CoLiDE: Concomitant Linear DAG Estimation

Gonzalo Mateos

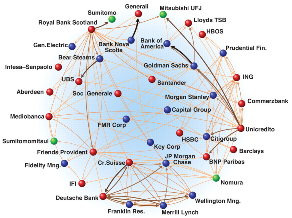
Dept. of ECE and Goergen Institute for Data Science
University of Rochester
gmateosb@ece.rochester.edu
http://hajim.rochester.edu/ece/sites/gmateos

Collaborators: S. Saman Saboksayr and Mariano Tepper
Acknowledgment: NSF Award ECCS-2231036, NY CoE in Data Science

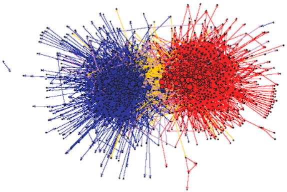
TU Delft
March 11, 2025

- Graphs are natural models for relational data that can help to learn in various timely applications

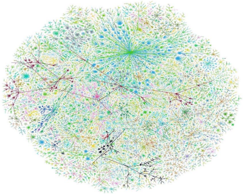
Economic Networks



Social and Information Networks

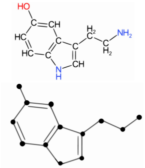


Internet

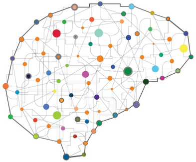


3D Meshes

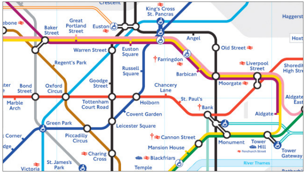
Molecules



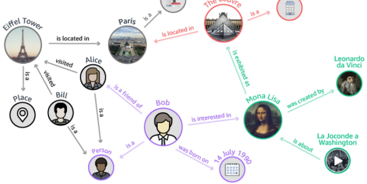
Brain Connectomes



Transportation Networks



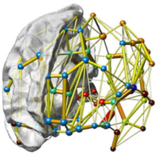
Knowledge Graphs



- Undirected topology inference from nodal observations [Kolaczyk'09]
  - Partial correlations and conditional dependence [Dempster'74]
  - Sparsity [Friedman et al'07] and consistency [Meinshausen-Buhlmann'06]
- Key in neuroscience and bioinformatics

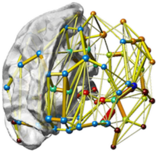
  ⇒ Functional network from fMRI signals [Sporns'10]

  ⇒ Gene-regulatory networks from microarray data [Mazumder-Hastie'12]



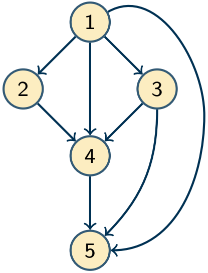
Learning graphs from data: Accounting for directionality and cycles

- Undirected topology inference from nodal observations [Kolaczyk'09]
  - Partial correlations and conditional dependence [Dempster'74]
  - Sparsity [Friedman et al'07] and consistency [Meinshausen-Buhlmann'06]



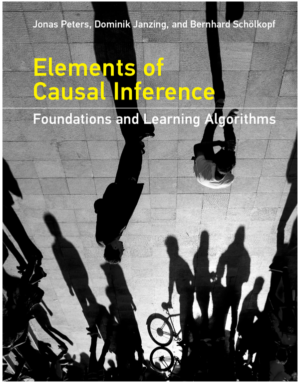
- Key in neuroscience and bioinformatics
  - ⇒ Functional network from fMRI signals [Sporns'10]
  - ⇒ Gene-regulatory networks from microarray data [Mazumder-Hastie'12]

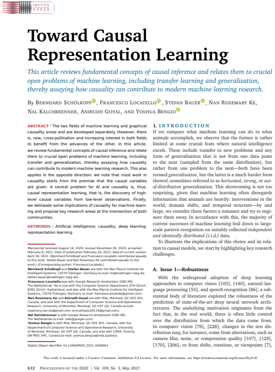
- **This work:** learn the structure of directed acyclic graphs (DAGs)

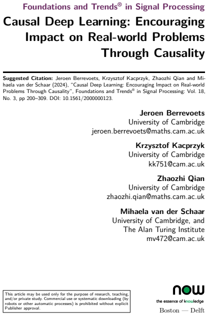


- DAGs have become prominent models in various ML applications
  - ⇒ Conditional independences among variables in Bayesian networks
  - ⇒ DAG edges may have causal interpretations
  - ⇒ Bio [Sachs et al'05], genetics [Zhang et al'13], finance [Sanford-Moosa'12]
- **Challenges:** directionality, acyclicity (combinatorial constraint), identifiability

- While our focus is on how optimization and statistical learning can aid inference of causal structures. . .



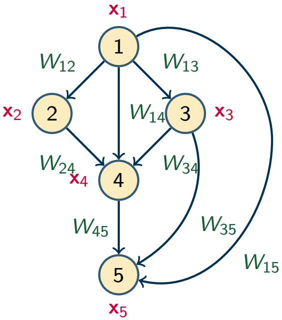




. . . causal reasoning can inform how we do ML (transferability, generalization, distribution shifts)

