On the Convergence of Continuous Constrained Optimization for Structure Learning

Ignavier Ng[1]*, Sébastien Lachapelle[2], Nan Rosemary Ke[2], Simon Lacoste-Julien[2,3]
[1] University of Toronto
[2] Mila, Université de Montréal
[3] Canada CIFAR AI Chair

Abstract

Structure learning of directed acyclic graphs (DAGs) is a fundamental problem in many scientific endeavors. A new line of work, based on NOTEARS (Zheng et al., 2018), reformulates the structure learning problem as a continuous optimization one by leveraging an algebraic characterization of DAG constraint. The constrained problem is typically solved using the augmented Lagrangian method (ALM) which is often preferred to the quadratic penalty method (QPM) by virtue of its convergence result that does not require the penalty coefficient to go to infinity, hence avoiding ill-conditioning. In this work, we review the standard convergence result of the ALM and show that the required conditions are not satisfied in the recent continuous constrained formulation for learning DAGs. We demonstrate empirically that its behavior is akin to that of the QPM which is prone to ill-conditioning, thus motivating the use of second-order method in this setting. We also establish the convergence guarantee of QPM to a DAG solution, under mild conditions, based on a property of the DAG constraint term.

1 Introduction

Structure learning of directed acyclic graphs (DAGs) is a fundamental problem in many scientific endeavors such as biology (Sachs et al., 2005) and economics (Koller and Friedman, 2009). Traditionally, score-based structure learning methods cast the problem into a discrete optimization program using a predefined score function. Most of these methods, such as GES (Chickering, 2002), involve local heuristics owing to the large search space of graphs (Chickering, 1996; He et al., 2015).

A recent line of work, based on NOTEARS (Zheng et al., 2018), reformulates the problem of score-based structure learning as a continuous constrained optimization problem. At the heart of the proposed method is an algebraic characterization of acyclicity, which is used to minimize the least square objective for learning linear DAGs while enforcing acyclicity. Various works have extended this idea to support nonlinear models (Kalainathan et al., 2018; Yu et al., 2019; Ng et al., 2019a,b; Lachapelle et al., 2020; Zheng et al., 2020), time series data (Pamfil et al., 2020), incomplete data (Wang et al., 2020), latent variables (Zeng et al., 2020), unmeasured confounding (Bhattacharya et al., 2020), interventional data (Brouillard et al., 2020; Ke et al., 2020) and flexible score functions (Zhu et al., 2020). On the application side, similar continuous constrained formulations have been adopted in other domains such as transfer learning (Pruthi et al., 2020), normalizing flows (Wehenkel and Louppe, 2020) and disentangled representation learning (Yang et al., 2020; Moraffah et al., 2020).

Like in the original work, most of these extensions rely on the *augmented Lagrangian method* (ALM) (Bertsekas, 1982, 1999) to solve the continuous constrained optimization problem. This choice of

*Work was done during an internship at Mila.

Causal Discovery & Causality-Inspired Machine Learning Workshop at Neural Information Processing Systems, 2020.

algorithm was originally motivated by the fact that the convergence result of ALM, under some regularity conditions, does not require increasing the penalty coefficient to infinity (Zheng et al., 2018), as is the case in the classical *quadratic penalty method* (QPM) (Powell, 1969; Fletcher, 1987), hence avoiding ill-conditioning that may lead to numerical difficulties. However, it remains unclear whether the continuous constrained formulation proposed by Zheng et al. (2018) satisfies the regularity conditions. Furthermore, to the best of our knowledge, none of the existing works have formally studied whether the continuous constrained formulation is guaranteed to converge to a DAG solution, which is a key to structure learning problems.

Contributions. In this work, we take a closer look at the properties of the continuous constrained optimization for learning DAGs and conclude that: (i) the conditions behind standard convergence result of the ALM are not satisfied, (ii) consequently the ALM behaves just like the QPM on this problem which requires increasing the penalty coefficient to infinity, and (iii) recognizing this provides important insight that translates into practical improvement. We also establish the convergence guarantee of QPM to a DAG solution, under mild conditions.

Although developed independently, our work is closely related to the analysis by Wei et al. (2020), both of which investigate the regularity conditions of the continuous constrained formulation. The main difference is that Wei et al. (2020) focused on the general KKT optimality conditions, while our study focuses on the convergence of specific optimization procedures (i.e., ALM and QPM).

2 Background

We provide a brief review of score-based structure learning, the NOTEARS method (Zheng et al., 2018) and the standard convergence result of ALM.

