DAG Learning on the Permutahedron

Valentina Zantedeschi, Luca Franceschi, Jean Kaddour, Matt J. Kusner, Vlad Niculae

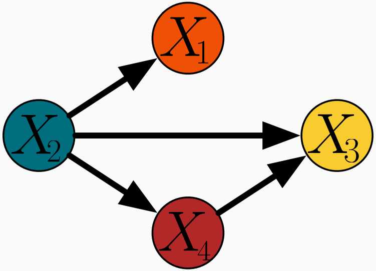
To appear at ICLR 2023, Rwanda

January 28, 2023

Problem Statement

Bayesian Network

Directed Acyclic Graph (DAG)



Markov Factorization of joint distribution

$$
\begin{aligned}
p(X_1, X_2, X_3, X_4) &= \prod_{i=1}^{4} p(X_i \mid \mathrm{pa}(X_i)) \\
&= p(X_1 \mid X_2) p(X_2) p(X_3 \mid X_2, X_4) p(X_4 \mid X_2)
\end{aligned}
$$

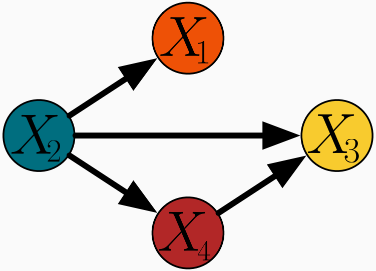
a DAG represents

- parent-child dependences
- conditional independences

Problem Statement

Bayesian Network

Directed Acyclic Graph (DAG)



Markov Factorization of joint distribution

$$p(X_1, X_2, X_3, X_4) = \prod_{i=1}^{4} p(X_i \mid \mathrm{pa}(X_i))$$
$$= p(X_1 \mid X_2)p(X_2)p(X_3 \mid X_2, X_4)p(X_4 \mid X_2)$$

a DAG represents

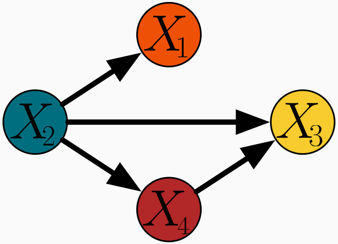
- parent-child dependences
- conditional independences

How can we learn DAG from data generated from joint distribution?

Applications

Bayesian Network

Directed Acyclic Graph (DAG)

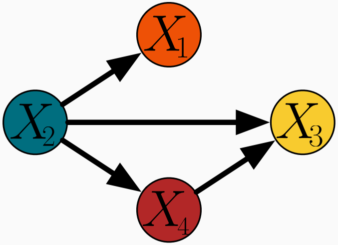


- Causal Discovery edge := cause-effect link
  →help reason about interventions:
  What happens if we increase interest rates?

Applications

Bayesian Network

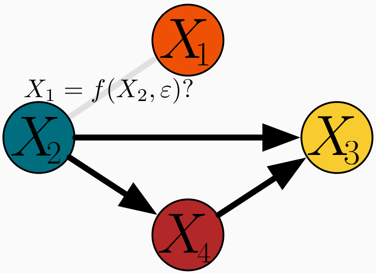
Directed Acyclic Graph (DAG)



- Causal Discovery edge := cause-effect link
  →help reason about interventions:
  What happens if we increase interest rates?

- Interpretability sparsest set of dependences
  →help interpret model predictions:
  Which features were decisive?

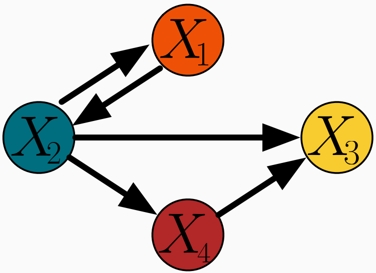
Challenges



Estimation:

model speficication	assumptions on edge functions $p(X_i \mid \mathrm{pa}(X_i))$
identifiability	non-uniqueness (identify up to Markov Equivalence Class [PJS17])
approximation	finite sample from joint distribution

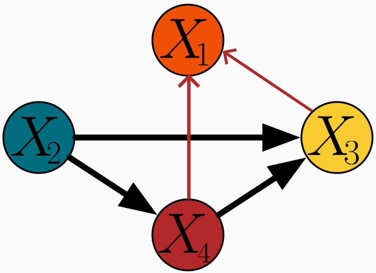
Challenges



Estimation:

model speficication	assumptions on edge functions $p(X_i \mid \mathrm{pa}(X_i))$
identifiability	non-uniqueness (identify up to Markov Equivalence Class [PJS17])
approximation	finite sample from joint distribution

