &= 0 \quad \text{node 3} \\ w_4 + w_5 + w_6 &= 0 \quad \text{node 4} \end{aligned}$$

KCL $(A^T w = 0)$ *enforces flow in equals flow out*

- $A^T w = 0$: currents (flows) that "balance themselves" without external source
- $A^T w = f$: currents (flows) that "balance themselves" with external source
- $1^T (A^T) = 0$ implies $1^T f$=0 (that is, sum of external flows sum to zero)

Graphs and Kirchoff's and Ohm's Laws

- **null space of** A: is there a combination of the columns of A such that $Ax = 0$?

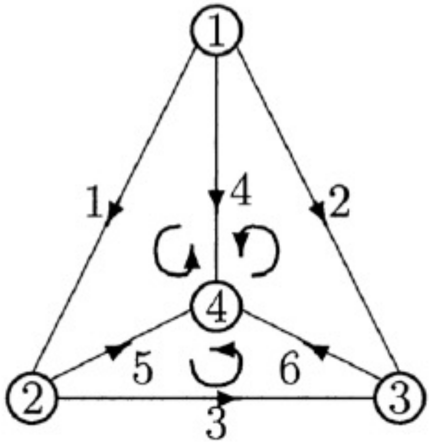
$$u = c \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad \text{will satisfy} \quad Au = 0$$

(null space is a line)

- Rank(A)=n−1, which implies $Ax = b$ cannot be solved uniquely!
- dimension of Null(A)=1 implies graph is connected (assumption)
- to make A independent: (i) add a self-loop; (ii) remove a column (set $u_n = 0$)

Graphs and Kirchoff's and Ohm's Laws

- **Kirchoff's Voltage Law (KVL)**: sum of potential differences around a loop is zero
- *Matrix language* : let $e = [e_1, \dots, e_m]^T$ be the difference vector then $e = Au$

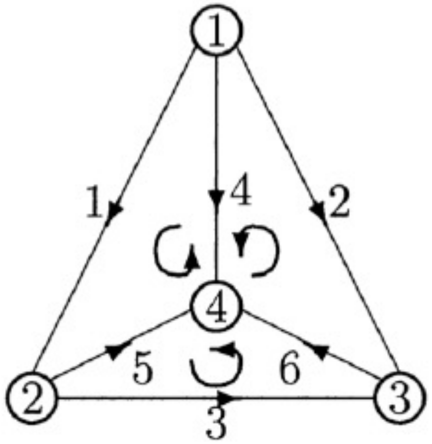


Loop flows $w = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 1 \end{bmatrix}$

loop currents (flows) solve KCL: $A^T w = 0$

Graphs and Kirchoff's and Ohm's Laws

- **Kirchoff's Voltage Law (KVL)**: sum of potential differences around a loop is zero
- *Matrix language* : let $e = [e_1, \ldots, e_m]^T$ be the difference vector then $e = Au$



Loop flows $w = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 1 \end{bmatrix}$

loop currents (flows) solve KCL: $A^T w = 0$

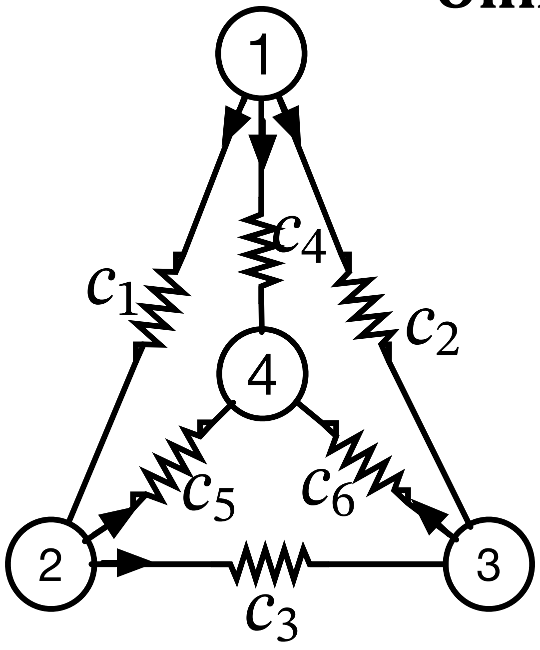
- sum of differences: $e^T w$
- KVL states: $e^T w = 0$
- Fundamental theorem of LA:

$$\mathrm{Range}(A) \perp \mathrm{Null}(A^T)$$

Main point: if $A^T w = 0$ exists $e = Au$ exists too ("potentials exist")

Graphs and Kirchoff's and Ohm's Laws

- a graph becomes a **network** when weights $c_1, \ldots, c_m$ are assigned to the edges.
- Ohm's law specifies these weights based on physical principles.