2.1 Score-Based Structure Learning

Structure learning refers to the problem of learning a graphical structure (in our case a DAG, also known as a Bayesian network) from data. Given a set of random variables $X = (X_1, \dots, X_d)$, we assume that the corresponding design matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ is generated from a distribution $P(X)$ (with a density $p(x)$) that is Markov with respect to a ground truth DAG $\mathcal{G}$, or equivalently, can be factorized as $p(x) = \prod_{i=1}^{d} p_i(x_i | x_{\mathsf{PA}_i^{\mathcal{G}}})$, where $\mathsf{PA}_i^{\mathcal{G}}$ designates the set of parents of X_i in $\mathcal{G}$. Under further assumptions, the DAG $\mathcal{G}$ is identifiable from the distribution $P(X)$, such as the linear non-Gaussian acyclic model (Shimizu et al., 2006) and linear Gaussian model with equal noise variances (Peters and Bühlmann, 2013a). In this work we focus on the latter case.

Under *score-based* methods, structure learning can be formulated as an optimization problem over the space of graphs (Peters et al., 2017) using some goodness-of-fit measure with a sparsity penalty term. Some examples of score-based methods are GES (Chickering, 2002), methods based on integer linear programming (Jaakkola et al., 2010; Cussens, 2011) and CAM (Bühlmann et al., 2014). Most of these methods tackle the structure search problem in its natural discrete form.

2.2 NOTEARS: Continuous Constrained Optimization for Structure Learning

NOTEARS (Zheng et al., 2018) adopts a continuous constrained formulation for score-based learning of linear DAGs. In particular, the directed graph is encoded as a *weighted adjacency matrix* $B \in \mathbb{R}^{d\times d}$ that represents the coefficients of a linear structural equation model (SEM), i.e., $X = B^{\mathsf{T}} X + N$, where N is a noise vector with independent entries. The authors have showed that $\operatorname{tr}(e^{B\circ B}) - d = 0$ holds if and only if B represents a DAG. Since the ground truth is assumed to be acyclic, we have the following constrained optimization problem

$$\min_{B\in\mathbb{R}^{d\times d}} \ \frac{1}{2n}\|\mathbf{X} - \mathbf{X}B\|_F^2 + \lambda\|B\|_1 \ \text{ subject to } \ \operatorname{tr}(e^{B\circ B}) - d = 0, \tag{1}$$

where $\frac{1}{2n}\|\mathbf{X} - \mathbf{X}B\|_F^2$ is the least squares objective and is equal, up to a constant, to the log-likelihood function of linear Gaussian DAGs assuming equal noise variances, and $\|B\|_1$ denotes the ℓ_1 penalty term defined element-wise. To solve the constrained problem (1), NOTEARS relies on the ALM, a general method for continuous constrained optimization, which we review next.

2.3 Augmented Lagrangian Method

Consider the following generic constrained optimization problem

$$\min_{\theta \in \mathbb{R}^m} \; f(\theta) \;\text{ subject to }\; h(\theta) = 0, \tag{2}$$

where $f : \mathbb{R}^m \to \mathbb{R}$ and $h : \mathbb{R}^m \to \mathbb{R}^p$ are both twice continuously differentiable.

The ALM transforms a constrained optimization problem like (2) into a sequence of unconstrained problems with solutions converging to a solution of the original problem. At the heart of this approach is the augmented Lagrangian function defined as

$$L(\theta, \alpha; \rho) = f(\theta) + \alpha^{\mathsf{T}} h(\theta) + \frac{\rho}{2}\|h(\theta)\|^2,$$

where $\alpha \in \mathbb{R}^p$ is an estimate of the Lagrange multiplier, ρ is the penalty coefficient, and $\|\cdot\|$ denotes the Euclidean norm. A typical update rule (Bertsekas, 1999) reads

$$\theta^{k+1} = \underset{\theta \in \mathbb{R}^m}{\arg\min}\, L(\theta, \alpha^k; \rho^k), \tag{3}$$

$$\alpha^{k+1} = \alpha^k + \rho^k h(\theta^{k+1}), \tag{4}$$

$$\rho^{k+1} = \begin{cases} \beta\rho^k & \text{if } \|h(\theta^{k+1})\| > \gamma\|h(\theta^k)\|, \\ \rho^k & \text{otherwise,} \end{cases}$$

where $\beta > 1$ and $\gamma < 1$ are hyperparameters. Note that problem (3) can sometimes be solved only to stationarity, if, for example, the set of feasible points is not convex, as is the case in the NOTEARS formulation (1).

Based on the procedure of ALM outlined above, we review one of its standard convergence results (Bertsekas, 1982, 1999; Nocedal and Wright, 2006). The following definition is required to state this result and is crucial to the contribution of our work.