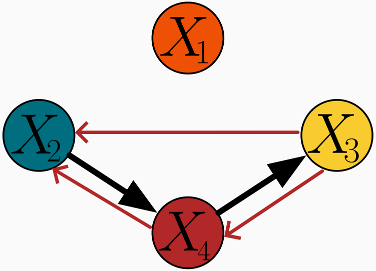
Challenges



Estimation:

model speficication	assumptions on edge functions $p(X_i \mid \mathrm{pa}(X_i))$
identifiability	non-uniqueness (identify up to Markov Equivalence Class [PJS17])
approximation	finite sample from joint distribution

Challenges



Estimation:

model speficication	assumptions on edge functions $p(X_i \mid \mathrm{pa}(X_i))$
identifiability	non-uniqueness (identify up to Markov Equivalence Class [PJS17])
approximation	finite sample from joint distribution

Computation:

NP-hard	because of acyclicity constraint [Chi95]

Continuous Characterization - NoTears [ZARX18]

d variables, binary adjacency matrix $B \in \{0,1\}^{d^2}$

Similar characterizations: [YCGY19, BAR22].

Continuous Characterization - NoTears [ZARX18]

d variables, binary adjacency matrix $B \in \{0,1\}^{d^2}$

hop-1 if $\exists\ i \mid B_{ii} = 1$ then self-loop exists
$trace(B^1)$ counts the number of such cycles

Similar characterizations: [YCGY19, BAR22].

Continuous Characterization - NoTears [ZARX18]

d variables, binary adjacency matrix $B \in \{0,1\}^{d^2}$

hop-1 if $\exists\ i \ \mid\ B_{ii} = 1$ then self-loop exists
$trace(B^1)$ counts the number of such cycles

hop-2 if $\exists\ i, j \ \mid\ B_{ij}B_{ji} = 1$ then length-2 cycle exists
$trace(B^2)$ counts the number of such cycles

Similar characterizations: [YCGY19, BAR22].

Continuous Characterization - NoTears [ZARX18]

d variables, binary adjacency matrix $B \in \{0,1\}^{d^2}$

hop-1 if $\exists\ i \mid B_{ii} = 1$ then self-loop exists
$trace(B^1)$ counts the number of such cycles

hop-2 if $\exists\ i,j \mid B_{ij}B_{ji} = 1$ then length-2 cycle exists
$trace(B^2)$ counts the number of such cycles

hop-k if $\exists\ \{i_1, i_2, \ldots, i_k\} \mid \prod_j B_{i_j i_{j+1}} = 1$ then length-k cycle exists
$trace(B^k)$ counts the number of such cycles

Similar characterizations: [YCGY19, BAR22].

Continuous Characterization - NoTears [ZARX18]

d variables, binary adjacency matrix $B \in \{0,1\}^{d^2}$

$$\sum_{k=1}^{\infty} \frac{trace(B^k)}{k!} = trace(\exp(B)) - trace(B^0) = trace(\exp(B)) - d$$

Similar characterizations: [YCGY19, BAR22].

Continuous Characterization - NoTears [ZARX18]

d variables, binary adjacency matrix $B \in \{0,1\}^{d^2}$

Constrained Optimization Problem

Data $X \in \mathbb{R}^{nd}$ and weighted adjacency matrix $W \in \mathbb{R}^{d^2}$

$$\begin{aligned} &\arg\min_W \mathcal{L}(X, W) \\ s.t. \quad &trace(\exp(W \circ W)) - d = 0 \end{aligned}$$

Solve by e.g. Augmented Lagrangian. Then, threshold W to get B.

Similar characterizations: [YCGY19, BAR22].

Continuous Characterization - NoTears [ZARX18]

d variables, binary adjacency matrix $B \in \{0,1\}^{d^2}$

Advantages

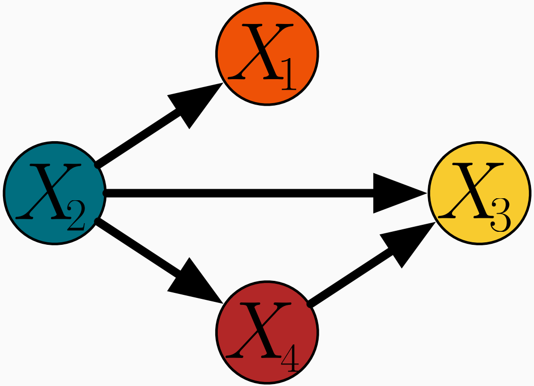
1. genericity: nonparametric (neural) edge functions (e.g. [ZDA+20, LBDL20])
2. scalability: data size, number of parameters
   cubic complexity in number of variables (up to ~ 500)

Downsides

1. invalidity: not a DAG at training and at convergence
2. non-modularity: require differentiable operations
3. scale-sensitive: tend to order variables (root to sink) by marginal variance [RSW21]

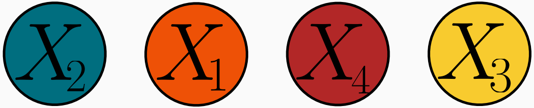
Similar characterizations: [YCGY19, BAR22].

