Global Optimality in Bivariate Gradient-based DAG Learning

Chang Deng[†] Kevin Bello[†‡] Bryon Aragram[†] Pradeep Ravikumar[‡]
[†]Booth School of Business, University of Chicago
[‡]Machine Learning Department, Carnegie Mello University

https://arxiv.org/abs/2306.17378

A class of nonconvex problem

Problem: We study a class of constrained nonconvex optimization problem [Zheng et al., 2018], which is related to learning Directed Acyclic Graphs(DAG) from observational data, and defined as follows :

$$\min_{\Theta} \quad Q(\Theta) \qquad \text{subject to } h(W(\Theta)) = 0 \tag{1}$$

where Θ^l corresponds to all model parameters, and $W(\Theta) \in \mathbb{R}^{d\times d}$ is weigted adjacency matrix, induced by Θ. Moreover, $Q : \mathbb{R}^l \to \mathbb{R}$ is refer to as the score function. $h : \mathbb{R}^{d\times d} \to [0, \infty)$ is nonnegative **non-convex** differentiable function that penalizes cycles.

Motivation

Multiple empirical studies have shown the the global or near-global minimizer of (1) can be found in a variety of setting, such as linear models with Gaussian and non-Gaussian noises [Bello et al., 2022, Ng et al., 2022, Zheng et al., 2018], and nonlinear models, represented by neural networks, with additive Gaussian noises [Lachapelle et al., 2020, Yu et al., 2019, Zheng et al., 2020].

Instead of solving (1) directly, researchers have considered some type of penalty method such as augmented Lagrangian, quadratic penalty, and a log-barrier. In all cases, the penalty approach consists of solving a *sequence* of unconstrained non-convex problem, where the constrained is enforced progressively.

$$\min_{\Theta} g_{\mu_k}(\Theta) := \mu_k f(\Theta) + h(W(\Theta)) \tag{2}$$

These methods are called homotopy methods.

Two natural questions

Motivated by the empirical success of solving (1) by penalty method, one is inevitably led to ask the following questions

- *Are the loss lanscapes* $g_{\mu_k}(\Theta)$ benign for different μ_k?
- *Is there a (tracable) solution path* $\{\Theta_k\}$ *that converges to a gloabl minimum of* (1)*?*

Due to the NP-completeness of learning DAGs, the first answer would be expected to be negative.

For second question, we seek a solution path that can be tractably computed in practice, e.g. by gradient descent.

Bivariate case

An ideal case

We focus on the perhaps the simplest setting (Bivariate case) where interesting phenomena take place. Although the simplistic of bivariate setting, it provides a valuable starting point for future research in more complex settings! Moreover, we study how (2) is **actually solved** in practice.

- Random variables: $X = (X_1, X_2) \in \mathbb{R}^2$
- Independent errors: $N = (N_1, N_2) \in \mathbb{R}^2$ with equal variance, i.e., $\mathrm{Var}(N_1) = \mathrm{Var}(N_2)$.
- Structural Equation Model: $X = W_*^\top X + N$ where W_* is a weighted adjacent matrix encoding the coefficients in the linear model. Moreover, W_* is acyclic. Without loss of generality,

$$W_* = \begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix}$$

Mathematical formulation

In our setting, the problem can be formulated as

$$\begin{aligned}\min_{W} f(W) :=& \frac{1}{2}\mathbb{E}_X\left[\|X - W^\top X\|_2^2\right] \\ \min_{x,y} f(x,y) :=& \frac{1}{2}\left((1-ay)^2 + y^2 + (a-x)^2 + 1\right) \\ \text{s.t. } h(x,y) :=& \frac{x^2y^2}{2} = 0\end{aligned} \tag{3}$$

The penalized version can be written as

$$\begin{aligned}\min_{x,y} g_\mu(x,y) :=& \mu f(x,y) + h(x,y) \\ =& \frac{\mu}{2}\left((1-ay)^2 + y^2 + (a-x)^2 + 1\right) + \frac{x^2y^2}{2}\end{aligned} \tag{4}$$

Geometry of $g_\mu(W)$

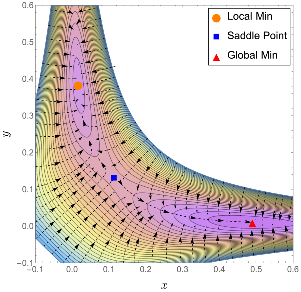
Lemma

There exists $\tau > 0$, then $\forall \mu < \tau$, the equation $\nabla g_\mu(W) = 0$ has three different solutions, denoted as $W_\mu^*, W_\mu^{**}, W_\mu^{***}$.

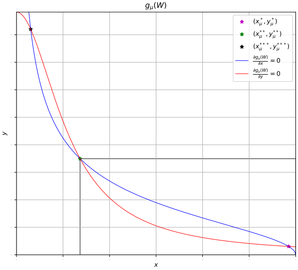
$$\lim_{\mu\to 0} W_\mu^* = \begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix}, \lim_{\mu\to 0} W_\mu^{**} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \lim_{\mu\to 0} W_\mu^{***} = \begin{pmatrix} 0 & 0 \\ \frac{a}{a^2+1} & 0 \end{pmatrix}$$

Moreover, W_μ^* is global minima, W_μ^{***} is local minima, and W_μ^{**} is a saddle point.