Definition 1 (Regular Point). *We say that a point θ^* is regular, or that it satisfies the linear independence constraint qualification (LICQ) if the Jacobian matrix of h evaluated at θ^*, $\nabla_\theta h(\theta^*) \in \mathbb{R}^{p\times m}$, has full rank.*

Theorem 1 (Nocedal and Wright (2006, Theorem 17.5 & 17.6)). *Let θ^* be a regular point of (2) that satisfies the second-order sufficient conditions (see Appendix A) with vector α^*. Then there exist positive scalars $\bar{\rho}$ (sufficiently large), δ, ϵ, and M such that the following claims hold:*

(a) For all α_k and ρ_k satisfying

$$\|\alpha_k - \alpha^*\| \le \rho_k\delta, \quad \rho_k \ge \bar{\rho}, \tag{5}$$

the problem

$$\min_{\theta \in \mathbb{R}^m} \; L(\theta, \alpha_k; \rho_k) \;\text{ subject to }\; \|\theta - \theta^*\| \le \epsilon$$

has a unique solution θ_k. Moreover, we have

$$\|\theta_k - \theta^*\| \le M\|\alpha_k - \alpha^*\|/\rho_k. \tag{6}$$

(b) For all α_k and ρ_k that satisfy (5), we have

$$\|\alpha_{k+1} - \alpha^*\| \le M\|\alpha_k - \alpha^*\|/\rho_k\,, \tag{7}$$

where α_{k+1} is given by Eq. (4).

An important consequence of Theorem 1 is that if ρ_k is larger than both $\bar{\rho}$ and $\|\alpha_k - \alpha^*\|/\delta$, then inequalities (6) and (7) hold. If, in addition, $\rho_k > M$, then $\alpha_k \to \alpha^*$ by (7) and $\theta_k \to \theta^*$ by (6), *without increasing ρ_k to infinity*. This property often motivates the usage of the ALM over the other approaches for constrained optimization. For example, the classical QPM requires bringing the penalty coefficient ρ_k to infinity, which may lead to ill-conditioning issues when solving (3). This was the original motivation given by Zheng et al. (2018) and its extensions for using the ALM. In this work, we show that this standard result does not apply in the context of learning DAGs and that, as a consequence, the ALM behaves similarly to the QPM, which we present in Section 3.2.

3 Convergence of the Continuous Constrained Optimization Problem

In this section, we take a closer look at the continuous constrained optimization for learning DAGs in the context of ALM and QPM. With a slight abuse of notation, we consider the following problem

$$\min_{B \in \mathbb{R}^{d \times d}} \ f(B) \ \text{ subject to } \ h(B) = 0, \tag{8}$$

where $f(B) = \frac{1}{2n}\|\mathbf{X} - \mathbf{X}B\|_F^2$ and $h(B)$ is a (scalar-valued) constraint term that enforces acyclicity on B. We will consider two acyclicity constraints proposed in the literature:

$$h_{\text{exp}}(B) = \operatorname{tr}(e^{B \circ B}) - d \qquad \text{(Zheng et al., 2018)},$$
$$h_{\text{bin}}(B) = \operatorname{tr}\left[(I + cB \circ B)^d\right] - d, \ c > 0 \qquad \text{(Yu et al., 2019)}.$$

To simplify our analysis, we ignore the sparsity penalty term in the objective of Zheng et al. (2018) to ensure that it is differentiable. Note that in Section 4, we study empirically the continuous constrained formulation under less restrictive assumptions.

3.1 Regularity of DAG Constraint Term

To investigate whether the key benefits of ALM illustrated by Theorem 1 apply for problem (8), one has to first verify if the DAG constraint term $h(B)$ satisfies the regularity conditions. The following condition is required for our analysis.

Assumption 1. *The function $h(B) = 0$ if and only if its gradient $\nabla_B h(B) = 0$.*

Both DAG constraint terms proposed by Zheng et al. (2018) and Yu et al. (2019) satisfy the assumption above, with a proof provided in Appendix B.

Proposition 1. *The functions $h_{exp}(B)$ and $h_{bin}(B)$ satisfy Assumption 1.*

Assumption 1 implies that the Jacobian matrix of function $h(B)$ (after reshaping) evaluated at any feasible point of problem (8) corresponds to a zero row vector, which, therefore, does not have full rank and leads to the following remark.