Order-based Approach [FK03]



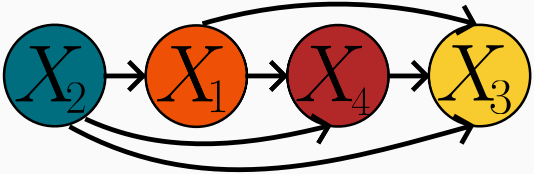
Order-based Approach [FK03]

1 Learn total ordering of variables



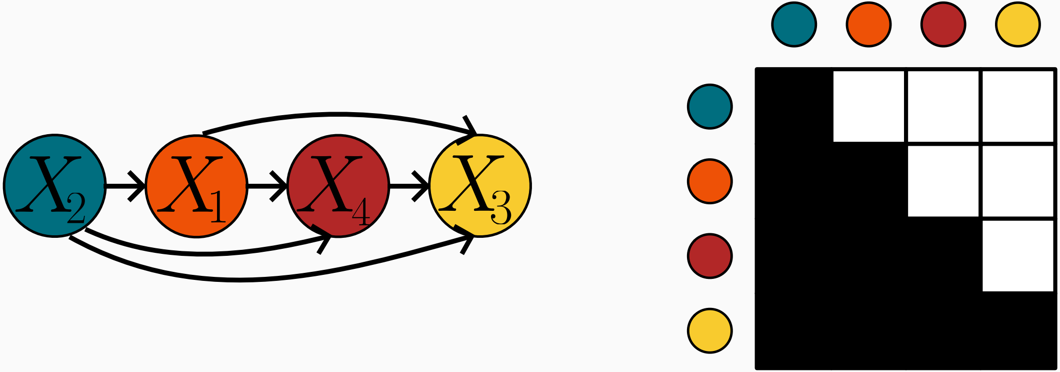
Order-based Approach [FK03]

2 Get corresponding complete DAG

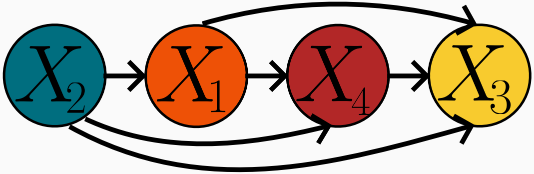


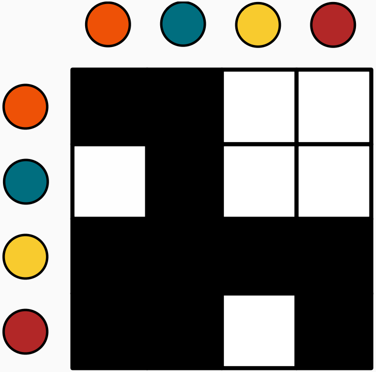
Order-based Approach [FK03]

3 Mask out inconsistent edges



3 Mask out inconsistent edges

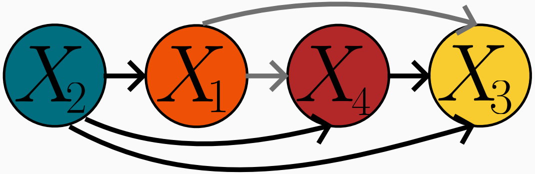


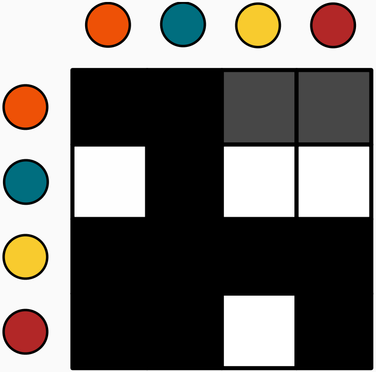


$\mathbf{R}^\sigma$: row and column permutation of strictly upper-triangular binary matrix: $\mathbf{R} \in \{0,1\}^{d \times d}$

Order-based Approach [FK03]

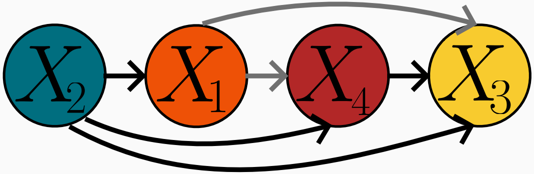
4 Prune unnecessary edges

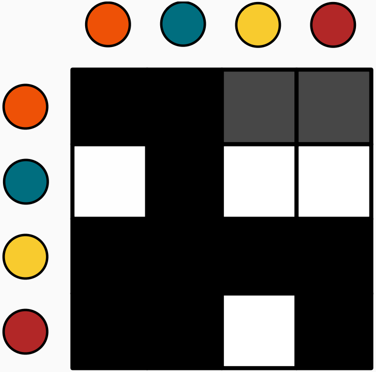




Order-based Approach [FK03]