Roadmap

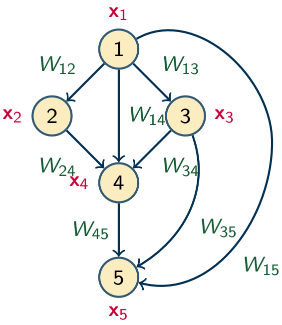
- DAG $\mathcal{G}(\mathcal{V},\mathcal{E},\mathbf{W}) \in \mathbb{D}$, vertices $\mathcal{V} = \{1,\ldots,d\}$, edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$
  - $\Rightarrow$ Adjacency matrix $\mathbf{W} = [\mathbf{w}_1,\ldots,\mathbf{w}_d] \in \mathbb{R}^{d\times d}$ of edge weights
  - $\Rightarrow$ Entry $W_{ij} \neq 0$ indicates a directed link from node i to j
- Random vector $\mathbf{x} = [x_1,\ldots,x_d] \in \mathbb{R}^d$, joint $p(\mathbf{x})$ Markov w.r.t. $\mathcal{G} \in \mathbb{D}$
  - $\Rightarrow$ DAG $\mathcal{G}$ encodes conditional independencies among variables in $\mathbf{x}$
  - $\Rightarrow$ Each x_i depends only on its parents $\mathrm{PA}_i = \{j \in \mathcal{V} : W_{ji} \neq 0\}$



- DAG $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W}) \in \mathbb{D}$, vertices $\mathcal{V} = \{1, \ldots, d\}$, edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$
  - $\Rightarrow$ Adjacency matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_d] \in \mathbb{R}^{d \times d}$ of edge weights
  - $\Rightarrow$ Entry $W_{ij} \neq 0$ indicates a directed link from node i to j
- Random vector $\mathbf{x} = [x_1, \ldots, x_d] \in \mathbb{R}^d$, joint $p(\mathbf{x})$ Markov w.r.t. $\mathcal{G} \in \mathbb{D}$
  - $\Rightarrow$ DAG $\mathcal{G}$ encodes conditional independencies among variables in $\mathbf{x}$
  - $\Rightarrow$ Each x_i depends only on its parents $\mathrm{PA}_i = \{j \in \mathcal{V} : W_{ji} \neq 0\}$
- Linear structural equation model (SEM) to generate $p(\mathbf{x})$ consists of

$$x_i = \mathbf{w}_i{}^\top \mathbf{x} + z_i, \quad \forall i \in \mathcal{V}$$

  - $\Rightarrow$ Mutually independent, exogenous noises $\mathbf{z} = [z_1, \ldots, z_d]^\top \in \mathbb{R}^d$
  - $\Rightarrow$ Ex: $x_4 = \mathbf{w}_4{}^\top \mathbf{x} + z_4 = W_{14}x_1 + W_{24}x_2 + W_{34}x_3 + z_4$

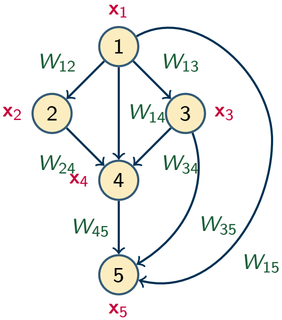


Linear structural equation (causal) models

- DAG $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W}) \in \mathbb{D}$, vertices $\mathcal{V} = \{1, \ldots, d\}$, edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$
  - $\Rightarrow$ Adjacency matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_d] \in \mathbb{R}^{d \times d}$ of edge weights
  - $\Rightarrow$ Entry $W_{ij} \neq 0$ indicates a directed link from node i to j
- Random vector $\mathbf{x} = [x_1, \ldots, x_d] \in \mathbb{R}^d$, joint $p(\mathbf{x})$ Markov w.r.t. $\mathcal{G} \in \mathbb{D}$
  - $\Rightarrow$ DAG $\mathcal{G}$ encodes conditional independencies among variables in $\mathbf{x}$
  - $\Rightarrow$ Each x_i depends only on its parents $\text{PA}_i = \{j \in \mathcal{V} : W_{ji} \neq 0\}$
- Linear structural equation model (SEM) to generate $p(\mathbf{x})$ consists of

$$x_i = \mathbf{w}_i{}^\top \mathbf{x} + z_i, \quad \forall i \in \mathcal{V}$$

  - $\Rightarrow$ Mutually independent, exogenous noises $\mathbf{z} = [z_1, \ldots, z_d]^\top \in \mathbb{R}^d$
  - $\Rightarrow$ Ex: $x_4 = \mathbf{w}_4{}^\top \mathbf{x} + z_4 = W_{14}x_1 + W_{24}x_2 + W_{34}x_3 + z_4$



- **Q:** Estimate $\mathbf{W}$ (learn DAG $\mathcal{G}$) using dataset $\mathbf{X} \in \mathbb{R}^{d \times n}$ with n i.i.d. samples from $p(\mathbf{x})$?

Given the data matrix $\mathbf{X}$ adhering to a linear SEM, learn the latent DAG $\mathcal{G} \in \mathbb{D}$ by estimating its adjacency matrix $\mathbf{W}$ as the solution to the score-minimization problem

$$\min_{\mathcal{G}(\mathbf{W})} \; \mathcal{S}(\mathcal{G}(\mathbf{W}); \mathbf{X}) \;\text{ subject to }\; \mathcal{G}(\mathbf{W}) \in \mathbb{D}$$

- Learning a DAG solely from observational data $\mathbf{X}$ is NP-hard [Chickering'96]
  - $\Rightarrow$ Combinatorial acyclicity constraint $\mathcal{G} \in \mathbb{D}$ nasty to enforce
  - $\Rightarrow$ Multiple DAGs may generate the same observational distribution $p(\mathbf{x})$

