DAGs with No Curl:

An Efficient DAG Structure Learning Approach

Yue Yu
Department of Mathematics, Lehigh University
Tian Gao
IBM Research
Naiyu Yin, Qiang Ji
Department of Electrical, Computer, and Systems Engineering, RPI

Present at ICML 2021

DAG Learning: SoTA Methods

- DAG learning plays a vital part in other machine learning sub-areas such as causal inference However, it is proven to be NP-hard.

- **Conventional DAG learning methods:**

  1) Make a parametric (e.g. Gaussian) assumption for continuous variables: may result in model misspecification.

  2) Perform score-and-search for **discrete variables**: with a constraint stating that the graph must be acyclic.

$$A^* = \underset{A}{\operatorname{argmin}} \quad F(A, \mathbf{X}), \qquad \text{subject to} \qquad \mathcal{G}_A \in \mathbb{D}$$

- **DAG learning as a continuous optimization:**

  An equivalent acyclicity constraint by Zheng et al*: $\mathcal{G}_A \in \mathbb{D} \Longleftrightarrow h(A) = \operatorname{tr}(\exp(A \circ A)) - m = 0$

continuous constraint optimization

$$A^* = \underset{A}{\operatorname{argmin}} \quad F(A, \mathbf{X}), \qquad \text{subject to} \qquad h(A) = 0,$$

Zheng, X., Aragam, B., Ravikumar, P. K., Xing, E. P. (2018). DAGs with NO TEARS: Continuous Optimization for Structure Learning. In Advances in Neural Information Processing Systems (pp. 9472-9483).

An Equivalent DAG Space

Idea: Can we remove the constraint, by solving directly in the DAG space?

- Reformulate the DAG space: **acyclic = a (curl-free) gradient flow**

Topological Ordering!!

Theorem 1:
Consider a complete undirected graph G(V,E), given any function $p \in L^2(V) = \mathbb{R}^m$ and any skew-symmetric matrix $W \in \mathbb{R}^{m\times m}$, $W \circ \mathrm{ReLU}(\mathrm{grad}(p))$ is the weighted adjacency matrix of a DAG.

Theorem 2:
Let $A \in \mathbb{R}^{m\times m}$ be the weighted adjacency matrix of a DAG, then there exists a skew-symmetric matrix $W \in \mathbb{R}^{m\times m}$ and a potential function $p \in L^2(V) = \mathbb{R}^m$ such that $A = W \circ \mathrm{ReLU}(\mathrm{grad}(p))$.

⇒ $$\{\mathcal{G}_{W\circ \mathrm{ReLU}(\mathrm{grad}(p))}\} = \{\mathrm{DAGs}\}.$$

An Efficient Projection Algorithm

Idea: Can we remove the constraint, by solving directly in the DAG space?

- Reformulate the DAG space: $W \in \mathbb{R}^{m \times m},\ \ p \in L^2(V) = \mathbb{R}^m$

$$\{\mathcal{G}_{W \circ \mathrm{ReLU}(\mathrm{grad}(p))}\} = \{\mathrm{DAGs}\}.$$

- A new continuous optimization **without constraint**:

$$(W^*, p^*) = \underset{W \in \mathbb{R}^{m \times m}, W = -W^T, p \in \mathbb{R}^m}{\mathrm{argmin}} F(W \circ \mathrm{ReLU}(\mathrm{grad}(p)), \mathbf{X})$$

- Instead of solving for A, we solve for W and p, then obtain an optimal DAG via:

$$A^* = W^* \circ \mathrm{ReLU}(\mathrm{grad}(p^*))$$

However, like NOTEARS, this optimization problem is non-convex, even for linear SEM

An Efficient Projection Algorithm

Idea: Can we provide initial guesses by projecting any cyclic graph to the DAG space?