4 Prune unnecessary edges





Space of orderings is smaller and more regular than space of DAGs [FK03, TK05]

Differentiable rank learning [BTBD20]

Score vector $\boldsymbol{\theta} \in \mathbb{R}^d$ inducing an ordering $\sigma(\boldsymbol{\theta}) \in \Sigma_d$
the smaller the score, the lower the rank

Differentiable rank learning [BTBD20]

Score vector $\boldsymbol{\theta} \in \mathbb{R}^d$ inducing an ordering $\sigma(\boldsymbol{\theta}) \in \Sigma_d$
the smaller the score, the lower the rank

Optimization Problem

$$\sigma(\boldsymbol{\theta}) \in \arg\max_{\sigma \in \Sigma_d} \boldsymbol{\theta}^\top \boldsymbol{\rho}^\sigma \,, \quad \text{where } \boldsymbol{\rho} = [1, 2, \ldots, d] \,.$$

degeneracy in case of ties (some components of $\boldsymbol{\theta}$ are equal)

Differentiable rank learning [BTBD20]

Score vector $\boldsymbol{\theta} \in \mathbb{R}^d$ inducing an ordering $\sigma(\boldsymbol{\theta}) \in \Sigma_d$

Optimization Problem

$$\sigma(\boldsymbol{\theta}) \in \arg\max_{\sigma \in \Sigma_d} \boldsymbol{\theta}^\top \boldsymbol{\rho}^\sigma \,, \quad \text{where } \boldsymbol{\rho} = [1, 2, \ldots, d] \,.$$

degeneracy in case of ties (some components of $\boldsymbol{\theta}$ are equal)

ORACLE $\sigma(\boldsymbol{\theta}) = \arg\operatorname{sort}(\boldsymbol{\theta})$ (due to The Rearrangement Inequality [HLP52]).

Differentiable rank learning [BTBD20]

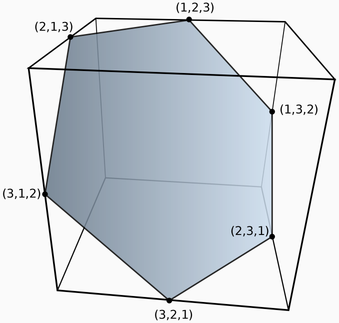
Score vector $\boldsymbol{\theta} \in \mathbb{R}^d$ inducing an ordering $\sigma(\boldsymbol{\theta}) \in \Sigma_d$

Relaxed Optimization Problem

$$\boldsymbol{\mu}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\mu} \in \mathbb{P}[d]} \boldsymbol{\theta}^\top \boldsymbol{\mu} - \frac{\tau}{2} \|\boldsymbol{\mu}\|_2^2$$

soft ordering $\boldsymbol{\mu}(\boldsymbol{\theta})$

Permutahedron $\mathbb{P}[d]$

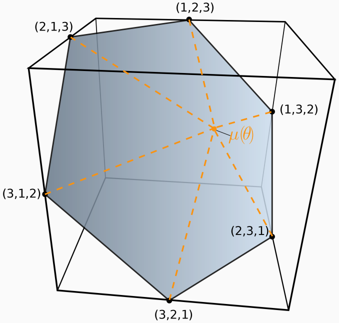


cc R. A. Nonenmacher

Differentiable rank learning [BTBD20]

Score vector $\boldsymbol{\theta} \in \mathbb{R}^d$ inducing an ordering $\sigma(\boldsymbol{\theta}) \in \Sigma_d$

Permutahedron $\mathbb{P}[d]$



cc R. A. Nonenmacher

Relaxed Optimization Problem

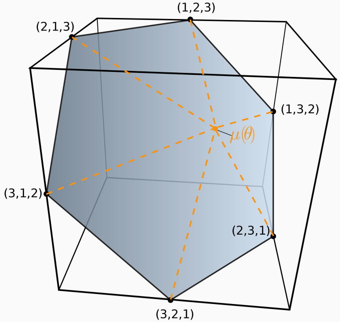
$$\boldsymbol{\mu}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\mu} \in \mathbb{P}[d]} \boldsymbol{\theta}^\top \boldsymbol{\mu} - \frac{\tau}{2} \|\boldsymbol{\mu}\|_2^2$$

soft ordering $\boldsymbol{\mu}(\boldsymbol{\theta})$

Differentiable rank learning [BTBD20]