Remark 1. *If the function $h(B)$ satisfies Assumption 1, any feasible solution of problem (8) is not regular and Theorem 1 does not apply.*

With Proposition 1, this shows that the benefits of ALM (demonstrated by Theorem 1) do not apply for the DAG constraints developed recently, i.e., by Zheng et al. (2018) and Yu et al. (2019). In this case, the only convergence option for ALM is by increasing the penalty coefficient ρ to infinity, whose behavior is similar to the QPM outlined in the next subsection.

3.2 Quadratic Penalty Method

We now describe the procedure of QPM for solving the constrained optimization problem (8). The quadratic penalty function is given by

$$Q(B; \rho) = f(B) + \frac{\rho}{2} h(B)^2, \tag{9}$$

with a complete procedure described in Algorithm 1.

This approach adds a quadratic penalty term for the constraint violation to the objective function $f(B)$. By gradually increasing the penalty coefficient ρ, we penalize the constraint violation with increasing severity. Therefore, it makes intuitive sense to think that the procedure converges to a feasible solution (i.e., a DAG solution) as we bring ρ to infinity. However, this is not necessarily true: in the rather general case, Algorithm 1 returns only a stationary point of the quadratic penalty term $h(B)^2$ (Nocedal and Wright, 2006, Theorem 17.2).

Fortunately, if the DAG constraint term $h(B)$ satisfies Assumption 1, the procedure is guaranteed to converge to a feasible solution, under mild conditions, formally stated in Theorem 2. Note that this theorem and its proof are a simple modification of Theorem 17.2 in Nocedal and Wright (2006).

Theorem 2. *Suppose that the penalty coefficients and tolerances in Algorithm 1 satisfy $\rho_k \to \infty$ and $\{\tau_k\}$ being bounded. Suppose also that the function $h(B)$ satisfies Assumption 1. Then every limit point B^* of the sequence $\{B_k\}$ is feasible.*

Algorithm 1 Quadratic Penalty Method (Nocedal and Wright, 2006)

Require: starting penalty coefficient $\rho_0 > 0$; multiplicative factor $\beta > 1$; nonnegative sequence $\{\tau_k\}$; starting point B_0.

for $k = 1, 2, \ldots$ **do**
    Find an approximate minimizer B_k of $Q(\cdot; \rho_k)$, starting at B_{k-1},
        and terminating when $\|\nabla_B Q(B; \rho_k)\|_F \leq \tau_k$
    if final convergence test satisfied **then**
        stop with approximate solution B_k
    end if
    Update penalty coefficient $\rho_{k+1} = \beta \rho_k$
end for

Proof. By differentiating $Q(B; \rho_k)$ in Eq. (9), we obtain

$$\nabla_B Q(B_k; \rho_k) = \nabla_B f(B_k) + \rho_k h(B_k) \nabla_B h(B_k).$$

From the termination criterion of the subproblems in Algorithm 1, we have

$$\|\nabla_B f(B_k) + \rho_k h(B_k) \nabla_B h(B_k)\|_F \leq \tau_k.$$

By rearranging and using the inequality $\|a\|_F - \|b\|_F \leq \|a + b\|_F$, we obtain

$$\|h(B_k) \nabla_B h(B_k)\|_F \leq \frac{1}{\rho_k} \left(\tau_k + \|\nabla_B f(B_k)\|_F \right).$$

Let B^* be a limit point of the sequence of iterates. Then there is a subsequence $\mathcal{K}$ such that $\lim_{k \in \mathcal{K}} B_k = B^*$. When we take limits as $k \to \infty$ for $k \in \mathcal{K}$, the bracketed term on the right-hand-side approaches $\tau^* + \|\nabla_B f(B^*)\|_F$, so because $\rho_k \to \infty$, the right-hand-side approaches zero. From the corresponding limit on the left-hand-side, we obtain

$$h(B^*) \nabla_B h(B^*) = 0,$$

which, by Assumption 1, yields

$$h(B^*) = 0.$$

□

Theorem 2 and Proposition 1 ensure that, using the DAG constraint terms $h_{\text{exp}}(B)$ and $h_{\text{bin}}(B)$, Algorithm 1 will converge to a DAG solution based on inexact minimizations of $Q(\cdot; \rho_k)$, which is a key to structure learning problems. This also explains why the implementations of ALM with these two constraints often return DAG solutions in practice (after a thresholding step). Furthermore, if Assumption 1 is satisfied, Theorem 2 verifies that one could directly use the value of DAG constraint term as an indicator for the final convergence test in Algorithm 1, i.e., $h(B_k) \leq \varepsilon$ with ε being a small tolerance. This has been adopted in the current implementation of NOTEARS (Zheng et al., 2018) and most of its extensions.

