A Density Evolution Framework for Recovery of Covariance and Causal Graphs from Compressed Measurements

Murali G Sethuraman[1] Hang Zhang[2] Faramarz Fekri[1]

[1]School of ECE, Georgia Insititute of Technology

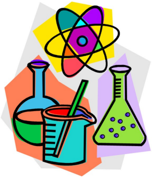
[2]Amazon, USA

Georgia Institute of Technology

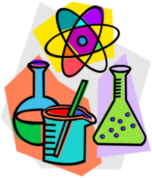
ALLERTON CONFERENCE
September 26-29, 2023 at Monticello, Illinois

Motivation

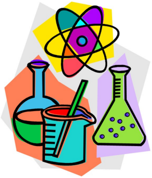
Motivation

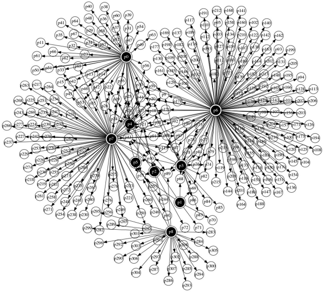


Motivation

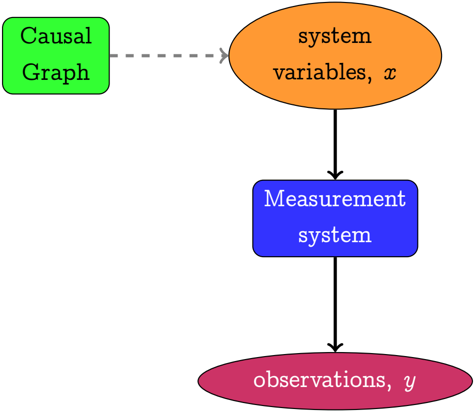


Motivation



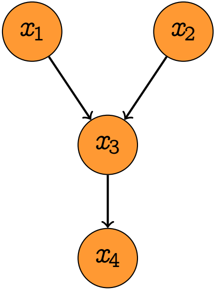


Motivation

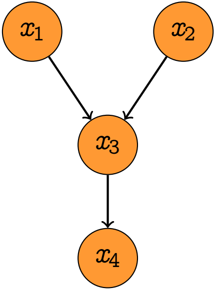


Causal Graph
system variables, x
Measurement system
observations, y

Motivation



Motivation



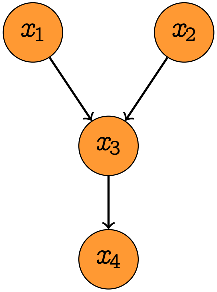
Linear Gaussian Structural Equations

$$x_i = \sum_{x_j \in \mathrm{Pa}(x_i)} w_{ji} x_j + \varepsilon_i$$

$$\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$$

Objective: Recover the *edge set*, E.

Motivation



Linear Gaussian Structural Equations

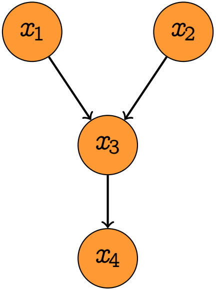
$$x = W^\top x + \varepsilon$$
$$\varepsilon \sim \mathcal{N}(0, \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2))$$

Thus $x \sim \mathcal{N}(0, \Sigma_x)$,

$$\boxed{\Sigma_x = (I - W^\top)^{-1} \mathrm{d}(\sigma_1^2, \ldots, \sigma_p^2)(I - W^\top)^{-\top}}$$

Objective: Recover the *edge set*, E.

Motivation



Linear Gaussian Structural Equations

$$x = W^\top x + \varepsilon$$
$$\varepsilon \sim \mathcal{N}(0, \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2))$$

Thus $x \sim \mathcal{N}(0, \Sigma_x)$,

$$\boxed{\Sigma_x = (I - W^\top)^{-1} \mathrm{d}(\sigma_1^2, \ldots, \sigma_p^2)(I - W^\top)^{-\top}}$$

Contains everything we need

Need good estimator

Objective: Recover the *edge set*, E. (recover Σ_x)

Problem Setup

Linear Measurement System

Let $x \in \mathbb{R}^p$. Consider the following measurement system,

$$y = Ax$$

- $y \in \mathbb{R}^d$ where $(d < p)$ - indirect observations.
- $A \in \mathbb{R}^{d \times p}$ - sensing matrix.