Score vector $\boldsymbol{\theta} \in \mathbb{R}^d$ inducing an ordering $\sigma(\boldsymbol{\theta}) \in \Sigma_d$

Permutahedron $\mathbb{P}[d]$



cc R. A. Nonenmacher

Relaxed Optimization Problem

$$\boldsymbol{\mu}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\mu} \in \mathbb{P}[d]} \boldsymbol{\theta}^\top \boldsymbol{\mu} - \frac{\tau}{2} \|\boldsymbol{\mu}\|_2^2$$

soft ordering $\boldsymbol{\mu}(\boldsymbol{\theta})$

but cannot rank variables. We need a tractable decomposition of $\boldsymbol{\mu}(\boldsymbol{\theta})$ into hard orderings: cannot use all $d!$ orderings

SparseMAP [NMBC18]

Let $D = d!$ be the total number of orderings, and $\triangle^D$ be the D-dimensional simplex

$$\boldsymbol{\mu} = \sum_{\sigma \in \Sigma_d} \boldsymbol{\alpha}_\sigma \boldsymbol{\rho}^\sigma$$

SparseMAP [NMBC18]

Let $D = d!$ be the total number of orderings, and $\triangle^D$ be the D-dimensional simplex

$$\boldsymbol{\mu} = \sum_{\sigma \in \Sigma_d} \boldsymbol{\alpha}_\sigma \boldsymbol{\rho}^\sigma$$

Sparse decomposition - categorical regularization

$$\boldsymbol{\alpha}^{\text{sparseMAP}}(\boldsymbol{\theta}) \in \arg\max_{\boldsymbol{\alpha} \in \triangle^D} \boldsymbol{\theta}^\top \mathbb{E}_{\sigma \sim \boldsymbol{\alpha}}[\boldsymbol{\rho}_\sigma] - \frac{\tau}{2} \|\mathbb{E}_{\sigma \sim \boldsymbol{\alpha}}[\boldsymbol{\rho}_\sigma]\|_2^2 \,,$$

solved by Active-Set Algorithm [NW99] $\rightarrow$ calls to argsort oracle

Top-k Sparsemax [CNAM20]

Let $D = d!$ be the total number of orderings, and $\triangle^D$ be the D-dimensional simplex

$$\boldsymbol{\mu} = \sum_{\sigma \in \Sigma_d} \boldsymbol{\alpha}_\sigma \boldsymbol{\rho}^\sigma$$

Top-k Sparsemax [CNAM20]

Let $D = d!$ be the total number of orderings, and $\triangle^D$ be the D-dimensional simplex

$$\boldsymbol{\mu} = \sum_{\sigma \in \Sigma_d} \boldsymbol{\alpha}_\sigma \boldsymbol{\rho}^\sigma$$

Sparse decomposition - marginal regularization

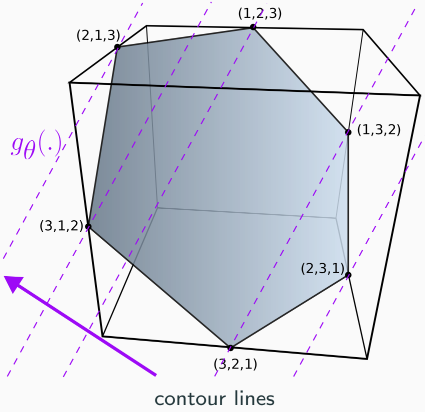
For $k > 2$

$$\boldsymbol{\alpha}^{\text{top-}k \text{ sparsemax}}(\boldsymbol{\theta}) \in \arg\max_{\boldsymbol{\alpha} \in \triangle^D, \|\boldsymbol{\alpha}\|_0 \leq k} \boldsymbol{\theta}^\top \mathbb{E}_{\sigma \sim \boldsymbol{\alpha}}[\boldsymbol{\rho}^\sigma] - \frac{\tau}{2} \|\boldsymbol{\alpha}\|_2^2 \,,$$

$\rightarrow$ calls to top-k permutations oracle

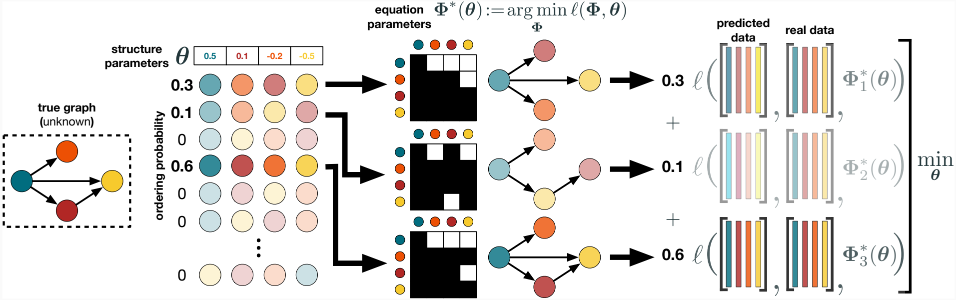
Top-k Permutations Oracle - Contribution!