Practical issue. Due to the limit of machine precision, the final solution, in practice, may not correspond exactly to a DAG and contain many entries close to zero, as in (Zheng et al., 2018). Hence, one has to eliminate those weights using a thresholding step on the estimated entries. The experiments in Section 4.1 suggest that a small threshold suffices to obtain DAGs. Nevertheless, a moderately large threshold (e.g., 0.3) is still useful to remove false discoveries.

4 Experiments

We conduct experiments on the structure learning task and take a closer look at the optimization process to verify our study in Section 3, showing that ALM behaves similarly to QPM, both of which converge to a DAG solution when the penalty coefficient goes to infinity. We then compare the ability of different optimization algorithms to handle the ill-conditioning issues.

Methods. To demonstrate that our study generalizes to nonlinear cases, we experiment with both NOTEARS (Zheng et al., 2018) and NOTEARS-MLP (Zheng et al., 2020). We also consider their variants with the ℓ_1 penalty term, denoted as NOTEARS-L1 and NOTEARS-MLP-L1, respectively.

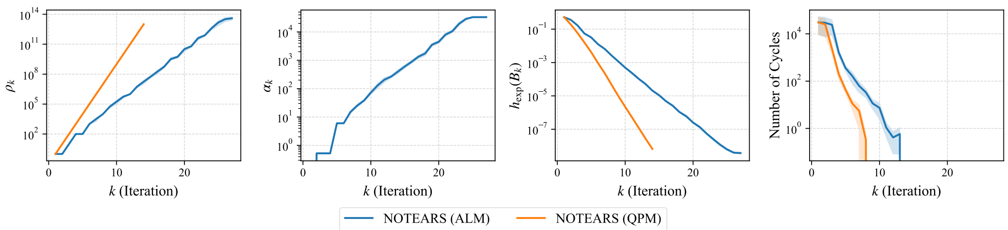


Figure 1: Optimization process of NOTEARS on the linear DAG model using ALM and QPM. Each data point corresponds to the k-th iteration. Shaded area denotes standard errors over 12 trials.

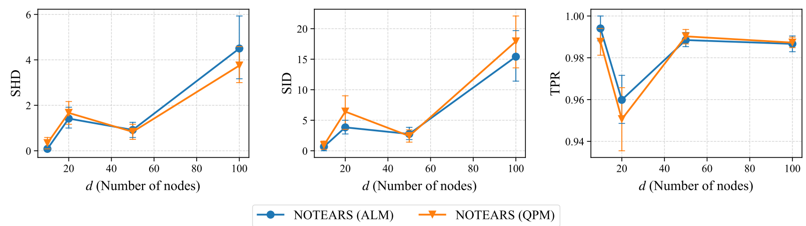


Figure 2: Empirical results of NOTEARS on the linear DAG model using ALM and QPM. Lower is better, except for TPR. Error bars denote standard errors over 12 trials.

Implementations. Our implementations are based on the code[2] released by Zheng et al. (2018). Note, however, that the code implemented the DAG constraint term $h_{\text{bin}}(B)$ proposed by Yu et al. (2019). Here we adopt the original one $h_{\exp}(B)$ to be consistent with existing works. We also use the least squares objective in our experiments. Unless otherwise stated, we employ the L-BFGS algorithm (Byrd et al., 2003) to solve each subproblem and use a threshold of 0.3 for post-processing.

Simulations. We simulate the ground truth DAGs using the *Erdös–Rényi* model (Erdös and Rényi, 1959) with d edges on average. Based on different graph sizes and data generating procedure, we generate 1000 samples with standard Gaussian noise. For NOTEARS and NOTEARS-L1, we simulate the linear DAG model with edge weights independently sampled from $\text{Uniform}\left([-2, -0.5] \cup [0.5, 2]\right)$, similar to (Zheng et al., 2018). For the nonlinear variants NOTEARS-MLP and NOTEARS-MLP-L1, we consider the data generating procedure used by Ng et al. (2019a); Zheng et al. (2020), where each function is sampled from a Gaussian process (GP) with RBF kernel of bandwidth one. Both data models are known to be identifiable (Peters and Bühlmann, 2013a; Peters et al., 2014).

Metrics. We report the structural hamming distance (SHD), structural intervention distance (SID) (Peters and Bühlmann, 2013b) and true positive rate (TPR), averaged over 12 random trials.

4.1 ALM Behaves Similarly to QPM

We conduct experiments to show that ALM behaves similarly to QPM in the context of structure learning, and that both of them converge to a DAG solution. Note that our goal here is not to show that QPM performs better than ALM, but rather to demonstrate that they have similar behavior.