Geometry of $g_\mu(W)$



(a) Contour plot



(b) Stationary points

Figure: Visualizing the nonconvex landscape. (a) A contour plot of g_μ for $a = 0.5$ and $\mu = 0.005$. We only show a section of the landscape for better visualization. The solid lines represent the contours, while the dashed lines represent the vector field $-\nabla g_\mu$. (b) Stationary points of g_μ

A good scheduling of μ_k is needed to avoid being trapped in a local minimum!

Algorithm

Algorithm 1 GradientFlow(f, z_0)

set $z(0) = z_0$
$\frac{d}{dt} z(t) = -\nabla f(z(t))$
return $\lim_{t\to\infty} z(t)$

Algorithm 2 Homotopy algorithm for solving (3)

Input: Initial $W_0 = W(x_0, y_0)$, $\mu_0 \in \left[\frac{a^2}{4(a^2+1)^3}, \frac{a^2}{4}\right)$
Output: $\{W_{\mu_k}\}_{k=0}^{\infty}$
$W_{\mu_0} \leftarrow$ GradientFlow(g_{μ_0}, W_0)
for $k = 1, 2, \ldots$ **do**
    Let $\mu_k = (2/a)^{2/3} \mu_{k-1}^{4/3}$
    $W_{\mu_k} \leftarrow$ GradientFlow$(g_{\mu_k}, W_{\mu_{k-1}})$
end for

Convergence to Global Optimum

Theorem

For any initialization W_0 and $a \in \mathbb{R}$, the solution path provided in Algorithm 2 converges to the global optimum of (3), i.e.,

$$\lim_{k \to \infty} W_{\mu_k} = W_{\mathsf{G}}.$$

where W_{G} is global optimum of (3).

Under our setting, $W_{\mathsf{G}} = W_*$, which implies we recover the ground truth W_*.

A practical homotopy algorithm

Algorithm 3 Practical (i.e. independent of a and W_*) Homotopy algorithm for solving (3)

1: **Input:** Initial $W_0 = W(x_0, y_0)$
2: **Output:** $\{W_{\mu_k}\}_{k=0}^{\infty}$
3: $\mu_0 \leftarrow \frac{1}{27}$
4: $W_{\mu_0} \leftarrow \text{GradientFlow}(g_{\mu_0}, W_0)$
5: **for** $k = 1, 2, \ldots$ **do**
6: Let $\mu_k = (2/\sqrt{5\mu_0})^{2/3}\mu_{k-1}^{4/3}$
7: $W_{\mu_k} \leftarrow \text{GradientFlow}(g_{\mu_k}, W_{\mu_{k-1}})$
8: **end for**

Lemma

Assume $a > \sqrt{5/27}$, then for any initialization W_0, Algorithm 3 outputs the global optimal solution to (3), i.e., $\lim_{k\to\infty} W_{\mu_k} = W_{\mathsf{G}}$.

From gradient flow to gradient descent

Gradient flow is used to locate the next stationary points, which is not practically feasible. A viable alternative is to replace gradient flow with gradient descent.

Theorem (Informal)

For any $\varepsilon_{\textit{dist}} > 0$, set μ_0 satisfy a mild condition, and use $\epsilon_k = \min\{\beta a \mu_k, \mu_k^{3/2}\}, \mu_{k+1} = (2\mu_k^2)^{2/3} \frac{(a+\epsilon_k/\mu_k)^{2/3}}{(a-\epsilon_k/\mu_k)^{4/3}}$, and let $K \equiv K(\mu_0, a, \varepsilon_{\textit{dist}}) \in O\left(\ln \frac{\mu_0}{a \varepsilon_{\textit{dist}}}\right)$. Then, for any initialization W_0, following the updated procedure above for $k = 0, \ldots, K$, we have:

$$\| W_{\mu_k, \epsilon_k} - W_{\mathsf{G}} \|_2 \leq \varepsilon_{\textit{dist}}$$

that is, W_{μ_k, ϵ_k} is $\varepsilon_{\textit{dist}}$-close in Frobenius norm to global optimum W_{G}. Moreover, the total number of gradient descent steps is upper bounded by $O\left(\left(\mu_0 a^2 + a^2 + \mu_0\right)\left(\frac{1}{a^6} + \frac{1}{\varepsilon_{\textit{dist}}^6}\right)\right)$.

More details in paper!

Thanks for Listening!

References I

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. DAGMA: Learning dags via M-matrices and a log-determinant acyclicity characterization. In *Advances in Neural Information Processing Systems*, 2022.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural dag learning. In *International Conference on Learning Representations*, 2020.

Ignavier Ng, Sébastien Lachapelle, Nan Rosemary Ke, Simon Lacoste-Julien, and Kun Zhang. On the convergence of continuous constrained optimization for structure learning. In *International Conference on Artificial Intelligence and Statistics*, pages 8176–8198. PMLR, 2022.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163. PMLR, 2019.

References II

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, 2018.

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.