Ohm's law $w_i = c_i e_i$ (edge flow)=(conductance)(potential difference)

$$\underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}}_{w} = \underbrace{\begin{bmatrix} c_1 & & & \\ & c_2 & & \\ & & \ddots & \\ & & & c_m \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix}}_{e}$$

Equilibrium Equations: Strang Quartet

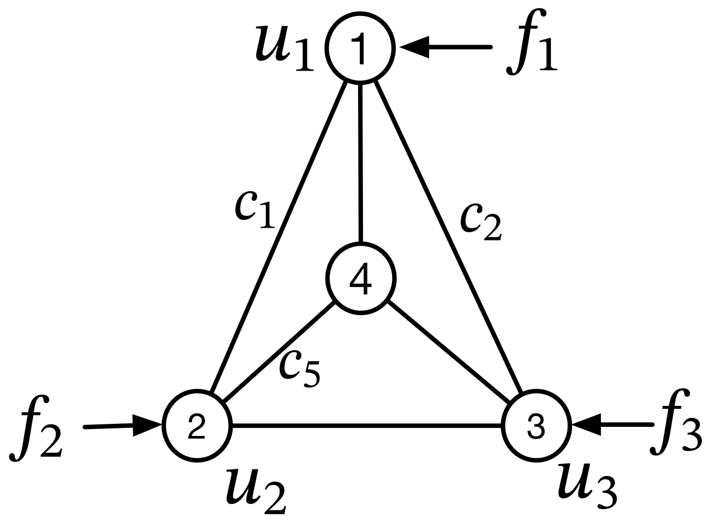
- **recipe for equilibrium equations:**

  (S1) express KVL via $\boldsymbol{e} = \boldsymbol{A}\boldsymbol{u}$

  (S1) express Ohm's law via $\boldsymbol{w} = \boldsymbol{C}\boldsymbol{e}$

  (S1) express KCL via $\boldsymbol{A}^T\boldsymbol{w} = \boldsymbol{f}$

  (S1) equilibrium equation: $\boldsymbol{A}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$