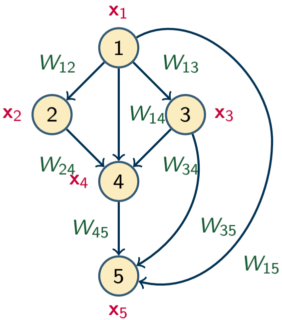
Given the data matrix $\mathbf{X}$ adhering to a linear SEM, learn the latent DAG $\mathcal{G} \in \mathbb{D}$ by estimating its adjacency matrix $\mathbf{W}$ as the solution to the score-minimization problem

$$\min_{\mathcal{G}(\mathbf{W})} \mathcal{S}(\mathcal{G}(\mathbf{W}); \mathbf{X}) \text{ subject to } \mathcal{G}(\mathbf{W}) \in \mathbb{D}$$

- Learning a DAG solely from observational data $\mathbf{X}$ is NP-hard [Chickering'96]
  - $\Rightarrow$ Combinatorial acyclicity constraint $\mathcal{G} \in \mathbb{D}$ nasty to enforce
  - $\Rightarrow$ Multiple DAGs may generate the same observational distribution $p(\mathbf{x})$
- **Discrete optimization:** combinatorial search methods
  - $\Rightarrow$ Penalized (BIC, MDL) likelihood and Bayesian scoring functions [Peters et al'17]
  - $\Rightarrow$ $|\mathbb{D}|$ grows superexponentially in d, methods face scalability issues
  - $\Rightarrow$ Approximate greedy search [Ramsey et al'17] and order-based methods [Park-Klabjan'17]

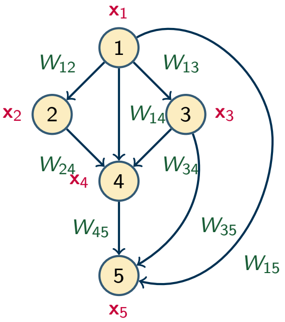
- If DAG's causal (partial) order were known $\Rightarrow$ **W** is upper-triangular

$$\mathbf{W} = \begin{bmatrix} 0 & W_{12} & W_{13} & W_{14} & W_{15} \\ 0 & 0 & 0 & W_{24} & 0 \\ 0 & 0 & 0 & W_{34} & W_{35} \\ 0 & 0 & 0 & 0 & W_{45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



- If DAG's causal (partial) order were known $\Rightarrow$ $\mathbf{W}$ is upper-triangular

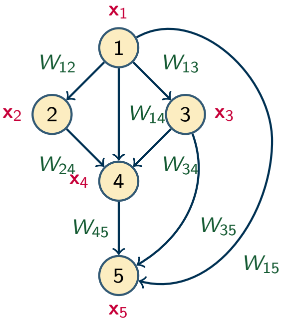
$$\mathbf{W} = \begin{bmatrix} 0 & W_{12} & W_{13} & W_{14} & W_{15} \\ 0 & 0 & 0 & W_{24} & 0 \\ 0 & 0 & 0 & W_{34} & W_{35} \\ 0 & 0 & 0 & 0 & W_{45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



- Exploit neat parameterization $\mathcal{G}(\mathbf{W}) \in \mathbb{D} \Leftrightarrow \mathbf{W} = \mathbf{\Pi}^\top \mathbf{U}\mathbf{\Pi}$
  - $\Rightarrow$ $\mathbf{U} \in \mathbb{R}^{d\times d}$ is an upper-triangular weight matrix
  - $\Rightarrow$ Permutation matrix $\mathbf{\Pi} \in \{0,1\}^{d\times d}$ encodes the causal ordering

- If DAG's causal (partial) order were known $\Rightarrow$ $\mathbf{W}$ is upper-triangular

$$\mathbf{W} = \begin{bmatrix} 0 & W_{12} & W_{13} & W_{14} & W_{15} \\ 0 & 0 & 0 & W_{24} & 0 \\ 0 & 0 & 0 & W_{34} & W_{35} \\ 0 & 0 & 0 & 0 & W_{45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



- Exploit neat parameterization $\mathcal{G}(\mathbf{W}) \in \mathbb{D} \Leftrightarrow \mathbf{W} = \mathbf{\Pi}^\top \mathbf{U} \mathbf{\Pi}$
  - $\Rightarrow$ $\mathbf{U} \in \mathbb{R}^{d \times d}$ is an upper-triangular weight matrix
  - $\Rightarrow$ Permutation matrix $\mathbf{\Pi} \in \{0,1\}^{d \times d}$ encodes the causal ordering
- Search over exact DAGs in an end-to-end differentiable fashion
  - $\Rightarrow$ Learn permutations with Gumbel-Sinkhorn [Cundy et al'21] or SoftSort [Charpentier et al'22]
  - $\Rightarrow$ Bi-level optimization, topological order swaps at the outer level [Deng et al'23]
- Accurately recovering the causal ordering is challenging, especially when data are limited

Breakthrough: From discrete to continuous optimization

- Acyclicity characterization using nonconvex, smooth functions $\mathcal{H}(\mathbf{W}) : \mathbb{R}^{d \times d} \mapsto \mathbb{R}$

  $\Rightarrow$ Zero level set corresponds to DAGs: $\mathcal{H}(\mathbf{W}) = 0 \iff \mathcal{G}(\mathbf{W}) \in \mathbb{D}$

- **Upshot:** from combinatorial search to nonconvex (smooth) continuous optimization

$$\min_{\mathcal{G}(\mathbf{W})} \ \mathcal{S}(\mathcal{G}(\mathbf{W}); \mathbf{X}) \ \text{ subject to } \ \mathcal{G}(\mathbf{W}) \in \mathbb{D} \quad \iff \quad \min_{\mathbf{W}} \ \mathcal{S}(\mathbf{W}; \mathbf{X}) \ \text{ subject to } \ \mathcal{H}(\mathbf{W}) = 0$$

- **Q:** What are these acyclicity functions $\mathcal{H}$? What about the DAG scoring functions $\mathcal{S}$?