Data: $k \in \{1, \ldots, d!\}$, $\boldsymbol{\theta} \in \mathbb{R}^d$
Result: top-k permutations $T_k(\boldsymbol{\theta})$
$P(\boldsymbol{\theta}) \leftarrow \{\sigma^1 \in_R \arg\max_{\sigma \in \Sigma_d} g_{\boldsymbol{\theta}}(\sigma)\}$;
while $|T_k(\boldsymbol{\theta})| \leq k$ **do**
  $\sigma \in_R \arg\max_{\sigma \in P(\boldsymbol{\theta}) \setminus T_k(\boldsymbol{\theta})} g_{\boldsymbol{\theta}}(\sigma)$;
  $P(\boldsymbol{\theta}) \leftarrow P(\boldsymbol{\theta}) \cup \{\sigma j \mid j \in \{1, \ldots, d-1\}\}$;
  $T_k(\boldsymbol{\theta}) \leftarrow T_k(\boldsymbol{\theta}) \cup \{\sigma\}$;
end



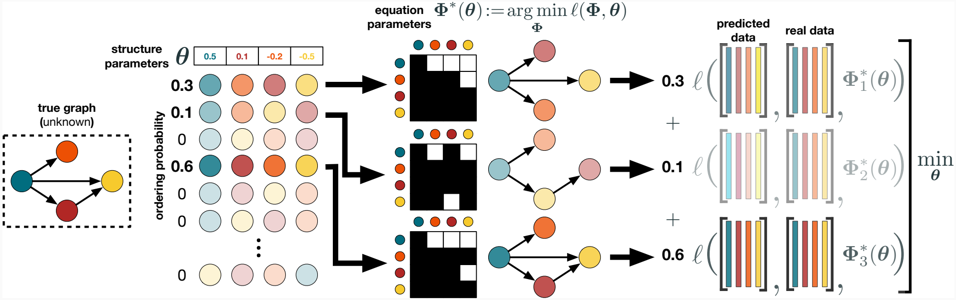
- set of candidates: $P(\boldsymbol{\theta})$
- best permutations: $T_k(\boldsymbol{\theta})$
- score: $g_{\boldsymbol{\theta}}(\sigma) = \boldsymbol{\theta}^\top \boldsymbol{\rho}^\sigma$
- adjacent transposition: $\sigma j := \sigma\ (j\ j{+}1)$

Overall DAG Learning Problem



$$\min_{\theta,\Phi} \mathbb{E}_{\sigma\sim\alpha^\star(\theta)}\left[\sum_{j=1}^{d} \ell\left(\mathbf{x}_j, f^{\phi_j}\left(\mathbf{X}\circ(\mathbf{R}^\sigma)_j\right)\right) + \lambda\Omega(\Phi)\right]$$

Overall DAG Learning Problem



$$\min_{\boldsymbol{\theta}} \ \mathbb{E}_{\sigma \sim \alpha^\star(\boldsymbol{\theta})} \left[\sum_{j=1}^{d} \ell\left(\mathbf{x}_j, f^{\phi^\star(\sigma)_j}\left(\mathbf{X} \circ (\mathbf{R}^\sigma)_j\right)\right) \right]$$

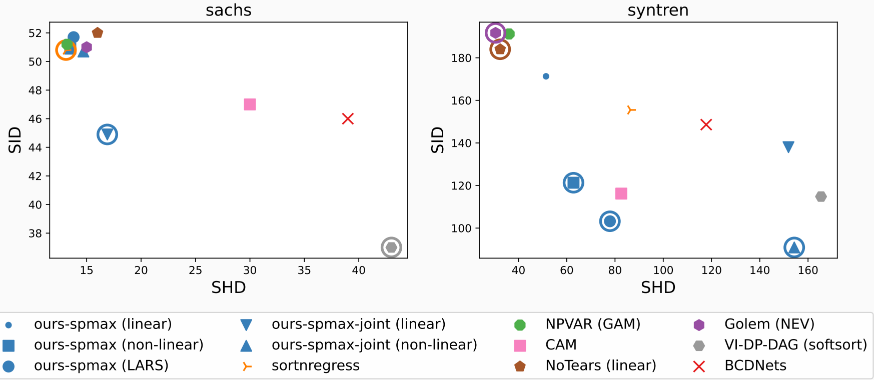
$$\text{s.t.} \quad \boldsymbol{\Phi}^\star(\sigma) = \arg\min_{\boldsymbol{\Phi}} \sum_{j=1}^{d} \ell\left(\mathbf{x}_j, f^{\phi_j}\left(\mathbf{X} \circ (\mathbf{R}^\sigma)_j\right)\right) + \lambda \Omega(\boldsymbol{\Phi})$$