node potentials $\boldsymbol{u}$

$\Downarrow$

edge differences $\boldsymbol{e}$

$\Downarrow$

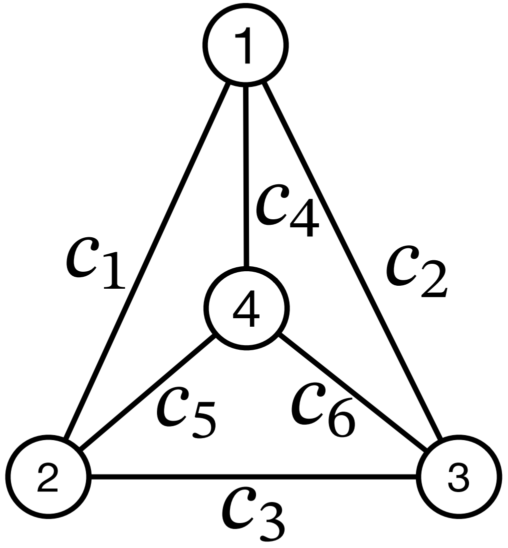
edge flows $\boldsymbol{w}$

$\Downarrow$

injected flows $\boldsymbol{f}$

Main point: equilibrium equation relate node potentials to injected flows

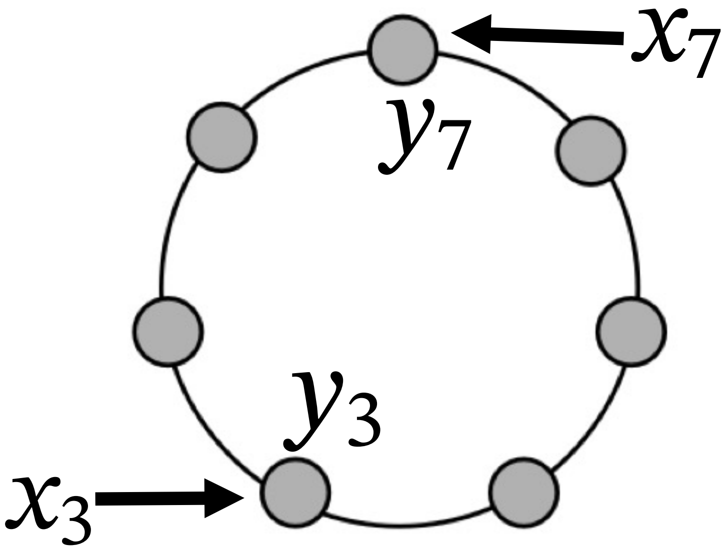
The Laplacian Matrix



$$\underbrace{\begin{bmatrix} c_1+c_2+c_4 & -c_1 & -c_2 & -c_4 \\ -c_1 & c_1+c_3+c_5 & -c_3 & -c_5 \\ -c_2 & -c_3 & c_2+c_3+c_6 & -c_6 \\ -c_4 & -c_5 & -c_6 & c_4+c_5+c_6 \end{bmatrix}}_{L=A^TCA}$$

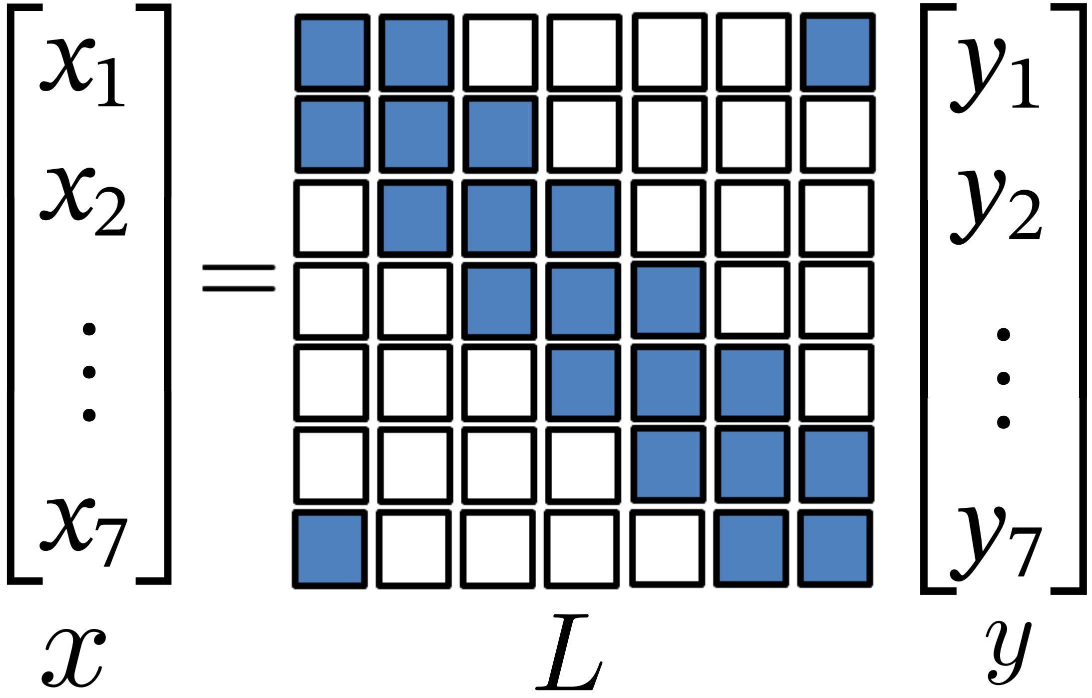
- Laplacian gives information about edge connections and weights
- Laplacian is not invertible (why?)
- reduced Laplacian is obtained by deleting a row-column pair