X. Zheng *et al*, "DAGs with NOTEARS: Continuous optimization for structure learning," *NeurIPS*, 2018

Acyclicity functions

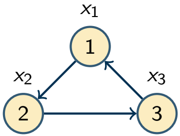
- Pioneering NOTEARS formulation proposed $\mathcal{H}_{\text{expm}}(\mathbf{W}) = \operatorname{Tr}\left(e^{\mathbf{W} \circ \mathbf{W}}\right) - d$ [Zheng et al'18]
  - $\Rightarrow$ Idea: diagonal entries of powers of $\mathbf{W} \circ \mathbf{W}$ encode information about cycles in $\mathcal{G}$

Acyclicity functions

- Pioneering NOTEARS formulation proposed $\mathcal{H}_{\text{expm}}(\mathbf{W}) = \text{Tr}\left(e^{\mathbf{W}\circ\mathbf{W}}\right) - d$ [Zheng et al'18]

  $\Rightarrow$ Idea: diagonal entries of powers of $\mathbf{W} \circ \mathbf{W}$ encode information about cycles in $\mathcal{G}$

$$e^{\mathbf{W}} = \sum_{k=0}^{\infty} \frac{(\mathbf{W})^k}{k!} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}}_{\text{self-loops}} + \frac{1}{2}\underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{\text{cycles of size 2}} + \frac{1}{6}\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{cycles of size 3}} + \cdots$$

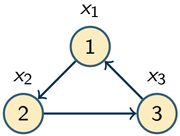


Acyclicity functions

- Pioneering NOTEARS formulation proposed $\mathcal{H}_{\text{expm}}(\mathbf{W}) = \operatorname{Tr}\left(e^{\mathbf{W}\circ\mathbf{W}}\right) - d$ [Zheng et al'18]

  $\Rightarrow$ Idea: diagonal entries of powers of $\mathbf{W}\circ\mathbf{W}$ encode information about cycles in $\mathcal{G}$

$$e^{\mathbf{W}} = \sum_{k=0}^{\infty} \frac{(\mathbf{W})^k}{k!} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}}_{\text{self-loops}} + \frac{1}{2}\underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{\text{cycles of size 2}} + \frac{1}{6}\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{cycles of size 3}} + \cdots$$



- To speed up computation, [Yu et al'19] advocates $\mathcal{H}_{\text{poly}}(\mathbf{W}) = \operatorname{Tr}\left((\mathbf{I} + \frac{1}{d}\mathbf{W}\circ\mathbf{W})^d\right) - d$

  $\Rightarrow$ Cayley-Hamilton: both $\mathcal{H}_{\text{expm}}$ and $\mathcal{H}_{\text{poly}}$ subsumed by $\operatorname{Tr}\left(\sum_{k=1}^{d} c_k(\mathbf{W}\circ\mathbf{W})^d\right) - d$

- Log-determinant function $\mathcal{H}_{\text{ldet}}(\mathbf{W}; s) = d\log(s) - \log(\det(s\mathbf{I} - \mathbf{W}\circ\mathbf{W})), \quad s > \rho(\mathbf{W}\circ\mathbf{W})$

  $\Rightarrow$ State-of-the-art with several attractive features at the heart of DAGMA

K. Bello *et al*, "DAGMA: Learning DAGs via M-matrices and a log-determinant acyclicity characterization," *NeurIPS*, 2022

- Ordinary LS loss augmented with an ℓ_1-norm regularizer

$$\mathcal{S}(\mathbf{W};\mathbf{X}) = \frac{1}{2n}\|\mathbf{X} - \mathbf{W}^\top\mathbf{X}\|_F^2 + \lambda\|\mathbf{W}\|_1$$

$\Rightarrow$ $\lambda \geq 0$ is a tuning parameter that controls edge sparsity

$\Rightarrow$ Computational efficiency, robustness, and even consistency [Loh-Buhlmann'15]

Regression-based DAG scoring functions

- Ordinary LS loss augmented with an ℓ_1-norm regularizer

$$\mathcal{S}(\mathbf{W};\mathbf{X}) = \frac{1}{2n}\|\mathbf{X} - \mathbf{W}^\top\mathbf{X}\|_F^2 + \lambda\|\mathbf{W}\|_1$$

$\Rightarrow$ $\lambda \geq 0$ is a tuning parameter that controls edge sparsity

$\Rightarrow$ Computational efficiency, robustness, and even consistency [Loh-Buhlmann'15]

- Multi-task variant of lasso [Tibshirani'96], when response and design matrices coincide

$\Rightarrow$ Optimal rates for $\lambda \asymp \sigma\sqrt{\log d/n}$ [Li et al'20]. But σ^2 is rarely known

- Key limitations we identify:

$\Rightarrow$ Requires carefully retuning λ when unknown σ^2 changes across problems

$\Rightarrow$ Implicitly relies on limiting homoscedasticity assumptions

Contributions

- New convex score function for sparsity-aware learning of linear DAGs
  - $\Rightarrow$ Incorporate concomitant estimation of scale parameters. Learn $\mathbf{W}$ and σ jointly
  - $\Rightarrow$ CoLiDE (**C**oncomitant **Li**near **D**AG **E**stimation) decouples λ and σ. No recalibration
  - $\Rightarrow$ Unlike ordinary LS, it accommodates heteroscedastic exogenous noise profiles