We first take a closer look at the optimization process of ALM and QPM on the 10-node graphs. Figure 1 and 5 depict the penalty coefficient ρ_k, estimate of Lagrange multiplier α_k (only for ALM), value of DAG constraint term $h_{\exp}(B_k)$ and number of cycles in the k-th iteration of the optimization. Note that we use a smaller threshold 0.05 when computing the number of cycles. Consistent with our study in Section 3.1, NOTEARS with ALM requires very large coefficient ρ_k so that $h_{\exp}(B_k)$ is close to zero, similar to QPM. Furthermore, one observes that they both converge to a DAG solution (after thresholding) when ρ_k is very large, which serves as an empirical validation of Theorem 2.

We further investigate whether ALM and QPM yield similar structure learning performance. The results are reported in Figure 2 and 6 with graph sizes $d \in \{10, 20, 50, 100\}$, showing that ALM performs similarly to QPM across all metrics. All these observations appear to generalize to the cases not covered by our analysis in Section 3, including those with nonlinear models and ℓ_1 penalty.

[2] `https://github.com/xunzheng/notears`

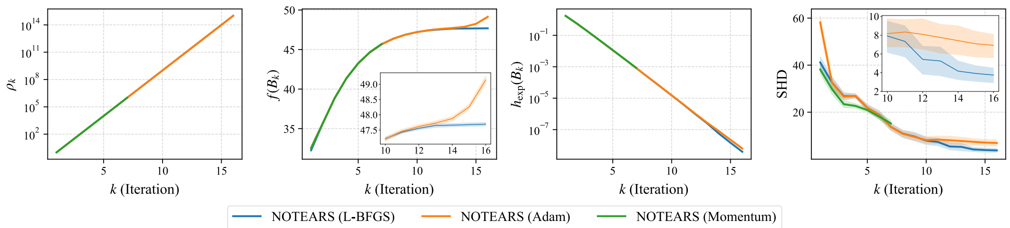


Figure 3: Optimization process of different optimization algorithms for solving the QPM subproblems of NOTEARS. Each data point corresponds to the k-th iteration. Shaded area denotes standard errors over 12 trials. The blue line overlaps with the orange line in the first panel.

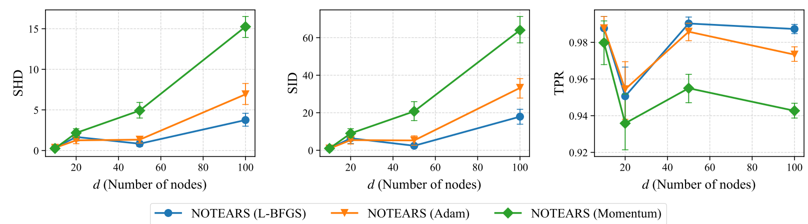


Figure 4: Empirical results of different optimization algorithms for solving the QPM subproblems of NOTEARS. Lower is better, except for TPR. Error bars denote standard errors over 12 trials.

4.2 Different Optimization Algorithms for Handling Ill-Conditioning

The previous experiments demonstrate that ALM behaves similarly to QPM that requires bringing the penalty coefficient to infinity, which is known to cause numerical difficulties and ill-conditioning issues on the objective landscape (Nocedal and Wright, 2006, p. 505). Here we experiment with different optimization algorithms for solving the QPM subproblems of NOTEARS, to investigate which of them handle ill-conditioning better. The optimization algorithms include gradient descent with momentum (Qian, 1999), Adam (Kingma and Ba, 2014) and L-BFGS (Byrd et al., 2003). Note that gradient descent with momentum often terminates earlier because of numerical difficulties, so we report its results right before termination.

In Figure 3 we visualize the optimization process on the 100-node graphs. One first observes that momentum terminates when the coefficient ρ_k reaches 10^7, indicating that it fails to handle ill-conditioning in most cases. Notice that L-BFGS is more stable than Adam for large ρ_k, thus returning solutions with lower objective value $f(B_k)$ and SHD. Similar observations are also made for the overall structure learning performance, as depicted in Figure 4 with graph sizes $d \in \{10, 20, 50, 100\}$. NOTEARS with L-BFGS performs the best across nearly all metrics, followed by Adam. Gradient descent with momentum gives rise to much higher SHD and SID, especially on large graphs. As compared to first-order gradient descent method, this demonstrates that second-order method such as L-BFGS handles ill-conditioning better by incorporating curvature information through approximations of the Hessian matrix, which is consistent with the optimization literature (Bertsekas, 1999). The Adam algorithm, on the other hand, lies in the middle as it employs diagonal rescaling on the parameter space by maintaining running averages of past gradients (Bottou et al., 2018).