Network Structure = Laplacian's Sparsity Pattern



infrastructure network

sparsity (zero & non-zero) of L
captures network connections

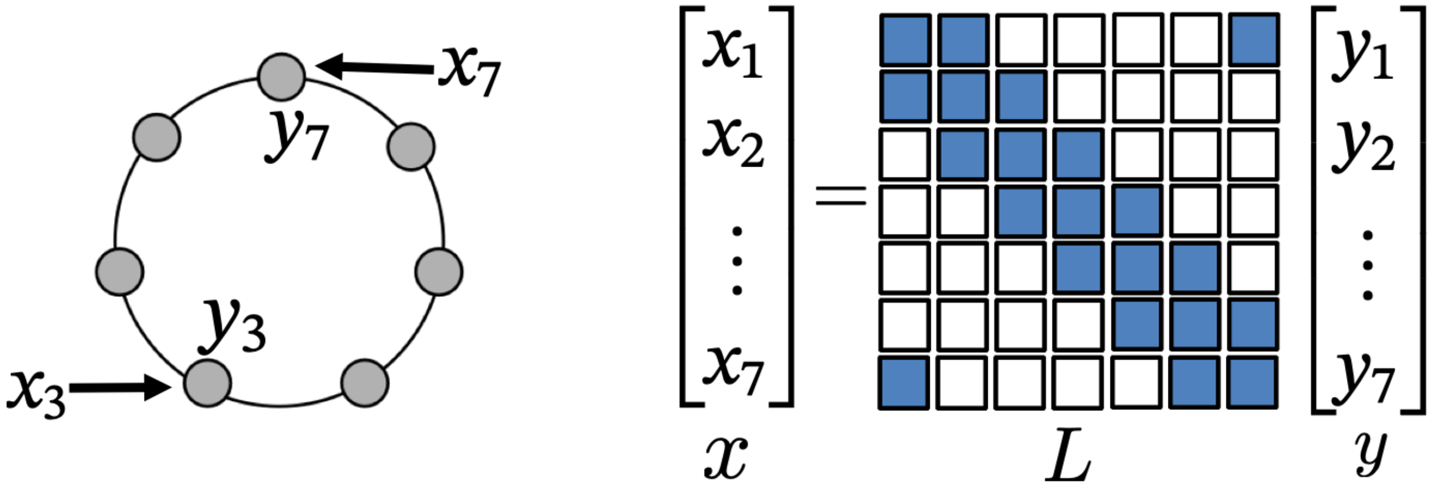


nodal injections

network Laplacian

node potentials

Measurement Models



problem: learn sparsity of L using data $x(k), y(k)$ (preferably less samples)

sensor coverage

- **full**: data from all nodes
- **partial:** data from some nodes (there are hidden nodes)

Structure Learning via Linear Models

- consider the equilibrium equation with additive noise:

$$\underbrace{\begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_p(k) \end{bmatrix}}_{x(k)} = \underbrace{\begin{bmatrix} L_{11} & L_{12} & \dots & L_{1p} \\ L_{21} & L_{22} & \dots & L_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ L_{p1} & L_{p2} & \dots & L_{pp} \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_p(k) \end{bmatrix}}_{y(k)} + \underbrace{\begin{bmatrix} \varepsilon_1(k) \\ \varepsilon_2(k) \\ \vdots \\ \varepsilon_p(k) \end{bmatrix}}_{\varepsilon(k)}$$

- stacking measurements $\{x(k), y(k)\}$ over time $k = 1, \dots, K$

$$\underbrace{\begin{bmatrix} x(1) & \cdots & x(k) \end{bmatrix}}_{X} = L \underbrace{\begin{bmatrix} y(1) & \cdots & y(k) \end{bmatrix}}_{Y} + \underbrace{\begin{bmatrix} \varepsilon(1) & \cdots & \varepsilon(k) \end{bmatrix}}_{E}$$

Laplacian Estimation: Full Coverage

- **ordinary least squares (OLS):** Matrices $X, Y \in \mathbb{R}^{p \times K}$ are known. Let $Y \in \mathbb{R}^{p \times K}$ be full row-rank matrix. Then,

$$\hat{L}_{\text{ols}} = \underset{L \in \mathbb{R}^{p \times p}}{\arg\min} \|LY - X\|_F^2$$

$$= XY^T (YY^T)^{-1}$$

- *proof*: write $||A||_F^2 = \mathrm{Tr}(AA^T)$ and set the derivative to zero
- need more samples $K \geq p$; and even more for consistency
- neglects: symmetry ($L = L^T$) and Laplacianity ($L1 = 0$)

Laplacian Estimation: Full Coverage

- **constrained least squares (CLS):**

$$\hat{L}_{\text{cls}} = \underset{L=L^T \in \mathbb{R}^{p\times p}}{\arg\min} \|LY - X\|_F^2$$

$$\text{subject to } L\mathbb{1} = 0$$

- convert the CLS problem into OLS via the vectorization trick:

$$\begin{aligned} \mathrm{Vec}(X) &= \mathrm{Vec}(LY) + \mathrm{Vec}(E) \\ &= (Y^T \otimes I_p)\,\mathrm{Vec}(L) + \mathrm{Vec}(E) \end{aligned}$$