- New convex score function for sparsity-aware learning of linear DAGs
  - ⇒ Incorporate concomitant estimation of scale parameters. Learn **W** and σ jointly
  - ⇒ CoLiDE (**C**oncomitant **Li**near **D**AG **E**stimation) decouples λ and σ. No recalibration
  - ⇒ Unlike ordinary LS, it accommodates heteroscedastic exogenous noise profiles

- CoLiDE outperforms state-of-the-art methods across graph ensembles and noise distributions
  - ⇒ Especially when DAGs are larger and the noise level profile is heterogeneous
  - ⇒ Enhanced stability via reduced standard errors across domain-specific metrics

Table: DAG recovery results for 200-node ER4 graphs under homoscedastic Gaussian noise

	Noise variance = 1.0				Noise variance = 5.0			
	GOLEM	DAGMA	CoLiDE-NV	CoLiDE-EV	GOLEM	DAGMA	CoLiDE-NV	CoLiDE-EV
SHD	468.6±144.0	100.1±41.8	111.9±29	**87.3±33.7**	336.6±233.0	194.4±36.2	157±44.2	**105.6±51.5**
SID	22260±3951	4389±1204	5333±872	**4010±1169**	14472±9203	6582±1227	6067±1088	**4444±1586**
SHD-C	473.6±144.8	101.2±41.0	113.6±29.2	**88.1±33.8**	341.0±234.9	199.9±36.1	161.0±43.5	**107.1±51.6**
FDR	0.28±0.10	0.07±0.03	0.08±0.02	**0.06±0.02**	0.21±0.13	0.15±0.02	0.12±0.03	**0.08±0.04**
TPR	0.66±0.09	0.94±0.01	0.93±0.01	**0.95±0.01**	0.76±0.18	0.92±0.01	0.93±0.01	**0.95±0.01**

S. S. Saboksayr *et al*, "CoLiDE: Concomitant linear DAG estimation," *ICLR*, 2024

- **Homoscedastic setting:** $z_1, \ldots, z_d$ in the linear SEM have identical variance σ^2

- Inspired by the smoothed concomitant lasso [Ndiaye et al'17], we propose CoLiDE-EV

$$\min_{\mathbf{W}, \sigma \geq \sigma_0} \underbrace{\left[\frac{1}{2n\sigma} \|\mathbf{X} - \mathbf{W}^\top \mathbf{X}\|_F^2 + \frac{d\sigma}{2} + \lambda \|\mathbf{W}\|_1 \right]}_{:=\mathcal{S}(\mathbf{W}, \sigma; \mathbf{X})} \quad \text{subject to} \quad \mathcal{H}(\mathbf{W}) = 0$$

$\Rightarrow$ Can be traced back to the robust linear regression work of [Huber'81]

$\Rightarrow$ Constraint $\sigma \geq \sigma_0$ safeguards against ill-posed scenarios. Set $\sigma_0 = \frac{\|\mathbf{X}\|_F}{\sqrt{dn}} \times 10^{-2}$

Concomitant Linear DAG Estimation

- **Homoscedastic setting:** $z_1, \ldots, z_d$ in the linear SEM have identical variance σ^2

- Inspired by the smoothed concomitant lasso [Ndiaye et al'17], we propose CoLiDE-EV

$$\min_{\mathbf{W}, \sigma \geq \sigma_0} \underbrace{\left[\frac{1}{2n\sigma}\|\mathbf{X} - \mathbf{W}^\top \mathbf{X}\|_F^2 + \frac{d\sigma}{2} + \lambda\|\mathbf{W}\|_1\right]}_{:=\mathcal{S}(\mathbf{W}, \sigma; \mathbf{X})} \quad \text{subject to} \quad \mathcal{H}(\mathbf{W}) = 0$$

$\Rightarrow$ Can be traced back to the robust linear regression work of [Huber'81]

$\Rightarrow$ Constraint $\sigma \geq \sigma_0$ safeguards against ill-posed scenarios. Set $\sigma_0 = \frac{\|\mathbf{X}\|_F}{\sqrt{dn}} \times 10^{-2}$

- Here λ decouples from σ as minimax optimality now requires $\lambda \asymp \sqrt{\log d/n}$

$\Rightarrow$ Score $\mathcal{S}(\mathbf{W}, \sigma; \mathbf{X})$ is jointly convex w.r.t. $\mathbf{W}$ and σ. Overall nonconvex due to $\mathcal{H}(\mathbf{W})$

$\Rightarrow$ Included $(d\sigma)/2$ so that $\hat{\sigma}^2$ is consistent under Gaussianity

Optimization considerations

- Solve a sequence of unconstrained problems where $\mathcal{H}$ is viewed as a regularizer [Bello et al'22]

  $\Rightarrow$ More effective in practice compared to an augmented Lagrangian method

- Given a decreasing sequence of values $\mu_k \to 0$, at step k of CoLiDE-EV solve

$$\text{(P1)} \quad \min_{\mathbf{W}, \sigma \geq \sigma_0} \mu_k \left[\frac{1}{2n\sigma} \|\mathbf{X} - \mathbf{W}^\top \mathbf{X}\|_F^2 + \frac{d\sigma}{2} + \lambda \|\mathbf{W}\|_1 \right] + \mathcal{H}_{\text{ldet}}(\mathbf{W}, s_k)$$

  $\Rightarrow$ Hyperparameters $\mu_k \geq 0$ and $s_k > 0$ must be prescribed prior to implementation

  $\Rightarrow$ Decreasing the value of μ_k enhances the influence of the acyclicity function