5 Conclusion

We took a closer look at the standard convergence result of ALM and showed that the required regularity conditions are not satisfied in the recently proposed continuous constrained optimization for learning DAGs. This implies that it behaves similarly to the QPM that is prone to ill-conditioning, which was verified in our experiments. We then showed theoretically and empirically that QPM guarantees convergence to a DAG solution, under mild conditions. These empirical studies also suggest that our convergence analysis generalizes to the cases with nonlinear models and ℓ_1 penalty. Finally, we conducted experiments to demonstrate that second-order method handles ill-conditioning better in this setting, which may provide practical insight for picking a proper optimization algorithm.

Acknowledgments

This research was partially supported by the Canada CIFAR AI Chair Program, by an IVADO excellence PhD scholarship and by a Google Focused Research award. The authors would like to thank the anonymous reviewers for valuable feedback. Simon Lacoste-Julien is a CIFAR Associate Fellow in the Learning in Machines & Brains program.

References

D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

R. Bhattacharya, T. Nagarajan, D. Malinsky, and I. Shpitser. Differentiable causal discovery under unmeasured confounding. *arXiv preprint arXiv:2010.06978*, 2020.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Rev.*, 60: 223–311, 2018.

P. Brouillard, S. Lachapelle, A. Lacoste, S. Lacoste-Julien, and A. Drouin. Differentiable causal discovery from interventional data, 2020.

R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16, 2003.

P. Bühlmann, J. Peters, and J. Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014.

D. M. Chickering. Learning Bayesian networks is NP-complete. In *Learning from Data: Artificial Intelligence and Statistics V*. Springer, 1996.

D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002.

J. Cussens. Bayesian network learning with cutting planes. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, 2011.

P. Erdös and A. Rényi. On random graphs I. *Publicationes Mathematicae*, 6:290–297, 1959.

R. Fletcher. *Practical Methods of Optimization*. Wiley-Interscience, 1987.

Y. He, J. Jia, and B. Yu. Counting and exploring sizes of Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 16:2589–2609, 2015.

T. Jaakkola, D. Sontag, A. Globerson, and M. Meila. Learning Bayesian Network Structure using LP Relaxations. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010.

D. Kalainathan, O. Goudet, I. Guyon, D. Lopez-Paz, and M. Sebag. Structural agnostic modeling: Adversarial learning of causal graphs. *arXiv preprint arXiv:1803.04929*, 2018.

N. R. Ke, O. Bilaniuk, A. Goyal, S. Bauer, H. Larochelle, B. Schölkopf, M. C. Mozer, C. Pal, and Y. Bengio. Learning neural causal models from unknown interventions, 2020.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009.

S. Lachapelle, P. Brouillard, T. Deleu, and S. Lacoste-Julien. Gradient-based neural DAG learning. In *International Conference on Learning Representations*, 2020.

R. Moraffah, B. Moraffah, M. Karami, A. Raglin, and H. Liu. Causal adversarial network for learning conditional and interventional distributions. *arXiv preprint arXiv:2008.11376*, 2020.

I. Ng, Z. Fang, S. Zhu, Z. Chen, and J. Wang. Masked gradient-based causal structure learning. *arXiv preprint arXiv:1910.08527*, 2019a.

I. Ng, S. Zhu, Z. Chen, and Z. Fang. A graph autoencoder approach to causal structure learning. *arXiv preprint arXiv:1911.07420*, 2019b.

J. Nocedal and S. J. Wright. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, 2nd edition, 2006.

R. Pamfil, N. Sriwattanaworachai, S. Desai, P. Pilgerstorfer, P. Beaumont, K. Georgatzis, and B. Aragam. DYNOTEARS: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, 2020.

J. Peters and P. Bühlmann. Identifiability of Gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2013a.

J. Peters and P. Bühlmann. Structural intervention distance (SID) for evaluating causal graphs. *Neural Computation*, 27, 2013b.

J. Peters, J. M. Mooij, D. Janzing, and B. Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15(1):2009–2053, 2014.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference - Foundations and Learning Algorithms*. MIT Press, 2017.

M. J. D. Powell. Nonlinear programming—sequential unconstrained minimization techniques. *The Computer Journal*, 12(3), 1969.

P. Pruthi, J. González, X. Lu, and M. Fiterau. Structure mapping for transferability of causal models. *arXiv preprint arXiv:2007.09445*, 2020.

N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999.

K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(Oct):2003–2030, 2006.

Y. Wang, V. Menkovski, H. Wang, X. Du, and M. Pechenizkiy. Causal discovery from incomplete data: A deep learning approach. *arXiv preprint arXiv:2001.05343*, 2020.

