Optimization for Structure Learning of a Directed Acyclic Graph

Chunlin Li Ziyue Zhu

December 2018

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 Yuan et al. (2018). This project focuses 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 Zheng et al. (2018). To circumvent this computational intractability, two different smooth relaxation strategies have been proposed (Yuan et al., 2018; Zheng et al., 2018), resulting in two different algorithms. In this project, our goal is to examine, compare, and improve the algorithms for structure learning and inference problems.

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

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

and

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

which is the adjacency matrix of a directed graph $G(W)$.

We define the subset of binary matrices

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

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

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

where μ 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.

In this report we focus on the least-squares (LS) loss. We replace $\text{loss}(W;X)$ in (4) by

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

2 Algorithms

2.1 NO TEARS

In this subsection we introduce the method DAGs with Non-combinatorial Optimization via Trace Exponential and Augmented lagRangian for Structure learning (NO TEARS) (Zheng et al., 2018). The following is a summary of their work.

Zheng et al. (2018) consider the penalty

$$\text{pen}(W) = |W|_1 = ||\text{vec}(W)||_1, \tag{6}$$

and discuss learning a *sparse* DAG via minimizing the following regularized score function

$$F(W) := Q(W; X) + \mu|W|_1. \tag{7}$$

In this subsection, we only consider the objective function without the penalty term by setting $\mu = 0$ in (7). Then *score-based learning* is the following nonconvex optimization problem

$$\min_{G(W)\in\mathbb{D}} Q(W; X). \tag{8}$$

Since the constraint is combinatorial, it is hard to enforce it. To solve this issue, they proposed a new method by constructing a smooth function $h : \mathbb{R}^{d\times d} \to \mathbb{R}$ satisfying

$$h(W) = 0 \Leftrightarrow G(W) \in \mathbb{D}. \tag{9}$$

Then if h exists, (8) is equivalent to

$$\min_{W\in\mathbb{R}^{d\times d}} Q(W; X) \quad \text{subject to } h(W) = 0. \tag{10}$$

The existence of h is guaranteed by the following theorem.

Theorem (Zheng et al., 2018)

A matrix $W \in \mathbb{R}^{d\times d}$ is a DAG if and only if

$$h(W) = \text{tr } \exp(W \circ W) - d = 0. \tag{11}$$

Furthermore, the gradient of h is

$$\nabla h(W) = [\text{tr } (\exp(W \circ W)]^T \circ 2W, \tag{12}$$

and satisfies
(1) $h(W) = 0 \Leftrightarrow W$ is acyclic;
(2) The values of h quantify the "DAG-ness" of the graph;
(3) h is smooth;
(4) h and its derivatives are easy to compute.

Since the feasible set is nonconvex, we cannot find an efficient projection operator. Therefore, the commonly used *projected gradient descent* (PGD) cannot be applied here. Instead, we solve (10) by adding a quadratic penalty

$$\min_{W\in\mathbb{R}^{d\times d}} Q(W;X)+\frac{\rho}{2}|h(W)|^2 \quad \text{subject to } h(W)=0. \tag{13}$$

Writing the augmented Lagrarian with dual variable α

$$L^\rho(W,\alpha)=Q(W;X)+\frac{\rho}{2}|h(W)|^2+\alpha h(W), \tag{14}$$

we now need to find a local solution to the dual problem

$$\max_{\alpha\in\mathbb{R}} D(\alpha), \quad D(\alpha):=\min_{W\in\mathbb{R}^{d\times d}} L^\rho(W,\alpha). \tag{15}$$

Fixing α, suppose W_α^* is a local solution

$$W_\alpha^*=\arg\min_{W\in\mathbb{R}^{d\times d}} L^\rho(W,\alpha), \tag{16}$$

which is a unconstrained smooth minimization problem and can be efficiently solved by using methods including the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm (Byrd et al., 1995). Suppose we have W_α^* and now we want to solve (15). Since

$$\nabla D(\alpha)=h(W_\alpha^*), \tag{17}$$

we can perform the dual gradient ascent

$$\alpha\leftarrow\alpha+\rho h(W_\alpha^*). \tag{18}$$

We write down the algorithm.

