Gradient-Based Neural DAG Learning

Sébastien Lachapelle[1] **Philippe Brouillard**[1] **Tristan Deleu**[1] **Simon Lacoste-Julien**[1,2]

[1]Mila, Université de Montréal [2]Canada CIFAR AI Chair

Overview

Summary

- **Goal:**
  - Estimating a (causal) graph from observational data
  - Allowing for nonlinear relationships between variables
  - Leveraging a continuous constrained optimization formulation [7]
- **Method:**
  We propose GraN-DAG, a score-based approach using *neural networks* (NN) and an acyclicity constraint extended from [7] allowing gradient-based optimization
- **Motivation:**
  - Avoid combinatorial optimization usually required to learn a DAG
  - Instead, use well-known continuous constrained optimization technique such as an *augmented Lagrangian method*
- **Application:** E.g.: protein expression levels in human cells [5]

Related Work

- Continuous constrained formulation: NOTEARS [7], DAG-GNN [6]
- Discrete formulation solved greedily: CAM [1], GSF [3], GES [2]

Contributions

- **GraN-DAG**: Extends NOTEARS [7] (linear) to support NN (nonlinear)
- Empirical comparison to both *continuous* and *discrete* approaches
- Show GraN-DAG is competitive on both synthetic and real-world tasks

Background

Causal graphical models (CGM)

- P_X is a distribution over variable $X \in \mathbb{R}^d$ and $\mathcal{G} = (V, E)$ is a DAG
- $p(x) = \prod_{j=1}^{d} p(x_j | x_{\pi_j^{\mathcal{G}}})$ ($\pi_j^{\mathcal{G}}$ = parents of j in $\mathcal{G}$)
- CGM is like a *bayesian network*, but arrows are given *causal* meaning
- CGMs allow to ask: "What will happen if I intervene on X_j?"

Structure/causal learning & Identifiability

- Given n i.i.d. samples from P_X, estimate $\mathcal{G}$
- In general, it is impossible i.e. $\mathcal{G}$ is not *identifiable* from P_X
- Given a set of assumptions A on a CGM $(P_X, \mathcal{G})$, we say that $\mathcal{G}$ is identifiable from P_X if there exists no other CGM $(\tilde{P}_X, \tilde{\mathcal{G}})$ satisfying A such that $P_X = \tilde{P}_X$ and $\mathcal{G} \neq \tilde{\mathcal{G}}$
- Need assumptions: *faithfulness* or restrictions on $p(x_j | x_{\pi_j^{\mathcal{G}}}) \ \ \forall j$
- E.g. $X_j | X_{\pi_j^{\mathcal{G}}} \sim \mathcal{N}(f_j(X_{\pi_j^{\mathcal{G}}}), \sigma_j^2) \ \ \forall j \implies \mathcal{G}$ is *identifiable* from P_X [1]
- Score-based formulation: $\hat{\mathcal{G}} = \arg\max_{\mathcal{G} \in \text{DAG}} \mathcal{S}(\mathcal{G})$
- Popular approaches greedily maximize a regularized likelihood [1, 3, 2]

Continuous optimization for DAG learning

- DAGs with NOTEARS [7] assumes a linear model: $X_j := u_j^\top X + \epsilon_j$
- $U = [u_1 | ... | u_d] \in \mathbb{R}^{d \times d}$ is interpreted as a *weighted adjacency matrix* and $U_{ij} = 0 \implies X_i$ is not a parent of X_j
- Enforce acyclicity by $\operatorname{Tr} e^{U \odot U} = d$ and solve w/ *augmented Lagrangian*
- **Constraint intuition:** Let B be a binary adjacency matrix
  $(B^k)_{ij}$ = number of paths of length k from i to j
  $\operatorname{Tr} e^B - d = \sum_{k=1}^{\infty} \frac{\operatorname{Tr} B^k}{k!} \approx$ number of cycles of every lengths

Performance metrics for graph estimation

SHD: Counts the number of missing, falsely detected or reversed edges
SID: Counts the number of couples (i, j) such that the interventional distribution $p(x_j | do(X_i = \bar{x}))$ would be miscalculated if we were to use the estimated graph to form the parent adjustment set

References

[1] Bühlmann, P., Peters, J., & Ernest, J. (2014). CAM: Causal additive models, high-dimensional order search and penalized regression. *Annals of Statistics*.

[2] Chickering, D. (2003). Optimal structure identification with greedy search. *Journal of Machine Learning Research*.

[3] Huang, B., Zhang, K., Lin, Y., Schölkopf, B., & Glymour, C. (2018). Generalized score functions for causal discovery. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[4] Lachapelle, S., Brouillard, P., Deleu, T., & Lacoste-Julien, S. (2019). Gradient-based neural DAG learning. *CoRR*, abs/1906.02226.

[5] Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D., & Nolan, G. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*.

[6] Yu, Y., Chen, J., Gao, T., & Yu, M. (2019). DAG-GNN: DAG structure learning with graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*.