In this case, $\boxed{\Sigma_y = A\Sigma_x A^\top}$.

Problem Setup

Linear Measurement System

Let $x \in \mathbb{R}^p$. Consider the following measurement system,

$$y = Ax$$

- $y \in \mathbb{R}^d$ where $(d < p)$ - indirect observations.
- $A \in \mathbb{R}^{d \times p}$ - sensing matrix.

In this case, $\boxed{\Sigma_y = A\Sigma_x A^\top}$.

Goal

Design a sparse sensing matrix A for recovering Σ_x from y.

Problem Setup

Linear Measurement System

Let $x \in \mathbb{R}^p$. Consider the following measurement system,

$$y = Ax$$

- $y \in \mathbb{R}^d$ where $(d < p)$ - indirect observations.
- $A \in \mathbb{R}^{d \times p}$ - sensing matrix.

In this case, $\boxed{\Sigma_y = A\Sigma_x A^\top}$.

Goal

Design a sparse sensing matrix A for recovering Σ_x from y.

- What makes a good A?

Problem Setup

Linear Measurement System

Let $x \in \mathbb{R}^p$. Consider the following measurement system,

$$y = Ax$$

- $y \in \mathbb{R}^d$ where $(d < p)$ - indirect observations.
- $A \in \mathbb{R}^{d \times p}$ - sensing matrix.

In this case, $\boxed{\Sigma_y = A\Sigma_x A^\top}$.

Goal

Design a sparse sensing matrix A for recovering Σ_x from y.

- What makes a good A?
- Incorporating additional constraints.

Covariance Recovery

Assume: Σ_x is sparse.

Recovery Problem

$$\min_{\Sigma_x} \|\Sigma_x\|_1 \quad \text{s.t.} \quad \Sigma_y = A\Sigma_x A^\top$$

Covariance Recovery

Assume: Σ_x is sparse.

Recovery Problem (finite samples)

$$\min_{\Sigma_x} \|\Sigma_x\|_1 \quad \text{s.t.} \quad \left\|\Sigma_y^{(n)} - A\Sigma_x A^\top\right\|_2 \leq \tau$$

Covariance Recovery

Assume: Σ_x is sparse.

Recovery Problem (finite samples)

$$\min_{\Sigma_x} \|\Sigma_x\|_1 \quad \text{s.t.} \quad \left\|\Sigma_y^{(n)} - A\Sigma_x A^\top\right\|_2 \leq \tau$$

Recovery Problem (finite samples) + (vectorization)

$$\min_{\chi} \left\|\gamma^{(n)} - A^{\otimes}\chi\right\|_2 + \beta\|\chi\|_1$$

where

- $\chi = \text{vec}(\Sigma_x)$.
- $\gamma = \text{vec}(\Sigma_y)$.
- $A^{\otimes} = A \otimes A$.

Covariance Recovery

Assume: Σ_x is sparse.

Recovery Problem (finite samples)

$$\min_{\Sigma_x} \|\Sigma_x\|_1 \quad \text{s.t.} \quad \left\|\Sigma_y^{(n)} - A\Sigma_x A^\top\right\|_2 \leq \tau$$

Recovery Problem (finite samples) + (vectorization)

$$\min_{\chi} \left\|\gamma^{(n)} - A^{\otimes}\chi\right\|_2 + \beta\|\chi\|_1$$

where

- $\chi = \text{vec}(\Sigma_x)$.
- $\gamma = \text{vec}(\Sigma_y)$.
- $A^{\otimes} = A \otimes A$.

Covariance Recovery

Assume: Σ_x is sparse.

