

# Structure Learning of a Directed Acyclic Graph

Chunlin Li              Ziyue Zhu

April 28, 2019

## 1 Introduction

The directed acyclic graph (DAG) model is useful in statistics, yet imposing a great challenge as the graph dimension grows. The challenge is twofold: statistical guarantee and optimization. The former is addressed in [5, 3]. Here, we focus on the latter: to solve the related optimization problem efficiently.

However, the exact maximization of likelihood for Gaussian DAG is NP hard due to the combinatorial constraint [6]. To circumvent this computational intractability, a smooth relaxation strategy have been proposed by [5].

For a real matrix  $W = (w_{ij}) \in \mathbb{R}^{d \times d}$ , define the binary matrix  $\mathcal{A}(W) \in \{0, 1\}^{d \times d}$  by

$$[\mathcal{A}(W)]_{ij} = 1 \Leftrightarrow w_{ij} \neq 0, \quad (1)$$

and

$$[\mathcal{A}(W)]_{ij} = 0 \Leftrightarrow w_{ij} = 0, \quad (2)$$

which is the adjacency matrix of a directed graph  $G(W)$ . Also define the subset of binary matrices

$$\mathbb{D} = \{B \mid B \in \{0, 1\}^{d \times d} \text{ and } B \text{ is the adjacency matrix of an acyclic graph}\}. \quad (3)$$

Given the data matrix  $X \in \mathbb{R}^{n \times d}$ , in high-dimensional statistics, a sparsity penalization is often imposed,

$$\min_{\mathcal{A}(W) \in \mathbb{D}} \text{loss}(W; X) + \mu \text{pen}(W), \quad (4)$$

where  $\mu$  is a tuning parameter,  $\text{loss}(W; X)$  is some loss function based on the data matrix  $X$  and  $\text{pen}(W)$  is a penalty producing sparse pattern.

Here, we focus on the least-squares (LS) loss, which is equivalent to maximum likelihood estimation in Gaussian model. We replace  $\text{loss}(W; X)$  in (4) by

$$Q(W; X) := \frac{1}{2n} \|X - XW\|_F^2. \quad (5)$$

## 2 Algorithms: $\Lambda$ -Score

In this section, we consider the problem (4) with

$$\text{pen}(W) = \sum_{i=1}^p \sum_{j=1}^p \frac{|w_{ij}|}{\tau} - \max\left(\frac{|w_{ij}|}{\tau} - 1, 0\right). \quad (6)$$

To deal with the condition  $\mathcal{A}(W) \in \mathbb{D}$ , Yuan et al. studied another continuous relaxation, which we refer to as the  $\Lambda$ -constraint.

**Theorem 2** [5]

The adjacency matrix  $W \in \mathbb{R}^{d \times d}$  is a DAG if and only if there exists a matrix  $\Lambda \in \mathbb{R}^{d \times d}$  such that the following constraints are satisfied by  $W$ ,

$$\lambda_{ik} + I(j \neq k) - \lambda_{jk} \geq I(W_{ij} \neq 0), \quad i, j, k = 1, \dots, p, \quad i \neq j, \quad (7)$$

where  $I(\cdot)$  denotes the indicator function.

Based on Theorem 2, a *difference convex* (DC) programming approach can be developed to iteratively relax the nonconvex constraints through a sequence of convex set approximations. Then each convex subproblem is solved by an *alternating direction method of multipliers* (ADMM) [1].

Specifically, decompose  $\text{pen}_\tau$  into a difference of two convex functions,  $\text{pen}_\tau(z) = |z|/\tau - \max(|z|/\tau - 1, 0) \equiv S_1(z) - S_2(z)$ . On this ground, a convex approximation at  $(t+1)$ -th iteration is constructed by replacing  $S_2(z)$  with its affine majorization  $S_2(z_t) + \nabla S_2(z_t)^T(z - z_t)$  at the solution  $z_t$  at  $t$ -th iteration, where  $\nabla S_2(z_t) = \tau^{-1} \text{sign}(z_t) I(|z_t| > \tau)$  is a subgradient of  $S_2$  at  $z_t$ . This leads to a convex subproblem at the  $(t+1)$ -th iteration,

$$\begin{aligned} \min_{(W, \Lambda) \in \mathbb{R}^{d \times d} \times \mathbb{R}^{d \times d}} \quad & Q(W; X) + \mu \tau^{-1} |B \circ W|_1, \\ \text{s.t.} \quad & \lambda_{jk} + I(i \neq k) - \lambda_{ik} \geq \tau^{-1} |W_{ij}| B_{ij} + (1 - B_{ij}), \\ & i, j, k = 1, \dots, p, \quad i \neq j, \end{aligned} \quad (8)$$

where  $B = B_t = H_\tau(W_t)$  and  $H$  is the elementwise hard  $\tau$ -threshold function.