DAGs with NO TEARS Algorithm (Zheng et al., 2018)

Setting progress rate $c\in(0,1)$, initial value (W_0,α_0), tolerance $\epsilon>0$ and threshold $\omega>0$, for $t=0,...,\infty$,
(a) Solve primal $W_{t+1}\leftarrow \text{argmin}_W L^\rho(W,\alpha_t)$ with ρ such that $h(W_{t+1})<ch(W_t)$.
(b) Dual ascent $\alpha_{t+1}\leftarrow\alpha_t+\rho h(W_{t+1})$.
(c) If $h(W_{t+1})<\epsilon$, stop and set $\tilde{W}=W_{t+1}$. Our solution is $\hat{W}=\tilde{W}\circ 1(|\tilde{W}|>\omega)$.

2.2 Λ-Scoring Method

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

$$\text{pen}(W)=\sum_{i=1}^{p}\sum_{j=1}^{p}\frac{|w_{ij}|}{\tau}-\max\left(\frac{|w_{ij}|}{\tau}-1,0\right). \tag{19}$$

To deal with the condition $\mathcal{A}(W) \in \mathbb{D}$, Yuan et al. studied another continuous relaxation, which we refer to as the Λ-constraint Yuan et al. (2018).

Theorem 2 (Yuan et al., 2018)
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,

$$\lambda_{ik} + I(j \neq k) - \lambda_{jk} \geq I(W_{ij} \neq 0), \quad i, j, k = 1, \ldots, p,\ i \neq j, \tag{20}$$

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) (Boyd et al., 2011).

Specifically, decompose pen_τ 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,

$$\begin{aligned} &\min_{(W,\Lambda)\in\mathbb{R}^{d\times d}\times\mathbb{R}^{d\times d}} Q(W; X) + \mu\tau^{-1}|B \circ W|_1, \\ &\quad \text{s.t.} \quad \lambda_{jk} + I(i \neq k) - \lambda_{ik} \geq \tau^{-1}|W_{ij}|B_{ij} + (1 - B_{ij}), \quad i, j, k = 1, \ldots, p, i \neq j, \end{aligned} \tag{21}$$

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

To solve (21), 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

$$\begin{aligned} &\min_{(W,V,\Lambda,\xi)} \quad Q(W; X) + \mu\tau^{-1}|B \circ V|_1, \\ &\quad \text{s.t.} \quad W - V = 0, \quad |V_{ij}|B_{ij} + \tau(1 - B_{ij}) + \xi_{ijk} - \tau\lambda_{jk} - \tau I(i \neq k) + \tau\lambda_{ik} = 0, \\ &\qquad \xi_{ijk} \geq 0, \quad i, j, k = 1, \ldots, p, i \neq j. \end{aligned} \tag{22}$$

It seems unclear if the second constraint (20) can be handled by the proximal operator. Alternatively, we follow Boyd et al. (2011) 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,

$$\begin{aligned} &L^\rho(W, V, \Lambda, \xi, \alpha, Z) = Q(W; X) + \mu\tau^{-1}|B \circ V|_1 + \frac{\rho}{2}\|W - V + Z\|_F^2 \\ &+ \frac{\rho}{2}\sum_k \sum_{i\neq j} \left(|V_{ij}|B_{ij} + \tau(1 - B_{ij}) + \xi_{ijk} - \tau\lambda_{jk} - \tau I(i \neq k) + \tau\lambda_{ik} + \alpha_{ijk}\right)^2, \end{aligned} \tag{23}$$