  $\Rightarrow$ Like central path approach of barrier methods. Limit $\mu_k \to 0$ is guaranteed to yield a DAG

Inexact block coordinate descent

- CoLiDE-EV jointly estimates noise level σ and adjacency matrix $\mathbf{W}$ for each μ_k

  $\Rightarrow$ Rely on inexact block coordinate descent (BCD) iterations

- **Step 1:** Fix σ to its most up-to-date value and minimize $\mathcal{S}(\mathbf{W}, \sigma; \mathbf{X})$ inexactly w.r.t. $\mathbf{W}$

  $\Rightarrow$ Run one iteration of the ADAM optimizer

- **Step 2:** Update σ in closed form given the latest $\mathbf{W}$

$$\hat{\sigma} = \max\left(\frac{1}{\sqrt{nd}}\|\mathbf{X} - \mathbf{W}^\top\mathbf{X}\|_F, \sigma_0\right) = \max\left(\sqrt{\mathrm{Tr}\left((\mathbf{I} - \mathbf{W})^\top \mathrm{cov}(\mathbf{X})(\mathbf{I} - \mathbf{W})\right)/d}, \sigma_0\right)$$

$\Rightarrow$ Precomputed sample covariance matrix $\mathrm{cov}(\mathbf{X}) := \frac{1}{n}\mathbf{X}\mathbf{X}^\top$

Inexact block coordinate descent

- CoLiDE-EV jointly estimates noise level σ and adjacency matrix $\mathbf{W}$ for each μ_k

  $\Rightarrow$ Rely on inexact block coordinate descent (BCD) iterations

- **Step 1:** Fix σ to its most up-to-date value and minimize $\mathcal{S}(\mathbf{W}, \sigma; \mathbf{X})$ inexactly w.r.t. $\mathbf{W}$

  $\Rightarrow$ Run one iteration of the ADAM optimizer

- **Step 2:** Update σ in closed form given the latest $\mathbf{W}$

$$\hat{\sigma} = \max\left(\frac{1}{\sqrt{nd}}\|\mathbf{X} - \mathbf{W}^\top \mathbf{X}\|_F, \sigma_0\right) = \max\left(\sqrt{\operatorname{Tr}\left((\mathbf{I} - \mathbf{W})^\top \operatorname{cov}(\mathbf{X})(\mathbf{I} - \mathbf{W})\right)/d}, \sigma_0\right)$$

  $\Rightarrow$ Precomputed sample covariance matrix $\operatorname{cov}(\mathbf{X}) := \frac{1}{n}\mathbf{X}\mathbf{X}^\top$

- Provably convergent block successive convex approximation (BSCA) algorithm also effective

S. S. Saboksayr *et al*, "Block successive convex approximation for concomitant linear DAG estimation," *SAM Workshop*, 2024

- **Heteroscedastic setting:** noise variables have non-equal variances (NV) $\sigma_1^2, \ldots, \sigma_d^2$

- Mimicking the optimization approach for the EV case, we propose CoLiDE-NV

$$\text{(P2)} \quad \min_{\mathbf{W}, \boldsymbol{\Sigma} \geq \boldsymbol{\Sigma}_0} \mu_k \left[\frac{1}{2n} \operatorname{Tr}\left((\mathbf{X} - \mathbf{W}^\top \mathbf{X})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{X} - \mathbf{W}^\top \mathbf{X}) \right) + \frac{1}{2} \operatorname{Tr}(\boldsymbol{\Sigma}) + \lambda \|\mathbf{W}\|_1 \right] + \mathcal{H}_{\text{ldet}}(\mathbf{W}, s_k)$$

$\Rightarrow$ $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_d)$ is a diagonal matrix of exogenous noise standard deviations

$\Rightarrow$ Special case $\boldsymbol{\Sigma} = \sigma \mathbf{I}$ yields CoLiDE-EV score function

- **Heteroscedastic setting:** noise variables have non-equal variances (NV) $\sigma_1^2, \ldots, \sigma_d^2$

- Mimicking the optimization approach for the EV case, we propose CoLiDE-NV

$$(\text{P2}) \quad \min_{\mathbf{W}, \boldsymbol{\Sigma} \geq \boldsymbol{\Sigma}_0} \mu_k \left[\frac{1}{2n} \operatorname{Tr}\left((\mathbf{X} - \mathbf{W}^\top \mathbf{X})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{X} - \mathbf{W}^\top \mathbf{X}) \right) + \frac{1}{2} \operatorname{Tr}(\boldsymbol{\Sigma}) + \lambda \|\mathbf{W}\|_1 \right] + \mathcal{H}_{\text{ldet}}(\mathbf{W}, s_k)$$

$\Rightarrow$ $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_d)$ is a diagonal matrix of exogenous noise standard deviations

$\Rightarrow$ Special case $\boldsymbol{\Sigma} = \sigma \mathbf{I}$ yields CoLiDE-EV score function

- Closed-form solution for $\boldsymbol{\Sigma}$ given $\mathbf{W}$

$$\hat{\boldsymbol{\Sigma}} = \max\left(\sqrt{\operatorname{diag}\left((\mathbf{I} - \mathbf{W})^\top \operatorname{cov}(\mathbf{X}) (\mathbf{I} - \mathbf{W}) \right)}, \boldsymbol{\Sigma}_0 \right) \quad \text{or} \quad \hat{\sigma}_i = \max\left(\frac{1}{\sqrt{n}} \|\mathbf{x}_i - \mathbf{w}_i{}^\top \mathbf{X}\|_2, \sigma_0 \right)$$