- How to solve the non-convex optimization problem

$$(W^*, p^*) = \underset{W \in \mathbb{R}^{m\times m}, W=-W^T, p\in\mathbb{R}^m}{\operatorname{argmin}} F(W \circ \mathrm{ReLU}(\mathrm{grad}(p)), \mathbf{X})$$

Theorem 3 (Projection Method):
Let $A \in \mathbb{R}^{m\times m}$ be the weighted adjacency matrix of a DAG, C(A) denotes the connectivity matrix of A, then

$$p = -\Delta_0^{\dagger}\mathrm{div}\left(\frac{1}{2}(C(A) - C(A)^T)\right),$$

preserves the topological order in A. Moreover, taking

$$[W]_{ij} = \begin{cases} 0, & \text{if } p(i) = p(j) \text{ or } A(i,j) = A(j,i) = 0; \\ \frac{A(i,j)}{p(j)-p(i)}, & \text{if } A(i,j) \neq 0 \text{ and } A(j,i) = 0; \\ \frac{A(j,i)}{p(j)-p(i)}, & \text{if } A(i,j) = 0 \text{ and } A(j,i) \neq 0. \end{cases}$$

we have $A = W \circ \mathrm{ReLU}(\mathrm{grad}(p))$

For any (cyclic) A, we can project it to the DAG space.

DAG-NoCurl

Overall recipe for DAG-NoCurl

1. **Prediction phase**: Solve for an initial prediction $A^{pre} \in \mathbb{R}^{m \times m}$ via:

$$A^{pre} = \underset{A}{\operatorname{argmin}} \; F(A, \mathbf{X}) + \lambda h(A)$$

2. **Projection phase**: Based on A^{pre}, obtain an approximate solution of p with the projection method:

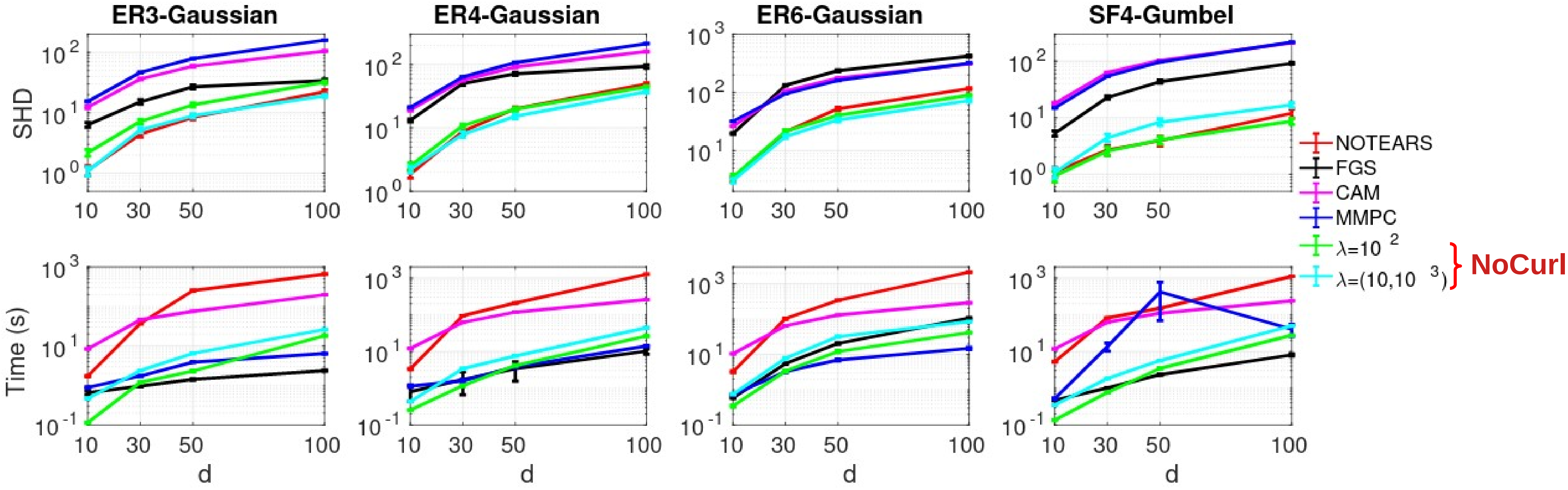
$$\tilde{p} = -\Delta_0^{\dagger} \operatorname{div} \left(\frac{1}{2} (C(A^{pre}) - C(A^{pre})^T) \right),$$

and solve for W via: $$\tilde{W} = \underset{W \in S}{\operatorname{argmin}} \; F(W \circ \operatorname{ReLU}(\operatorname{grad}(\tilde{p})), \mathbf{X}).$$

3. **Final approximation**: $\tilde{A} = \tilde{W} \circ \operatorname{ReLU}(\operatorname{grad}(\tilde{p}))$ and apply thresholding to remove false discoveries.