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

$$\begin{aligned}
W_{s+1} &= \arg\min_{W} L^{\rho}(W, V_s, \Lambda_s, \xi_s, \alpha_s, Z_s), \\
V_{s+1} &= \arg\min_{V} L^{\rho}(W_{s+1}, V, \Lambda_s, \xi_s, \alpha_s, Z_s), \\
\Lambda_{s+1} &= \arg\min_{\Lambda} L^{\rho}(W_{s+1}, V_{s+1}, \Lambda, \xi_s, \alpha_s, Z_s), \\
\xi_{s+1} &= \arg\min_{\xi_{ijk}\geq 0} L^{\rho}(W_{s+1}, V_{s+1}, \Lambda_{s+1}, \xi, \alpha_s, Z_s), \\
(\alpha_{s+1})_{ijk} &= \left((\alpha_s)_{ijk} + |(V_s)_{ij}|B_{ij} + \tau(1-B_{ij}) + (\xi_s)_{ijk} - \tau(\lambda_s)_{ik} - \tau I(j\neq k) + \tau(\lambda_s)_{jk}\right)^{+}, \\
Z_{s+1} &= Z_s + W_{s+1} - V_{s+1}.
\end{aligned} \tag{24}$$

The ADMM updating scheme has analytic formulas which greatly facilitate computation; see the attached program code.

Λ-Scoring Method Yuan et al. (2018)
Initiate an estimate (W_0, Λ_0) satisfying (20). Set $B_0 = H_\tau(W_0)$ and set the optimization accuracy $\epsilon > 0$, for $t = 1, \ldots, \infty$,
(a) Compute (W_t, Λ_t) by ADMM (24). 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 Boyd et al. (2011).
(2) The step (b) is implemented in addition to the original algorithm of Yuan et al. (2018), 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, this modification enhances the overall performance in structure learning. In (b), the cycle detection algorithm is based on the *depth-first search* (Cormen et al., 2001).

3 Simulation Study

In this section, we examine NOTEARS and Λ-scoring method using the simulated data.

3.1 Simulation Setting

For simulation study, we set the graph dimension $p = 20$ and sample size $n = 1000$. Two types of DAG are considered:
(a) Random graph. For all $i < j$, let $B^*_{ij} = 1$ with probability 0.15 and zero otherwise. Next, create the adjacency matrix W by $P(W^*_{ij} = 1) = P(W^*_{ij} = -1) = 0.5$ for $B^*_{ij} = 1$.
(b) Hub graph. Let $B^*_{i1} = 1$ for $i = 2, \ldots, p$ and create W by $P(W^*_{ij} = 1) = P(W^*_{ij} = -1) = 0.5$ for $B^*_{ij} = 1$.

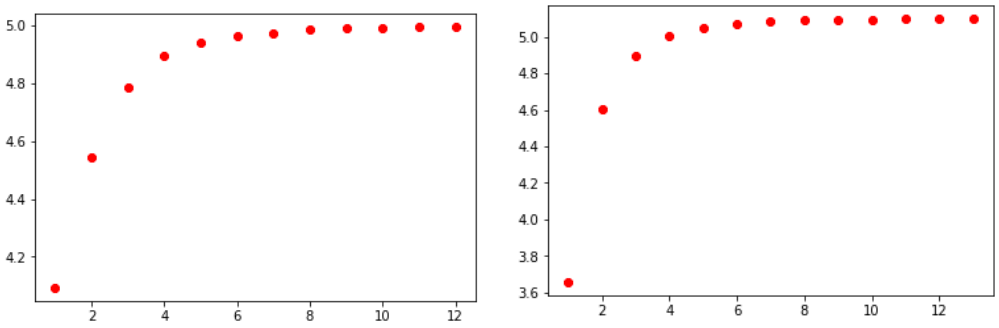

Figure 1: NO TEARS a (left) and b (right).

Based on the adjacency matrix W, define $\Sigma = (I - W)^{-1}(I - W)^{-T}$. Then we generate data $x_1, \ldots, x_n \overset{iid}{\sim} N(0, \Sigma)$.

For evaluation, we plot the objective values against the iterations. Statistically, we consider the oracle rate (ORR), i.e. the rate of correct recoveries of the whole truth graph W^*.