Comparison with SOTA on Real Data

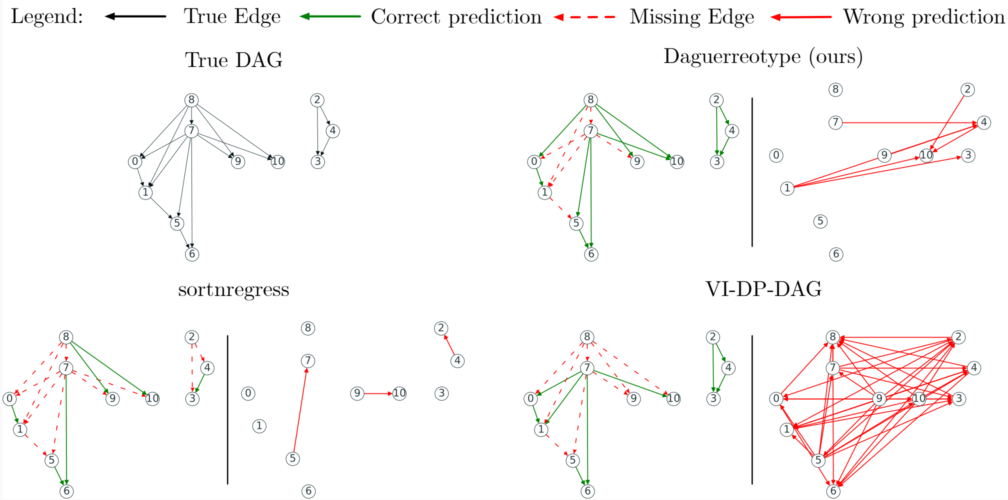
Metrics

SHD Structural Hamming Distance → # wrong edges

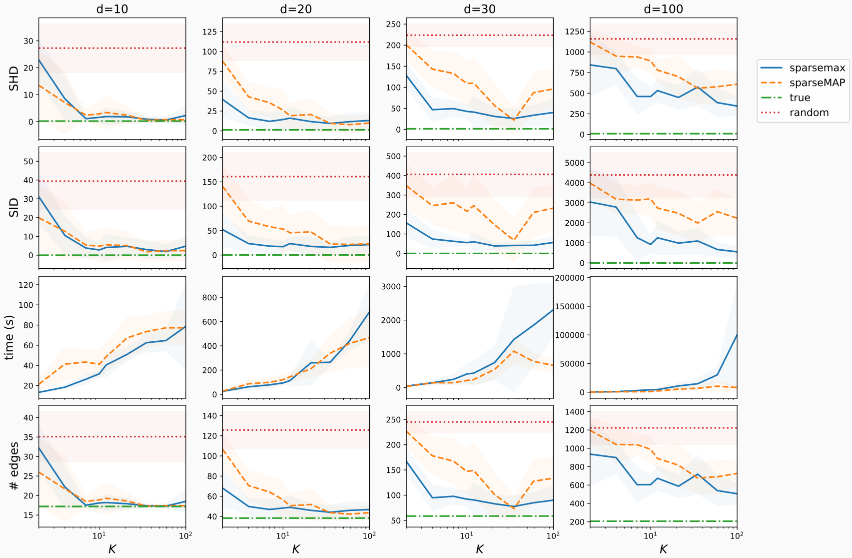
SID Structural Interventional Distance → # broken causal paths



Comparison with SOTA on Real Data



SparseMAP vs Top-k Sparsemax on Synthetic Data



Takeaways and Future Work

- Validity: DAG at any stage of training
- End-to-end: order and edges jointly optimized
- Modularity: can plug-in non-differentiable edge estimators
- Pareto-optimality: empirically best trade-off SHD-SID

Takeaways and Future Work

- Validity: DAG at any stage of training
- End-to-end: order and edges jointly optimized
- Modularity: can plug-in non-differentiable edge estimators
- Pareto-optimality: empirically best trade-off SHD-SID
- Scale-robustness? preliminary results suggest robust to variable scale