DAG-NoCurl

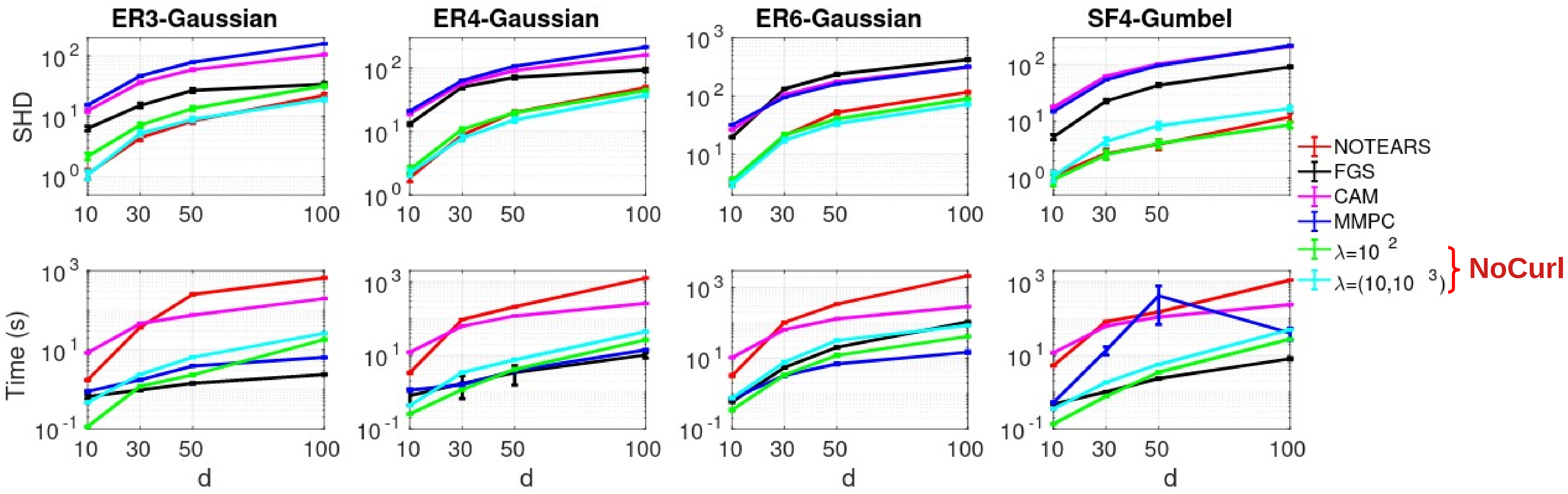
Results: linear SEM



Accuracy: NoCurl achieves a similar accuracy, and sometimes beats NOTEARS, especially on dense and large graphs

DAG-NoCurl

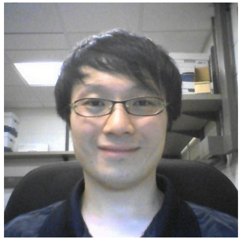
Results: linear SEM



Efficiency: NoCurl requires a similar runtime as FGS and MMPC, which is faster than NOTEARS by one or two orders of magnitude.

Thank You!

- Co-authors:
  Tian Gao (IBM), Naiyu Yin, Qiang Ji (RPI).



- Funding support:
  NSF CAREER award DMS1753031
  DARPA grant FA8750-17-2-0132

- For further analysis and results on nonlinear and real-world datasets, please stop by our poster.

Poster ID: 937

- Codes and datasets are available at
  https://github.com/fishmoon1234/DAG-NoCurl