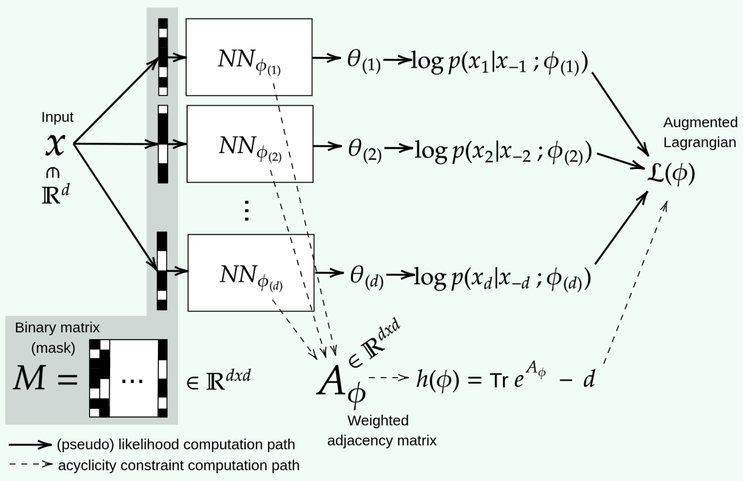
[7] Zheng, X., Aragam, B., Ravikumar, P., & Xing, E. (2018). Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems 31*.

GraN-DAG

The method

- Learns d MLPs of L hidden layers, each denoted by $\text{NN}_{\phi_{(j)}}$
  $\phi_{(j)} \triangleq \{W_{(j)}^{(\ell)}\}_{\ell=1}^{L+1}$ and $W_{(j)}^{(\ell)} \triangleq$ weight matrix of layer ℓ in NN j
- MLP j outputs a parameter $\theta_{(j)} \in \mathbb{R}^m$ used to compute $p(x_j | x_{-j}; \phi_{(j)})$
- $\prod_{j=1}^{d} p(x_j | x_{-j}; \phi_{(j)})$ is not a valid pdf (does not decompose wrt a DAG!)
- We propose a *weighted adjacency matrix* $A_\phi \in \mathbb{R}_{\geq 0}^{d \times d}$ which can be used in the NOTEARS constraint [7] to enforce acyclicity

Computation graph



Neural network connectivity

- *Path product*: $|W_{h_1 i}^{(1)}||W_{h_2 h_1}^{(2)}| \dots |W_{k h_L}^{(L+1)}| \geq 0 \quad (= 0$ path *inactive*)
  i.e. strength of the NN path $(i, h_1, h_2, ..., h_L, k)$
- For each NN j, consider the matrix product of the weights in abs. value:

$$C_{(j)} \triangleq |W_{(j)}^{(L+1)}| \dots |W_{(j)}^{(2)}||W_{(j)}^{(1)}| \in \mathbb{R}_{\geq 0}^{m \times d}$$

- $\sum_{k=1}^{m} (C_{(j)})_{ki} =$ sum of all the *path products* from X_i to parameter $\theta_{(j)}$

Constraint & Optimization

$$\text{Define} \quad (A_\phi)_{ij} \triangleq \begin{cases} \sum_{k=1}^{m} (C_{(j)})_{ki}, & \text{if } j \neq i \\ 0, & \text{otherwise} \end{cases}$$

- By construction, $(A_\phi)_{ij} = 0 \implies \theta_{(j)}$ does not depend on variable X_i
- Hence, we can use A_ϕ in the acyclicity constraint of [7], yielding

$$\max_\phi \mathbb{E}_{X \sim P_X} \sum_{j=1}^{d} \log p(X_j | X_{\pi_j^\phi}; \phi_{(j)}) \quad \text{s.t.} \quad \operatorname{Tr} e^{A_\phi} - d = 0$$

- Solve approximately using an *augmented Lagrangian method*
- A_ϕ is thresholded using a binary mask M (see figure and our paper [4])

Avoiding overfitting

- Note that adding more edges never reduces the maximum likelihood score
- To avoid spurious edges, we perform a final *DAG pruning* step identical to CAM [1] by fitting a generalized additive model and performing a significance test of covariates
- When $d \geq 50$, a *preliminary neighbors selection* step is applied to restrict the number of potential parents, similar to CAM [1]
- Moreover, we use *early stopping* on each subproblem of the augmented Lagrangian

Experiments and conclusion

- Synthetic data: performance averaged over 10 graphs
- ER and SF are two graph sampling schemes
- $d =$ number of nodes, $e =$ average number of edge per graph

	ER $d=50$ $e=50$		SF $d=50$ $e=200$		Protein data set [5]	
	SHD	SID	SHD	SID	SHD	SID
GraN-DAG	**5.1±3**	**22.4±18**	**111.3±12**	**271.2±65**	13	47
DAG-GNN	49.2±8	304.4±105	144.9±13	540.8±151	16	**44**
NOTEARS	62.8±9	327.3±120	153.7±12	558.4±154	21	**44**
CAM	**4.3±2**	**22.0±18**	**111.2±13**	**320.7±153**	**12**	55
GSF	25.6±5	[21.1±23 79.2±34]	**120.2±11**	[284.7±80 379.9±98]	18	[44, 61]
RANDOM	535.7±401	272.3±126	660.6±195	1198.9±305	21	60

- On synthetic tasks, GraN-DAG outperforms other continuous approaches and is competitive with best performing greedy approach (CAM)
- GraN-DAG is also competitive on the real-world protein data set
- See our paper [4] for more experiments (graphs of up to 100 nodes)
- Our code: `https://github.com/kurowasan/GraN-DAG`