Optimizing NOTEARS Objectives via Topological Swaps

Chang Deng[1] Kevin Bello[1 2] Bryon Aragram[1] Pradeep Ravikumar[2]

June 29, 2023

[1]Booth School of Business, University of Chicago, [2]Machine Learning Department, Carnegie Mello University

A Class of Nonconvex Problem

- Problem: we study a class of constrained nonconvex optimization problems (NOTEARS) defined as follows:

$$\min_{\Theta} \; Q(\Theta) \text{ subject to } h(W(\Theta)) = 0$$

- Goal: We solve this class of problems and provide optimality guarantees.

Background

Problem: Learning Directed Acyclic Graph From Data

- The goal (DAG learning) is to recover the underlying DAG of a structural equation model (SEM) from data. A nonparametric SEM consists of a set of equations of the form,

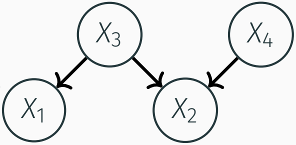
$$X_j = f_j(X, z_j), \quad j = 1, \ldots, d$$

  where each f_j is nonparametric function, z_j represents noise.
- The graphical structure implied by $(f_1, \ldots, f_d)$ can be represented by the following $d \times d$ weighted adjacency matrix

$$W = W(f) = (w_{ij}) \quad w_{ij} = \|\partial_i f_j\|_2$$

  Indeed, such a graphical structure is a DAG.

Example

- One of simple example is Linear SEM: $X_j = W_j^\top X + z_j$, where $X = [X_1, \ldots, X_d]$ is data and $W = [W_1, \ldots, W_d]$ represents the weighted adjacency matrix.
- $$\begin{array}{c} \begin{matrix} X_1 & X_2 & X_3 & X_4 \end{matrix} \\ \begin{bmatrix} 0.3 & -0.2 & 0.4 & 0.9 \\ 0.9 & 1.1 & 0.3 & -1.6 \\ 1.1 & 0.2 & -0.4 & 0.6 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \end{array} \xrightarrow{\text{estimate}}$$ 

- DAG learning is important in several fields, such as economics, social science, genetics, and causal inference.

Score-based approach

- Score-based methods searches for the (weighted) adjacency matrix W that minimizes a given score Q that measures how well W fits the observed data $\mathbf{X}$. That is we aim to solve

  $$\min_{W} \; Q(W) \quad \text{s.t.} \quad W \in \text{DAGs } (\textit{Combinatorial Constraint})$$

  The above problem is known to be *NP-complete* to solve(Chickering 1996).

A Continuous Non-convex Characterizations of DAGs

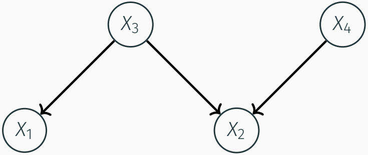
- Recent work (**NOTEARS**) by Zheng et al. (2018) has replaced the combinatorial DAG constraint to a continuous constraint via the smooth function $h_{\exp}(W) = \mathrm{Tr}(e^{W \circ W}) - d$. That is

$$\min_{W} \; Q(W) \qquad \text{s.t. } h_{\exp}(W) = 0$$

  $h_{\exp}(W) = 0$ if and only if W is a DAG.
- More formulation about the continuous characterization of DAGs has been proposed. Check Wei et al. (2020), Yu et al. (2019), and Bello et al.(2022) for more formulas.
- $h_{\exp}(W)$ is smooth nonconvex function.

Definition(Topological sort)

Topological sort π is a a permutation on vertices, $X_{\pi(i)} \to X_{\pi(j)} \Rightarrow i < j$.



X_3 comes before X_2 and X_1, X_4 comes before X_2, any order that is consistent with it will be topological sort, i.e. $\pi = [3, 4, 2, 1]$

Property of Topological sort

- For any DAG, there exists at least one corresponding topological sort π (maybe not unique).
- We call a graph G (resp. W) *consistent* with π if π is topological sort of G (resp. W) and write this as $G \sim \pi$ (resp. $W \sim \pi$)
- Given a permutation π, we then have the following order-constrained optimization problem:

$$\min_{W \sim \pi} Q(W)$$

  We denote the optimal solution by W^*_π.
- Basically, it is unconstrained optimization that can be solved efficiently.