Recovery Problem (finite samples)

$$\min_{\Sigma_x} \|\Sigma_x\|_1 \quad \text{s.t.} \quad \left\|\Sigma_y^{(n)} - A\Sigma_x A^\top\right\|_2 \leq \tau$$

Recovery Problem (finite samples) + (vectorization)

$$\min_{\chi} \left\|\gamma^{(n)} - A^{\otimes}\chi\right\|_2 + \beta\|\chi\|_1$$

where

- $\chi = \text{vec}(\Sigma_x)$.
- $\gamma = \text{vec}(\Sigma_y)$.
- $A^{\otimes} = A \otimes A$.

Design "good" A for recovery of χ!

Factor graph inference

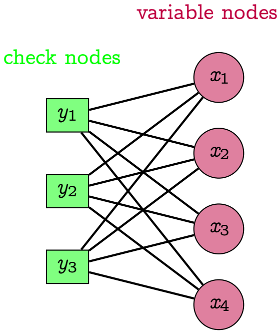
In general for,

$$\hat{x} = \arg\min_x \sum_j \left(y_j - \sum_i A_{ji} x_i \right)^2 + \beta \|x\|_1$$

Factor graph inference

In general for,

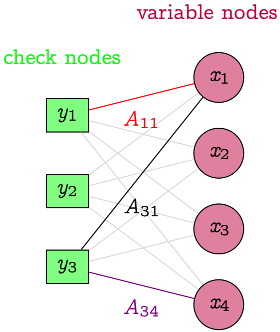
$$\hat{x} = \arg\max_x e^{-\sum_j \left(y_j - \sum_i A_{ji} x_i\right)^2} e^{-\beta \|x\|_1}$$



Factor graph inference

In general for,

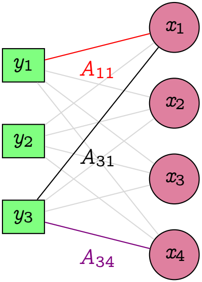
$$\hat{x} = \arg\max_x e^{-\sum_j \left(y_j - \sum_i A_{ji} x_i\right)^2} e^{-\beta ||x||_1}$$



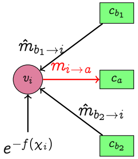
Factor graph inference

In general for,

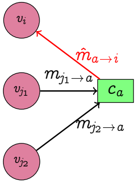
$$\hat{x} = \arg\max_x e^{-\sum_j \left(y_j - \sum_i A_{ji} x_i\right)^2} e^{-\beta \|x\|_1}$$



Variable to check node

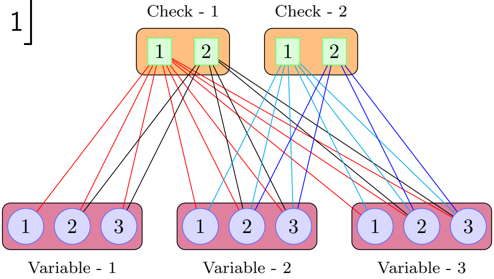


Check to variable node



Factor graph - Knronecker product matrix

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$



Analyzing the messages

Density Evolution

- Used in coding theory to design LDPC codes.
- $m_{i\to a}^{(t)} \sim \mathcal{N}(\mu_{i\to a}^{(t)}, v_{i\to a}^{(t)})$ and $\hat{m}_{a\to i}^{(t)} \sim \mathcal{N}(\hat{\mu}_{a\to i}^{(t)}, \hat{v}_{a\to i}^{(t)})$.
- Convergence analyzed using the following quantities

$$E^{(t)} = \frac{1}{d^2p^2}\sum_{a=1}^{d^2}\sum_{i=1}^{p^2}\left(\mu_{i\to a}^{(t)} - \chi_i\right)^2; \quad V^{(t)} = \frac{1}{d^2p^2}\sum_{a=1}^{d^2}\sum_{i=1}^{p^2} v_{i\to a}^{(t)}.$$

av. message error

av. message variance

Analyzing the messages

Density Evolution

- Used in coding theory to design LDPC codes.
- $m^{(t)}_{i\to a} \sim \mathcal{N}(\mu^{(t)}_{i\to a}, v^{(t)}_{i\to a})$ and $\hat{m}^{(t)}_{a\to i} \sim \mathcal{N}(\hat{\mu}^{(t)}_{a\to i}, \hat{v}^{(t)}_{a\to i})$.
- Convergence analyzed using the following quantities

$$E^{(t)} = \frac{1}{d^2p^2}\sum_{a=1}^{d^2}\sum_{i=1}^{p^2}\left(\mu^{(t)}_{i\to a} - \chi_i\right)^2; \quad V^{(t)} = \frac{1}{d^2p^2}\sum_{a=1}^{d^2}\sum_{i=1}^{p^2} v^{(t)}_{i\to a}.$$

av. message error

av. message variance

"Good" $A \implies \boxed{\lim_{t\to\infty}(E^{(t)}, V^{(t)}) = (0,0)}$.