- CoLiDE's per iteration cost is $\mathcal{O}(d^3)$, on par with state-of-the-art DAG learning methods

Summary and discussion points

Algorithm 1: CoLiDE optimization

In: data $\mathbf{X}$ and hyperparameters λ and $H = \{(\mu_k, s_k, T_k)\}_{k=1}^{K}$.
Out: DAG $\mathbf{W}$ and the noise estimate σ (EV) or $\mathbf{\Sigma}$ (NV).
Compute lower-bounds σ_0 or $\mathbf{\Sigma}_0$.
Initialize $\mathbf{W} = \mathbf{0}$, $\sigma = \sigma_0 \times 10^2$ or $\mathbf{\Sigma} = \mathbf{\Sigma}_0 \times 10^2$.
foreach $(\mu_k, s_k, T_k) \in H$ **do**
  for $t = 1, \ldots, T_k$ **do**
    Apply CoLiDE-EV or NV updates using μ_k and s_k.

Function *CoLiDE-EV update*:
  Update $\mathbf{W}$ with one iteration of a first-order method for (P1)
  Compute $\hat{\sigma}$ in closed form

Function *CoLiDE-NV update*:
  Update $\mathbf{W}$ with one iteration of a first-order method for (P2)
  Compute $\hat{\mathbf{\Sigma}}$ in closed form

- Decomposable: unlike Gaussian profile log-likelihood in GOLEM [Ng et al'20]

$$\mathcal{S}(\mathbf{W};\mathbf{X}) = -\frac{1}{2}\sum_{i=1}^{d}\log\left(\left\|\mathbf{x}_i - \mathbf{w}_i{}^{\top}\mathbf{X}\right\|_2^2\right) + \log\left(|\det(\mathbf{I}-\mathbf{W})|\right) + \lambda\|\mathbf{W}\|_1$$

- Guarantees: consider general (non-identifiable) linear Gaussian SEMs
  $\Rightarrow$ As $n \to \infty$ CoLiDE-NV outputs a DAG quasi-equivalent to the ground-truth graph

- Flexible: other convex losses beyond LS, other $\mathcal{H}$, nonlinear SEMs, impact to order-based methods

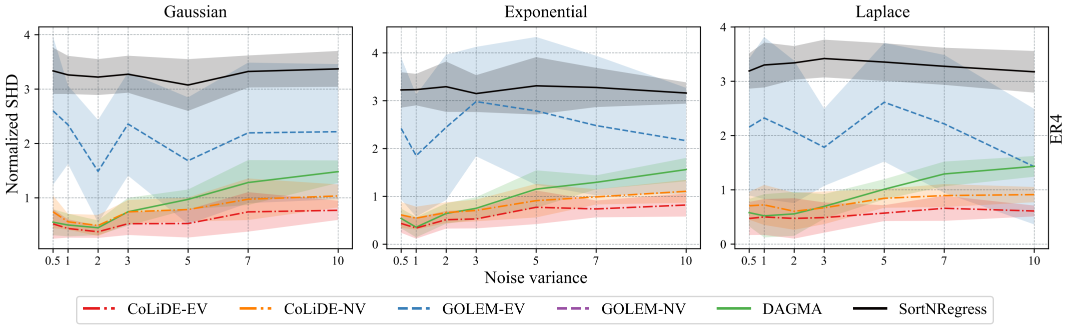
I. Ng *et al*, "On the role of sparsity and DAG constraints for learning linear DAGs," *NeurIPS*, 2020

Experimental performance evaluation

- Comprehensive evaluation to assess the effectiveness of the CoLiDE framework
  - $\Rightarrow$ Validate DAG recovery performance in synthetic EV and NV settings
  - $\Rightarrow$ Examine noise estimation performance
  - $\Rightarrow$ Evaluate DAG recovery performance on real-world datasets
  - $\Rightarrow$ Compare with other methods such as DAGMA, GOLEM, SortNRegress, GES, ...
- Tests across graph types (edge weights, average degree), noise distributions, values of d, n, σ
- Reproducibility: code to generate all figures at `https://github.com/SAMiatto/colide`

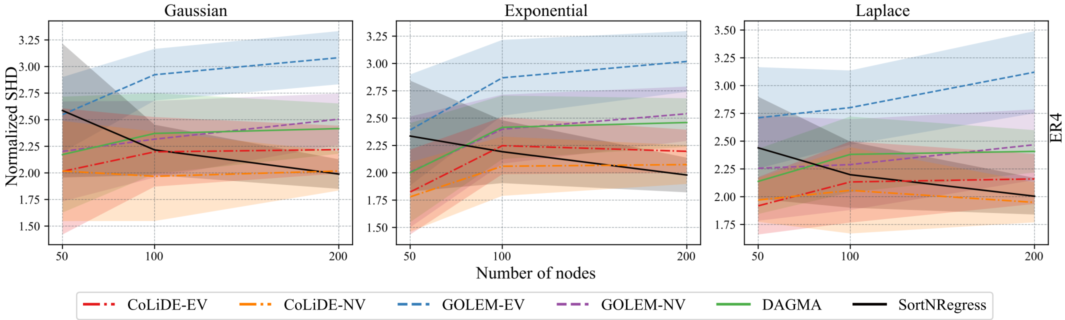
Experiments: Homoscedastic setting

- Investigate the impact of noise level σ^2 on DAG recovery performance
  - Graphs: 200-node ER4 graphs, W_{ij} drawn uniformly from $[-2, -0.5] \cup [0.5, 2]$
  - Data: $n = 1000$ samples via linear SEM, diverse noise distributions
  - Metric: SHD counts number of edge corrections required to recover true graph from estimate