- $\mathrm{Vec}(L) \in \mathbb{R}^{p^2}$ contains $\frac{p(p-1)}{2} + p$ redundant elements

Laplacian Estimation: Full Coverage

❧ there exists a Duplication and Transformation matrices D and T:

$$\mathrm{Vec}(L) = D \underbrace{\mathrm{Vech}(L)}_{\mathbb{R}^{[p(p-1)/2+p]}} = TD \underbrace{\mathrm{Ve}(L)}_{\mathbb{R}^{p(p-1)/2}}$$

example:

$$L = \begin{bmatrix} a_1 + a_2 & -a_1 & -a_2 \\ -a_1 & a_1 + a_3 & -a_3 \\ -a_2 & -a_3 & a_2 + a_3 \end{bmatrix} \quad \mathrm{vech}(L) = \begin{bmatrix} a_1 + a_2 \\ -a_1 \\ -a_2 \\ a_3 + a_1 \\ -a_3 \\ a_2 + a_3 \end{bmatrix} \quad \mathrm{ve}(L) = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Laplacian Estimation: Full Coverage

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{9\times 6}$$

(duplication matrix)

$$T = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}_{6\times 3}$$

(transformation matrix)

Laplacian Estimation: Full Coverage

- vectorized equilibrium equation:

$$\underbrace{\mathrm{Vec}(X)}_{z} = \underbrace{(Y^T \otimes I_p)\,\mathrm{Vec}(L)TD}_{H}\,\underbrace{\mathrm{Ve}(L)}_{\beta} + \underbrace{\mathrm{Vec}(E)}_{\eta}$$

- least squares estimators:

$$\hat{\beta} = \arg\min_{\beta} \|z - H\beta\|_2^2$$
$$\text{subject to } \mathcal{C}(\beta) \leq t$$

$$\hat{\beta} = \arg\min_{\beta} \|z - H\beta\|_2^2 + \lambda\mathcal{R}(\beta)$$

Laplacian Estimation: Partial Coverage

- partition the equilibrium equation:

$$\underbrace{\begin{bmatrix} x_O(k) \\ x_H(k) \end{bmatrix}}_{x(k)} = \underbrace{\begin{bmatrix} L_{OO} & L_{OH} \\ L_{HO} & L_{HH} \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} y_O(k) \\ y_H(k) \end{bmatrix}}_{y(k)}$$

- set the flow injection $x_H(k) = 0$ to get

$$\begin{bmatrix} x_O(k) \\ 0 \end{bmatrix} = \begin{bmatrix} L_{OO} & L_{OH} \\ L_{HO} & L_{HH} \end{bmatrix} \begin{bmatrix} y_O(k) \\ y_H(k) \end{bmatrix}$$

- here $L_{OH} = L_{HO}$; and L_{HH} is invertible (connectivity of L guarantees)

Laplacian Estimation: Partial Coverage

- Kron-reduction (aka Schur Complement):

$$\begin{bmatrix} x_O(k) \\ 0 \end{bmatrix} = \begin{bmatrix} L_{OO} & L_{OH} \\ L_{HO} & L_{HH} \end{bmatrix} \begin{bmatrix} y_O(k) \\ y_H(k) \end{bmatrix} \implies x_O = \underbrace{\left(L_{OO} - L_{OH} L_{HH}^{-1} L_{HO}\right)}_{\bar{L}} y_O$$

- solve the second block for *y_H* and substitute it in the first

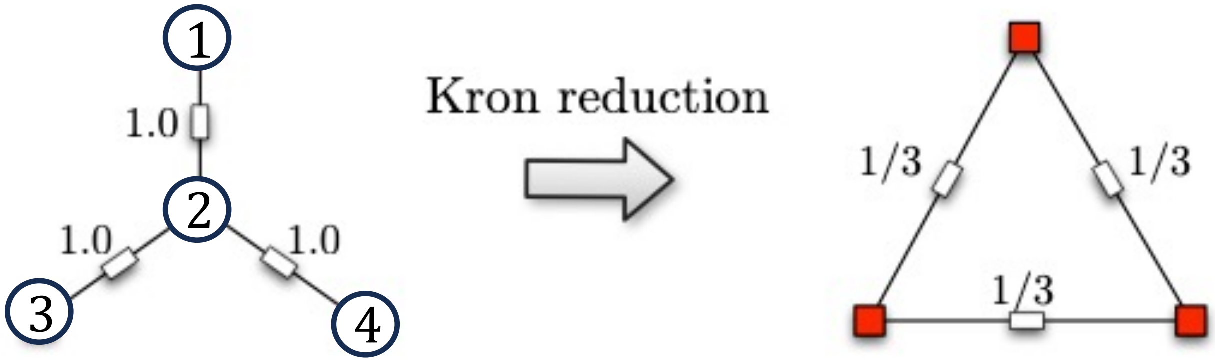


fig source: web

Laplacian Estimation: Partial Coverage

- Kron reduction for the figure on the previous slide:

$$L = \begin{bmatrix} L_{OO} & L_{OH} \\ L_{HO} & L_{HH} \end{bmatrix} = \left[\begin{array}{ccc|c} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 1 & 0 \\ \hline 0 & -1 & 0 & 1 \end{array}\right]$$

$$\overline{L} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

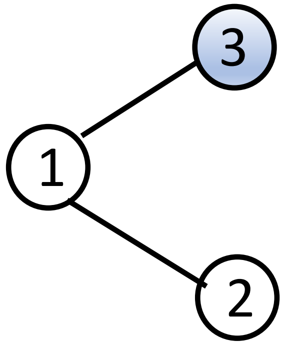
Kron Reduction is Not Injective

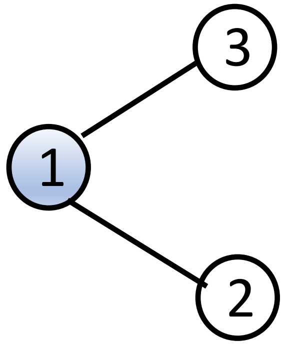
- learning $\bar{L}$ from $\{x_O(k), y_O(k)\}$ is straight forward

- is it possible to estimate L (i.e., sparsity pattern) from $\bar{L}$? No

$$\bar{L} = \begin{bmatrix} l & -l \\ -l & l \end{bmatrix} \quad L_1 = \left[\begin{array}{cc|c} l+l' & -l & -l' \\ -l & l & 0 \\ \hline -l' & 0 & l' \end{array}\right] \quad L_2 = \left[\begin{array}{cc|c} l_1 & 0 & -l_1 \\ 0 & l_2 & -l_2 \\ \hline -l_1 & -l_2 & l_1 + L_2 \end{array}\right]$$

$$l_1 l_2 / (l_1 + l_2) = l$$





- for tree networks, it is possible under certain assumptions!

Structure Learning via Covariance Models

- equilibrium equation: $y = L^{-1}x,$ where $y \in \mathbb{R}^p$
- flow injections are Gaussian: $x \sim \mathcal{N}(0, \Sigma_x)$
- so node potentials are Gaussian: $y \sim \mathcal{N}(0, L^{-1}\Sigma_x L^{-1})$
- **learning problem:** given samples $y_1, \dots, y_K$, estimate the sparsity pattern of $p{\times}p$ matrix L; preferably $(K \ll p)$
- this problem has connections to factor models: $y = \Lambda x + \Psi u$

A. Kohei and N. T. Trendafilov (2018). Some mathematical properties of the matrix decomposition solution in factor analysis, Psychometrika 83: 407-424.

Structure Learning via Covariance Models

- **naïve approach:** suppose $\Sigma_y = L^{-1}\Sigma_x L^{-1}$ and Σ_x are known. then

$$\begin{aligned}\Sigma_x^{1/2}\Sigma_y\Sigma_x^{1/2} &= (\Sigma_x^{1/2}L^{-1}\Sigma_x^{1/2})(\Sigma_x^{1/2}L^{-1}\Sigma_x^{1/2}) \\ &= (\Sigma_x^{1/2}L^{-1}\Sigma_x^{1/2})^2\end{aligned}$$

- take the principal square root, and pre- and post-multiply $\Sigma_x^{-1/2}$

$$\Sigma_x^{-1/2}\left(\sqrt{\Sigma_x^{1/2}\Sigma_y\Sigma_x^{1/2}}\right)\Sigma_x^{-1/2} = L^{-1}$$