Takeaways and Future Work

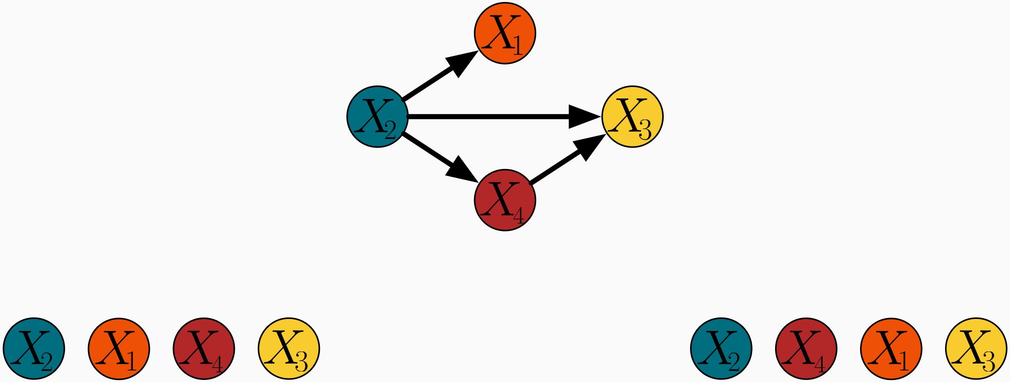
- Complexity: still at least quadratic in d

Takeaways and Future Work

- Complexity: still at least quadratic in d
- Sub-optimality: combinatorial space + relaxations

Takeaways and Future Work

- Complexity: still at least quadratic in d
- Sub-optimality: combinatorial space + relaxations
- Non-uniqueness: a DAG is consistent with multiple orderings



Takeaways and Future Work

- Complexity: still at least quadratic in d
- Sub-optimality: combinatorial space + relaxations
- Non-uniqueness: a DAG is consistent with multiple orderings
- need for better understanding of relationship DAG-space vs Order-space

Want to join the team?

Opening for research intern (remote or in Montreal)

https://www.servicenow.com/research/visiting_researcher.html

Luca Franceschi, AWS

Matt Kusner, UCL

Vlad Nicular, UVA

Link to arxiv: https://arxiv.org/submit/4710329

Thank you for your attention!

References i

[BAR22] Kevin Bello, Bryon Aragam, and Pradeep Ravikumar.
DAGMA: learning dags via m-matrices and a log-determinant acyclicity characterization.
Advances in Neural Information Processing Systems, 2022.

[BTBD20] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga.
Fast differentiable sorting and ranking.
In *International Conference on Machine Learning*, pages 950–959. PMLR, 2020.

[Chi95] David Maxwell Chickering.
Learning bayesian networks is np-complete.
In Doug Fisher and Hans-Joachim Lenz, editors, *Learning from Data - Fifth International Workshop on Artificial Intelligence and Statistics, AISTATS 1995, Key West, Florida, USA, January, 1995. Proceedings*. Springer, 1995.

[CNAM20] Gonçalo Correia, Vlad Niculae, Wilker Aziz, and André Martins.
Efficient marginalization of discrete and structured latent variables via sparsity.
Advances in Neural Information Processing Systems, 33, 2020.

[FK03] Nir Friedman and Daphne Koller.
Being bayesian about network structure. A bayesian approach to structure discovery in bayesian networks.
Machine learning, 50, 2003.

References ii

[HLP52] Godfrey Harold Hardy, John Edensor Littlewood, , and György Pólya.
Inequalities.
Cambridge University Press, 1952.

[LBDL20] Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien.
Gradient-based neural DAG learning.
In *ICLR*, 2020.

[NMBC18] Vlad Niculae, Andre Martins, Mathieu Blondel, and Claire Cardie.
Sparsemap: Differentiable sparse structured inference.
In *International Conference on Machine Learning*, pages 3799–3808. PMLR, 2018.

[NW99] Jorge Nocedal and Stephen J Wright.
Numerical optimization.
Springer, 1999.

[PJS17] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf.
Elements of causal inference: foundations and learning algorithms.
The MIT Press, 2017.

References iii

[RSW21] Alexander G Reisach, Christof Seiler, and Sebastian Weichwald.
Beware of the simulated dag! varsortability in additive noise models.
Advances in Neural Information Processing Systems, 34, 2021.

[TK05] Marc Teyssier and Daphne Koller.
Ordering-based search: A simple and effective algorithm for learning bayesian networks.
In *UAI*, pages 548–549. AUAI Press, 2005.

[YCGY19] Yue Yu, Jie Chen, Tian Gao, and Mo Yu.
DAG-GNN: DAG structure learning with graph neural networks.
In *ICML*, Proceedings of Machine Learning Research, 2019.

[ZARX18] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing.
Dags with NO TEARS: continuous optimization for structure learning.
In *Advances in Neural Information Processing Systems*, 2018.

[ZDA+20] Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing.
Learning sparse nonparametric dags.
In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. PMLR, 2020.