Topological sort

- Equivalent formulation

$$
\begin{aligned}
W^*_\pi = & \arg\min_W Q(W) \\
& \text{s.t. } W_{\pi(i),\pi(j)} = 0, \forall j \le i
\end{aligned}
$$

- $\pi = [3, 1, 2] \xrightarrow{\text{opt}} W^*_\pi \xrightarrow{\text{recover}} X_1 \leftarrow X_3 \rightarrow X_2$ 😎

A Topological Sort-Based Algorithm Informed by KKT Condition

An Useful Set

For any $\tau, \xi \geq 0$ and any W, define a set of pairs

$$\mathcal{Y}(W, \tau, \xi) \stackrel{\text{def}}{=} \left\{ (i,j) \mid [\nabla h(|W|)]_{ij} \leq \tau, \left\| \frac{\partial Q(W)}{\partial W_{ij}} \right\|_1 > \xi \right\}.$$

Theorem (Property of $\mathcal{Y}(W, \tau, \xi)$)

- $\mathcal{Y}(W_\pi^*, 0, 0) = \emptyset \Rightarrow W_\pi^*$ satisfies the KKT conditions. Under the assumption Q is convex, W_π^* is a local minimal.
- Under some mild condition, $\mathcal{Y}(W_\pi^*, 0, 0) \neq \emptyset$ for some topological sort π, then

$$Q(W_{\pi_{ij}}^*) < Q(W_\pi^*)$$

for every $(i,j) \in \mathcal{Y}(W_\pi^*, 0, 0)$, where π_{ij} is the topological sort that is learned through a simple procedure.

Implication of the Set

- In a word, $\mathcal{Y}(W_\pi^*, \tau, \xi)$ provides information about whether W_π^* is locally optimal and also identifies which new topological sort π_{ij} could potentially improve the score.
- We replace $\mathcal{Y}(W_\pi^*, 0, 0)$ by $\mathcal{Y}(W_\pi^*, \tau, \xi)$ where τ, ξ are positive, to enlarge the searching space, and we can design an algorithm based on set $\mathcal{Y}(W_\pi^*, \tau, \xi)$.

Algorithm

TOPO

- Initialize arbitrary sort π, get W^*_π.
- Define a candidate set of possible swaps by $\mathcal{Y}(W^*_\pi, \tau, \xi)$
- Choose the best swap from this set to obtain a new topological sort; i.e., the swap that decreases the score Q the most.
- Repeat until there is no sufficient improvement in the score.

Property of TOPO Algorithm

Theorem

Under some mild conditions, and Q is convex (resp. non-convex). Then TOPO with arbitrary initial topological sort π returns a local minimum (resp. KKT point) of problem, where the score is decreased at each iteration. Moreover, the solution at each iteration is also a local minimum. (resp. KKT point)

Why care about KKT/local optimal points?

- KKT conditions are indeed necessary conditions of optimality, i.e. they are satisfied by all local minima. When Q is convex, the KKT condition is also the sufficient condition of optimality.
- Improving the solution of NOTEARS objective can lead to better recovery of the underlying structure.

Experiments

Linear Model

Method	Metric	$d = 20$	$d = 40$	$d = 100$
GOLEM-EV	KKT	No	No	No
	Loss	10.7 ± 0.12	40.7 ± 4.8	68.8 ± 3.9
	SHD	11.4 ± 3.4	51.4 ± 28.3	145.2 ± 52.6
NOTEARS	KKT	No	No	No
	Loss	11.9 ± 0.1	62.1 ± 8.8	73.1 ± 7.6
	SHD	28.6 ± 3.2	129 ± 25.5	140.0 ± 30.1
NOFEARS	KKT	Yes	Yes	Yes
	Loss	11.5 ± 0.3	47.6 ± 1.6	61.2 ± 2.6
	SHD	23.2 ± 4.5	69.8 ± 16.0	87.5 ± 19.2
NOTEARS-TOPO	KKT	Yes	Yes	Yes
	Loss	**9.8 ± 0.1**	**38.4 ± 0.1**	**47.5 ± 0.1**
	SHD	**0.4 ± 0.2**	9.2 ± 0.8	**14.2 ± 1.9**
RANDOM-TOPO	KKT	Yes	Yes	Yes
	Loss	**9.8 ± 0.1**	**38.4 ± 0.1**	**47.5 ± 0.1**
	SHD	**0.4 ± 0.2**	**8.6 ± 0.9**	16.3 ± 2.6

Table 1: Experiments on linear DAGs on ER4 graphs.

Neural Networks

Method	Metric	$d = 10$	$d = 20$	$d = 40$
NOTEARS-MLP	KKT	No	No	No
	Loss	7.2 ± 0.2	14.4 ± 0.3	28.5 ± 0.4
	SHD	5.6 ± 0.7	29.1 ± 3.1	112.3 ± 20.2
NOTEARS-TOPO	KKT	Yes	Yes	Yes
	Loss	**6.4 ± 0.1**	**11.6 ± 0.1**	**22.8 ± 0.6**
	SHD	**2.7 ± 0.5**	**12.1**	**36.3 ± 20.4**
TRUE	KKT	Yes	Yes	Yes
	Loss	6.3 ± 0.1	12.2 ± 0.1	23.4 ± 0.4
	SHD	2.1 ± 0.5	11.6 ± 0.6	36.1 ± 2.2

Table 2: Experiments on Nonlinear Model with Neural Network on ER4 graphs. Here 'True' means the solution W_π^* using the underlying true topological sort.