Analyzing the messages

Density Evolution

- Used in coding theory to design LDPC codes.
- $m_{i\to a}^{(t)} \sim \mathcal{N}(\mu_{i\to a}^{(t)}, v_{i\to a}^{(t)})$ and $\hat{m}_{a\to i}^{(t)} \sim \mathcal{N}(\hat{\mu}_{a\to i}^{(t)}, \hat{v}_{a\to i}^{(t)})$.
- Convergence analyzed using the following quantities

$$E^{(t)} = \frac{1}{d^2p^2}\sum_{a=1}^{d^2}\sum_{i=1}^{p^2}\left(\mu_{i\to a}^{(t)} - \chi_i\right)^2; \quad V^{(t)} = \frac{1}{d^2p^2}\sum_{a=1}^{d^2}\sum_{i=1}^{p^2} v_{i\to a}^{(t)}.$$

av. message error

av. message variance

"Good" $A \implies \boxed{\lim_{t\to\infty}(E^{(t)}, V^{(t)}) = (0,0)}$.

not practical!

Designing good sensing matrices

Define: $\lambda(\cdot)$, and $\rho(\cdot)$ to be the distribution of non-zero entries in the columns and rows of A.

Theorem

$\Sigma_x - k^2$ sparse and $\beta = p^2/(c_0 \log(p/k))$ for $c_0 > 0$. Then $a_1^2 \le p^2/k^2$ and $a_2 \le p^2/(2c_0 k^2 \log(p/k))$ implies $(E^{(t)}, V^{(t)}) \to (0,0)$. Where,

$$a_1 = \sum_{i,i',j,j'} \rho_i \rho_{i'} \lambda_j \lambda_{j'} \sqrt{ii'/jj'}$$

$$a_2 = \sum_{i,i',j,j'} \rho_i \rho_{i'} \lambda_j \lambda_{j'} (ii'/jj')$$

Designing good sensing matrices

$$\min_{\substack{\lambda \in \Delta_{d_v};\\ \rho \in \Delta_{d_c}}} \quad \frac{d}{p} = \frac{\sum_{i \geq 2} i \lambda_i}{\sum_{j \geq 2} j \rho_j} \quad \text{s.t} \quad a_1^2 \leq \frac{p^2}{k^2}; a_2 \leq \frac{p^2}{2 c_0 k^2 \log(p/k)}$$

- Let λ^* and ρ^* be the solution, sample A that satisfies λ^* and ρ^*.
- For every nonzero entry of A,

$$P(A_{ij} = A^{-1/2}) = P(A_{ij} = -A^{-1/2}) = \frac{1}{2}$$

Preferential recovery

- Certain nodes are given more importance

$$x = \begin{bmatrix} x_H \\ x_L \end{bmatrix} \quad \Sigma_x = \begin{bmatrix} \Sigma_{HH} & \Sigma_{HL} \\ \Sigma_{LH} & \Sigma_{LL} \end{bmatrix}$$

Preferential recovery

- Certain nodes are given more importance

$$x = \begin{bmatrix} x_H \\ x_L \end{bmatrix} \quad \Sigma_x = \begin{bmatrix} \Sigma_{HH} & \Sigma_{HL} \\ \Sigma_{LH} & \Sigma_{LL} \end{bmatrix}$$

- Key requirements:
  1. $\left(V_{HH}^{(t)}, V_{HL}^{(t)}, V_{LL}^{(t)}\right) \to (0, 0, 0)$. (zero message variance)