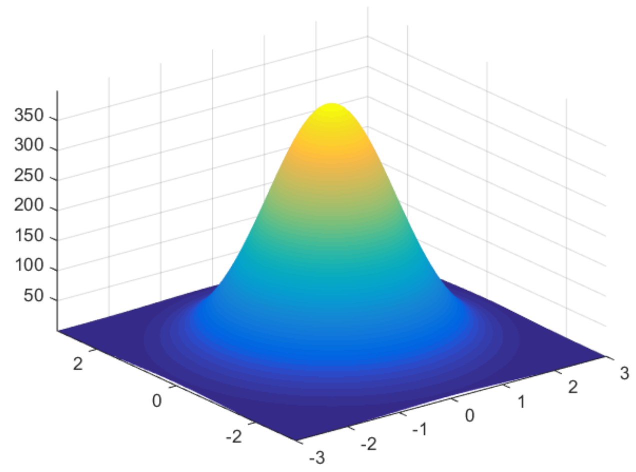
- approach fails: (i) Σ_y is replaced with estimate; (ii) Σ_x is unknown

Maximum Likelihood Estimate of $\boldsymbol{\Sigma_y}$

- let $y \sim \mathcal{N}(0, \Sigma_y)$. then,

$$f_{\mathbf{Y}}(y) = \frac{1}{\sqrt{(2\pi)^p \det(\Sigma_y)}} \exp\left(-y^T \Sigma_y^{-1} y/2\right)$$

$$\Sigma_y = \mathbb{E}[yy^T]$$



Lemma: Given i.i.d samples $y_1, \ldots, y_K \sim \mathcal{N}(0, \Sigma_y)$. The maximum-likelihood estimate of Σ_y (if $K \geq p$) is the sample covariance matrix:

$$S = \frac{1}{K} \sum_{k=1}^{K} y_k y_k^T = \frac{1}{K} YY^T$$

Maximum Likelihood Estimate of Σ_y

non-standard proof: ML-estimate is given by:

$$\rightarrow \max_{\Sigma \succ 0} -\frac{n}{2}\left(\log|\Sigma| + \mathrm{Tr}[S\Sigma^{-1}]\right)$$

determinant

$$\Updownarrow$$

$$\min_{\Sigma \succ 0} \log|\Sigma| + \mathrm{Tr}[S\Sigma^{-1}]$$

trick: $\log|\Sigma| + \mathrm{tr}[\Sigma^{-1}S] - \log|S| - \mathrm{tr}[S^{-1}S]$

$$\log|\Sigma S^{-1}| + \mathrm{tr}[\Sigma^{-1}S] - \mathrm{tr}[I] \rightarrow n$$

J. Gentle (2022). Matrix Algebra: Theory, Computations and Applications in Statistics, Third Edition, Springer. (Page 510, Chapter 9)

Maximum Likelihood Estimate of Σ_y

$$= -\log|\Sigma^{-1}S| + tr[\Sigma^{-1}S] - n$$

$$= -\log\left(\prod_i c_i\right) + \sum_i c_i - n$$

$$= \sum_i (-\log c_i + c_i - 1) \geq 0 \quad \text{——— (1)}$$

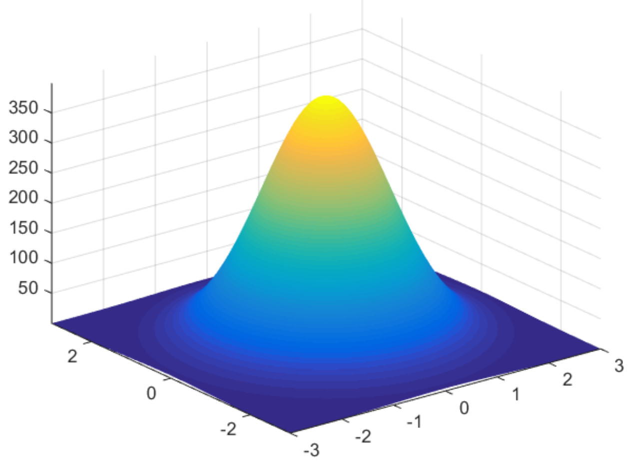
- $c_i > 0$ are eigen values of $\Sigma^{-1}S$
- minimum occurs at $c_i = 1$, which means $\Sigma^{-1}S = I$

or $\boxed{\hat{\Sigma}_{MLE} = S}$

J. Gentle (2022). Matrix Algebra: Theory, Computations and Applications in Statistics, Third Edition, Springer. (Page 510, Chapter 9)