- CoLiDE-EV outperforming DAGMA clearly demonstrates the gains come from $\mathcal{S}(\mathbf{W}, \sigma; \mathbf{X})$

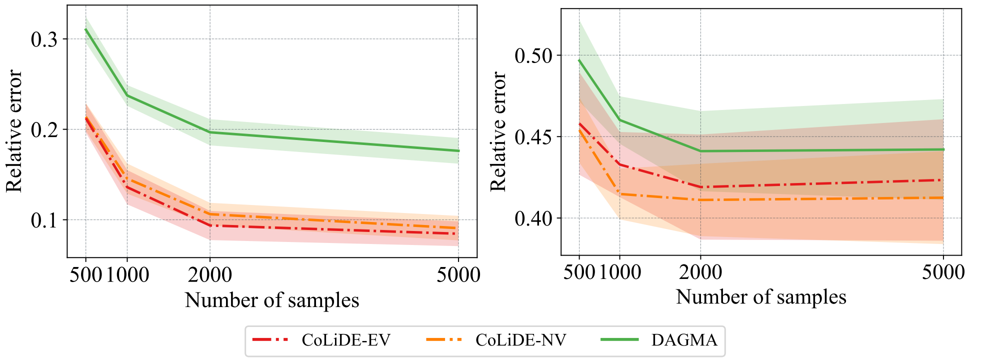
- Heteroscedastic scenario poses further challenges $\Rightarrow$ Non-indentifiable from observational data
  - Noise variance of each node σ_i^2 is uniformly drawn from $[0.5, 10]$
  - Graphs: ER4 graphs varying d; W_{ij} drawn from $[-1, -0.25] \cup [0.25, 1]$ (lower SNR)
  - Data: $n = 1000$ samples via linear SEM, diverse noise distributions



- CoLiDE-NV yields lower deviations than DAGMA and GOLEM, underscoring its robustness

Experiments: Noise estimation

- Method's ability to estimate noise variance $\Rightarrow$ Proficiency in recovering accurate edge weights
  - DAGMA does not explicitly estimate noise level, we use $\hat{\sigma}_i^2 = \frac{1}{n}\|\mathbf{x}_i - \hat{\mathbf{w}}_i^\top \mathbf{X}\|_2^2$
  - Graphs: 200-node ER4 graphs, W_{ij} drawn uniformly from $[-2, -0.5] \cup [0.5, 2]$
  - Signals: Linear SEM with Gaussian noise; vary n for EV (left) and NV (right) scenarios



- CoLiDE-NV provides lower error even when using half as many samples as DAGMA

Experiments: Cell-signaling data

- Tested CoLiDE on the Sachs dataset [Sachs et al'05]
  - ⇒ Cytometric measurements from human immune system
  - ⇒ Comprises $d = 11$ proteins, 17 edges, and $n = 853$ samples
  - ⇒ Associated DAG is obtained through experimental methods
- CoLiDE-NV attains lowest SHD to date for this problem

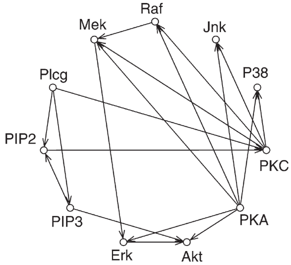


Table: DAG recovery performance on the Sachs dataset

	GOLEM-EV	GOLEM-NV	DAGMA	SortNRegress	DAGuerreotype	GES	CoLiDE-EV	CoLiDE-NV
SHD	22	15	16	13	14	13	13	**12**
SID	49	58	52	47	50	56	47	**46**
SHD-C	19	**11**	15	13	12	**11**	13	14
FDR	0.83	0.66	**0.5**	0.61	0.57	**0.5**	0.54	0.53
TPR	0.11	0.11	0.05	0.29	0.17	0.23	0.29	**0.35**

K. Sachs *et al*, "Causal protein-signaling networks derived from multiparameter single-cell data," *Science*, 2005

Concluding remarks and the road ahead

- DAGs as general descriptors of causal and (in)dependence relationships
  - $\Rightarrow$ Understanding the enforcement of acyclicty for DAG learning from observational data
  - $\Rightarrow$ Emphasizing the significance of the score function in continuous-optimization methods
- Proposed framework: CoLiDE (**C**oncomitant **Li**near **D**AG **E**stimation)
  - $\Rightarrow$ Jointly estimates the DAG structure and noise level
  - $\Rightarrow$ Adaptivity to changes in noise levels, requires less fine-tuning
  - $\Rightarrow$ Applicable to challenging heteroscedastic scenarios
  - $\Rightarrow$ Surpassing state-of-the-art in DAG recovery performance

Concluding remarks and the road ahead

- DAGs as general descriptors of causal and (in)dependence relationships
  - $\Rightarrow$ Understanding the enforcement of acyclicty for DAG learning from observational data
  - $\Rightarrow$ Emphasizing the significance of the score function in continuous-optimization methods
- Proposed framework: CoLiDE (**C**oncomitant **Li**near **D**AG **E**stimation)
  - $\Rightarrow$ Jointly estimates the DAG structure and noise level
  - $\Rightarrow$ Adaptivity to changes in noise levels, requires less fine-tuning
  - $\Rightarrow$ Applicable to challenging heteroscedastic scenarios
  - $\Rightarrow$ Surpassing state-of-the-art in DAG recovery performance
- Ongoing and future work:
  - $\Rightarrow$ Non-linear SEMs via neural networks or kernels
  - $\Rightarrow$ Online DAG learning from streaming signals, time-series data via SVAR models