To solve (8), we separate the differentiable from non-differentiable parts there by introducing a decoupling matrix  $V$  for  $W$ , in addition to slack variables  $\xi = (\xi_{ijk})_{p \times p \times p}$  to convert inequality to equality constraints. This yields

$$\begin{aligned} \min_{(W, V, \Lambda, \xi)} \quad & Q(W; X) + \mu\tau^{-1}|B \circ V|_1, \\ \text{s.t.} \quad & W - V = 0, \\ & |V_{ij}|B_{ij} + \tau(1 - B_{ij}) + \xi_{ijk} - \tau\lambda_{jk} - \tau I(i \neq k) + \tau\lambda_{ik} = 0, \\ & \xi_{ijk} \geq 0, \quad i, j, k = 1, \dots, p, i \neq j. \end{aligned} \quad (9)$$

It seems unclear whether the second constraint (7) can be handled by the proximal operator. Alternatively, we follow [1] and introduce scaled dual variables  $\alpha = (\alpha_{ijk})_{p \times p \times p}$  and  $Z = (z_{ij})_{p \times p}$ . This leads to an augmented Lagrangian,

$$\begin{aligned} L^\rho(W, V, \Lambda, \xi, \alpha, Z) = & Q(W; X) + \mu\tau^{-1}|B \circ V|_1 + \frac{\rho}{2}\|W - V + Z\|_F^2 \\ & + \frac{\rho}{2} \sum_k \sum_{i \neq j} (|V_{ij}|B_{ij} + \tau(1 - B_{ij}) + \xi_{ijk} - \tau\lambda_{jk} - \tau I(i \neq k) + \tau\lambda_{ik} + \alpha_{ijk})^2, \end{aligned} \quad (10)$$

where the minimization is solved iteratively. Specifically, at  $(s+1)$ -th iteration of ADMM, update the following steps:

$$\begin{aligned} W_{s+1} &= \arg \min_W L^\rho(W, V_s, \Lambda_s, \xi_s, \alpha_s, Z_s), \\ V_{s+1} &= \arg \min_V L^\rho(W_{s+1}, V, \Lambda_s, \xi_s, \alpha_s, Z_s), \\ \Lambda_{s+1} &= \arg \min_\Lambda L^\rho(W_{s+1}, V_{s+1}, \Lambda, \xi_s, \alpha_s, Z_s), \\ \xi_{s+1} &= \arg \min_{\xi_{ijk} \geq 0} L^\rho(W_{s+1}, V_{s+1}, \Lambda_{s+1}, \xi, \alpha_s, Z_s), \\ (\alpha_{s+1})_{ijk} &= ((\alpha_s)_{ijk} + |(V_s)_{ij}|B_{ij} + \tau(1 - B_{ij}) + (\xi_s)_{ijk} - \tau(\lambda_s)_{ik} - \tau I(j \neq k) + \tau(\lambda_s)_{jk})^+, \\ Z_{s+1} &= Z_s + W_{s+1} - V_{s+1}. \end{aligned} \quad (11)$$

The ADMM updating scheme has analytic formulas which greatly facilitate computation.

### **$\Lambda$ -Scoring Method [5, 3]**

Initiate an estimate  $(W_0, \Lambda_0)$  satisfying (7). Set  $B_0 = H_\tau(W_0)$  and set the

optimization accuracy  $\epsilon > 0$ , for  $t = 1, \dots, \infty$ ,

(a) Compute  $(W_t, \Lambda_t)$  by ADMM (11). Set  $B_t = H_\tau(W_t)$ .

(b) If  $B_t$  has a cycle, for each  $|(W_t)_{ij}| > 0$  in increasing order, if  $(i, j)$  is in a cycle,

$$(W_t)_{ij} \leftarrow 0, \quad (B_t)_{ij} \leftarrow 0.$$

**Remark** (1) For the convergence of ADMM, we use the stopping criteria (3.12) of [1]. (2) The step (b) is implemented in addition to the original algorithm of [5], which ensures that  $W_t$  satisfies the acyclicity condition by removing the weakest edge in an existing cycle, hence that it yields a DAG. Based on our limited numerical experience [3], this modification enhances the overall performance in structure learning. In (b), the cycle detection algorithm is based on the *depth-first search* [2].

### 3 Score-based Inference

For the development of inference theory, see [3].

### References

- [1] Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine learning*, 3(1), 1-122.
- [2] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to Algorithms*. MIT press.
- [3] Li, C., Shen, X., & Pan, W. (2019) Likelihood ratio tests of a large directed acyclic graph. Submitted.
- [4] Shen, X., Pan, W., & Zhu, Y. (2012). Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497), 223-232.
- [5] Yuan, Y., Shen, X., Pan, W., & Wang, Z. (2018). Constrained likelihood for reconstructing a directed acyclic Gaussian graph. *Biometrika*, 106(1), 109-125.
- [6] Zheng, X., Aragam, B., Ravikumar, P. K., & Xing, E. P. (2018). DAGs with NO TEARS: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, pp. 9472-9483.