Maximum Likelihood Estimate of $\Omega_y = \Sigma_y^{-1}$

- let $y \sim \mathcal{N}(0, \Sigma_y)$. then,



$$f_{\mathbf{Y}}(y) = \frac{1}{\sqrt{(2\pi)^p \det(\Sigma_y)}} \exp\left(-y^T \Sigma_y^{-1} y/2\right)$$

$$\Sigma_y = \mathbb{E}[yy^T] = L^{-1}\Sigma_x L^{-1}$$ (covariance matrix)

$$\Omega_y = L\Sigma_x^{-1}L$$ (inverse covariance or precision matrix)

- working with Ω_y is easier because it involves L

Maximum Likelihood Estimate of $\boldsymbol{\Omega_y = \Sigma_y^{-1}}$

- let Y be the samples from zero-mean Gaussian with precision Ω_y
- the rescaled log-likelihood is

$$\mathcal{L}(\Omega_y; Y) = \frac{1}{K}\sum_{k=1}^{K} \log f_y(y_i) = \log\det(\Omega_y) - \mathrm{Tr}(S\Omega_y)$$

- S is the sample covariance matrix; and

$$\log\det(\Omega_y) = \begin{cases} \sum_{j=1}^{p} \log\left(\lambda_j(\Omega_y)\right) & \text{if } \Omega_y \succ 0 \\ -\infty; & \text{otherwise} \end{cases}$$

- $\lambda_j(\Omega_y)$ is the j-th eigenvalue of Ω_y

Maximum Likelihood Estimate of $\Omega_y = \Sigma_y^{-1}$

- consider the unconstrained maximum likelihood estimator:

$$\hat{\Omega}_y \in \arg\max_{\Omega_y \succeq 0} \{\log\det(\Omega_y) - \mathrm{trace}(S\Omega_y)\}$$

- $\widehat{\Omega}_y$ converges to Ω as the number of samples $K \to \infty$

Lemma: Given i.i.d samples $y_1, \ldots, y_K \sim \mathcal{N}(0, \Sigma_y)$. The maximum-likelihood estimate of Ω_y (under sample, $K \leq p$) **fails** to exist:

proof: exercise (previous technique will not work); we "may" revisit this tomorrow.

Maximum Likelihood Estimate of $\boldsymbol{\Omega_y = \Sigma_y^{-1}}$

- two-stage constrained maximum likelihood estimator:

$$\hat{\Omega}_y \in \arg\max_{\Omega_y \succeq 0} \{\log\det(\Omega_y) - \mathrm{trace}(S\Omega_y)\}$$

$$\text{subject to } \mathcal{C}(\Omega_y) \leq t$$

- constraint could be sparsity or low-rank property
- process $\widehat{\Omega}_y$ to estimate L (the Laplacian matrix)

Maximum Likelihood Estimate of $\Omega_y = \Sigma_y^{-1}$

- direct constrained maximum likelihood estimator:

$$\hat{L} \in \arg\max_{L \in S_L} \{\log\det(\Omega_y) - \mathrm{trace}(S\Omega_y)\}$$

$$\text{subject to} \quad \Omega_y = L\Sigma_x^{-1}L$$

$$S_L = \{L : L_{ij} \leq 0 \; i \neq j, \; L1 = 0\}$$

$$\mathcal{C}(L) \leq t$$

- constraint could enforce sparsity or low-rank property

Wrap Up

- equilibrium equations help us model infrastructure networks
- **linear models:** nee data from “potentials” and “injections”
- **covariance models:** need “injections” & statistics of potentials

$$\hat{L} = \arg\min_L \; \mathrm{Loss}(L; x(k), y(k))$$

$$\text{subject to } \; L = \text{Laplacian}$$

To learn more…

Contact: rangulur@asu.edu (will send my lecture notes soon)

https://rajanguluri.github.io

Books:

- G. Strang (2019). Linear Algebra and Learning from Data, Wellelsey Publishers.
- G. Strang (2007). Computational Science and Engineering, Wellelsey Publishers.

Papers:

- G. Strang (1988) A framework for equilibrium equations, SIAM Review 30(2), 283-297
- Y. Yuan et. Al. (2022) Inverse power flow problem, IEEE Transactions on Control of Network Systems, 10(1), 261-273
- D. Deka, V. Kekatos, G. Cavraro (2023) Learning distribution grid topologies: a tutorial, IEEE Transactions on Smart Grids, 15(1), 991-1013
- Rayas. A, R. Anguluri, and G. Dasarathy (2022) Learning the structure of large networked systems obeying conservation laws, NeuRIPS, 35(2022): 14637-14650