We use $W_0 = 0_{d\times d}$ which is a $d \times d$ matrix with all elements equal to 0 and $\alpha_0 = 0$ as the initial estimates and set progress rate $c = 0.25$ for the NOTEARS algorithm. We use $W_0 = 0_{d\times d}$ and $\Lambda_0 = 1_{d\times d}$ which is a $d \times d$ matrix with all elements equal to 1 for the Λ-scoring method. For NOTEARS, we used the Python code from (Zheng et al., 2018). The default setting of threshold parameter in NOTEARS is 0.3. Hence, For the Λ-scoring method, we wrote R code and set the threshold parameter $\tau = 0.3$ and $\mu = 0$ (no penalty) for fair comparison.

3.2 Results

Figure 1 shows the plots of the iterations using NO TEARS based on one dataset generated by (a) and one dataset generated by (b), respectively. The objective function is the dual function, so its value increases as the iteration number increases. We can see that the algorithm converges in 12 iterations based on DAG type (a) and 13 iterations based on DAG type (b).

Figure 2 shows the plots of the iterations using Λ-scoring based on one dataset generated by (a) and one dataset generated by (b), respectively. The Λ-scoring method converges after several DC iterations, which verifies the *finite termination property* (Shen et al., 2012). Note that this is generally not true for nonconvex and nonsmooth optimization.

Finally, Table 1 shows the oracle rate of the methods. Both performed reasonably well in large sample setting.

Regarding computing speed, Λ-scoring seems faster than NOTEARS.

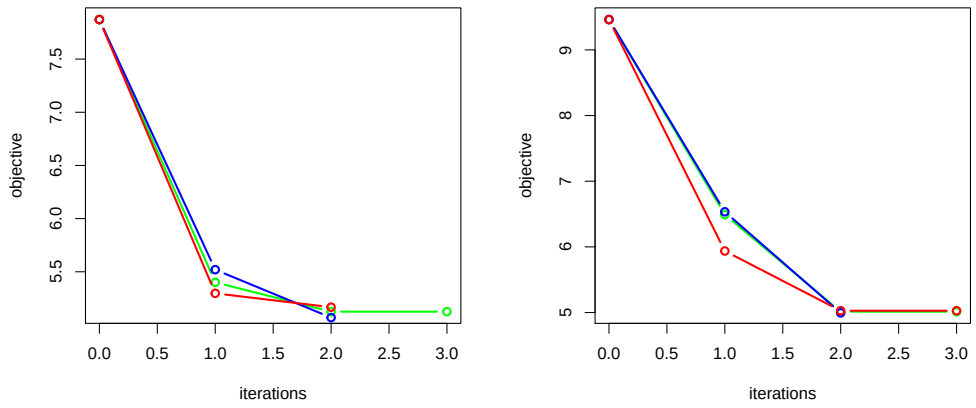

Figure 2: Λ-scoring a (left) and b (right): red $\rho = 0.5$; green $\rho = 1$; blue $\rho = 2$

	NOTEARS	Λ-scoring
a	100	100
b	100	100

Table 1: The oracle rate (%) of two methods in a and b of 100 simulation runs.

4 Discussion

In this report, we explored two optimization algorithms for structure learning of a DAG. Two methods are based on different continuous relaxation of the acyclic condition.

Although finding a global optimum is never guaranteed, both methods converges to a stationary point under the regularity conditions (Yuan et al., 2018; Zheng et al., 2018). The simulation study illustrates the operating characteristics of the algorithms and shows that both are able to learn a DAG reasonably well.

However, the numerical experience shows that the algorithms can hardly handle the problem of dimension $d > 1000$. The efficient method for large-scale data remains in exploration.

References

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. The MIT press, 2001.

X. Shen, W. Pan, and Y. Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of American Statistical Association*, 107:223–232, 2012.

Yiping Yuan, Xiaotong Shen, Wei Pan, and Zizhuo Wang. Constrained likelihood for reconstructing a directed acyclic gaussian graph. *Biometrika*, 2018.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P Xing. Dags with no tears: Smooth optimization for structure learning. *arXiv preprint arXiv:1803.01422*, 2018.