Preferential recovery

- Certain nodes are given more importance

$$x = \begin{bmatrix} x_H \\ x_L \end{bmatrix} \quad \Sigma_x = \begin{bmatrix} \Sigma_{HH} & \Sigma_{HL} \\ \Sigma_{LH} & \Sigma_{LL} \end{bmatrix}$$

- Key requirements:
  1. $\left(V_{HH}^{(t)}, V_{HL}^{(t)}, V_{LL}^{(t)}\right) \to (0, 0, 0)$. (zero message variance)
  2. We want $|\delta_{E,HH}^{(t)}| \leq |\delta_{E,HL}^{(t)}|$ and $|\delta_{E,HH}^{(t)}| \leq |\delta_{E,LL}^{(t)}|$ for all $t \geq T_0$ for some T_0. (HH has higher priority)

Preferential recovery

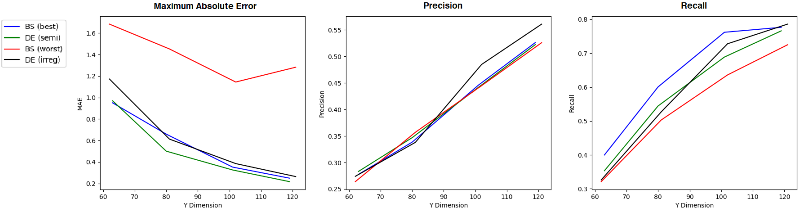
Define: degree distribution for the different parts of the sensing matrix i.e., $\lambda_H(\cdot), \lambda_L(\cdot), \rho_H(\cdot)$ and $\rho_L(\cdot)$.

Sensing Matrix Design

$$\min_{\substack{\lambda_H;\lambda_L;\\ \rho_H;\rho_L}} \quad \frac{d}{p} = \frac{n_L \sum_i i\lambda_{L,i} + n_H \sum_i i\lambda_{H,i}}{\sum_j j(\rho_{H,j} + \rho_{L,j})} \quad \text{s.t} \quad \text{Req. 1, and 2;}$$

Results - regular covariance recovery

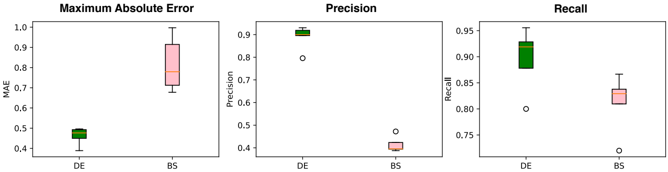
Dim. (x): 200, $\quad k = 0.3$



BS - denotes baseline sensing matrix (Dasarathy et al., "Sketching Sparse Matrices, Covariances, and Graphs via Tensor Products").

Results - preferential Covariance Recovery

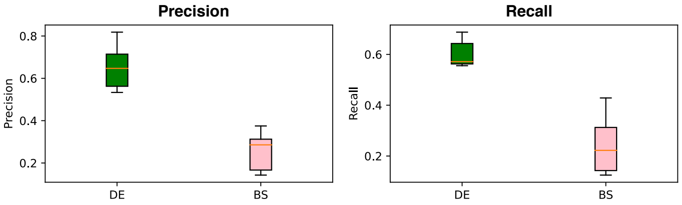
Dim. (x): 200, Dim. (x_H): 50, $k_H = 0.3$, Dim. (y): 60



BS - denotes baseline sensing matrix (Dasarathy et al., "Sketching Sparse Matrices, Covariances, and Graphs via Tensor Products").

Preferential Graph Recovery

Dim. (x): 200, Dim. (x_H): 50, $k_H = 0.3$, Dim. (y): 60



Graph Learning algorithm: Asish Ghoshal and Jean Honorio. “Learning Identifiable Gaussian Bayesian Networks in Polynomial Time and Sample Complexity.

Conclusion

- Using *Density Evolution* (DE) analysis the convergence of the message passing algorithm was reduced to a set of inequality constraints.

- The inequality constraints were then used to pose the design of A as a convex program.

- We also showed how additional constraints can be incorporated into the framework.

- Through numerical experiments we showed the efficacy of A-matrix designed using our framework.