A. Wehenkel and G. Louppe. Graphical normalizing flows. *arXiv preprint arXiv:2006.02548*, 2020.

D. Wei, T. Gao, and Y. Yu. DAGs with no fears: A closer look at continuous optimization for learning Bayesian networks. In *Advances in Neural Information Processing Systems*, 2020.

M. Yang, F. Liu, Z. Chen, X. Shen, J. Hao, and J. Wang. CausalVAE: Disentangled representation learning via neural structural causal models. *arXiv preprint arXiv:2004.08697*, 2020.

Y. Yu, J. Chen, T. Gao, and M. Yu. DAG-GNN: DAG structure learning with graph neural networks. In *International Conference on Machine Learning*, 2019.

Y. Zeng, S. Shimizu, R. Cai, F. Xie, M. Yamamoto, and Z. Hao. Causal discovery with multi-domain LiNGAM for latent factors. *arXiv preprint arXiv:2009.09176*, 2020.

X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, 2018.

X. Zheng, C. Dan, B. Aragam, P. Ravikumar, and E. P. Xing. Learning sparse nonparametric DAGs. In *International Conference on Artificial Intelligence and Statistics*, 2020.

S. Zhu, I. Ng, and Z. Chen. Causal discovery with reinforcement learning. In *International Conference on Learning Representations*, 2020.

Appendices

A Optimality Conditions for Equality Constrained Problems

We review the optimality conditions for equality constrained problems, which are required for the study in Section 2.3 and 3.1.

Definition 2 (First-Order Necessary Conditions). *Suppose that θ^* is a local solution of (2), that the functions f and h in (2) are continuously differentiable, and that the LICQ holds at θ^*. Define the Lagrangian function of (2) as*

$$\mathcal{L}(\theta, \alpha) = f(\theta) + \alpha^\mathsf{T} h(\theta),$$

where $\alpha \in \mathbb{R}^p$. Then there is a Lagrange multiplier vector α^* such that the following conditions are satisfied at (θ^*, α^*):

$$\nabla_\theta \mathcal{L}(\theta^*, \alpha^*) = 0, \tag{10a}$$

$$h(\theta^*) = 0. \tag{10b}$$

Definition 3 (Second-Order Sufficient Conditions). *Let $\mathcal{L}(\theta, \alpha)$ be the Lagrangian function of (2) as in Definition 2. Suppose that for some feasible point θ^* there is a Lagrange multiplier vector α^* such that the conditions (10) are satisfied. Suppose also that*

$$y^\mathsf{T} \nabla^2_{\theta\theta} \mathcal{L}(\theta^*, \alpha^*) y > 0, \ \text{ for all } y \neq 0 \text{ with } \nabla_\theta h(\theta^*) y = 0.$$

Then θ^* is a strict local solution for (2).

B Proof of Proposition 1

Proof. We first consider the proof of function $h_{\text{exp}}(B)$. Its gradient is given by

$$\nabla_B h_{\text{exp}}(B) = (e^{B \circ B})^\mathsf{T} \circ 2B.$$

If part:
First note that $h_{\text{exp}}(B) = 0$ implies that B represents a DAG. Since there is no self-loop, the diagonal entries of B are zeros, so are the diagonal entries of $\nabla_B h_{\text{exp}}(B)$. Now consider the (j, i)-th entry of $\nabla_B h_{\text{exp}}(B)$ with $j \neq i$:

- If $B_{ji} = 0$, then it is clear that $(\nabla_B h_{\text{exp}}(B))_{ji} = 0$.
- If $B_{ji} \neq 0$, then there is an edge from node j to node i with weight B_{ji}. The other term $\left((e^{B \circ B})^\mathsf{T}\right)_{ji}$ indicates the total number of weighted walks from node i to node j. If both B_{ji} and $\left((e^{B \circ B})^\mathsf{T}\right)_{ji}$ are nonzeros, then there are weighted closed walks passing through node i and node j, contradicting the statement that B represents a DAG.

This indicates that at least one of B_{ji} or $\left((e^{B \circ B})^\mathsf{T}\right)_{ji}$ must be zero.

Only if part:
It suffices to consider the (j, i)-th entry of $\nabla_B h_{\text{exp}}(B)$ with $j \neq i$. $(\nabla_B h_{\text{exp}}(B))_{ji} = 0$ indicates that at least one of B_{ji} or $\left((e^{B \circ B})^\mathsf{T}\right)_{ji}$ is zero. Hence, the edge from node j to node i, if exists, must not belong to any cycle. This shows that all edges must not be part of a cycle and that B represents a DAG, i.e., $h_{\text{exp}}(B) = 0$.

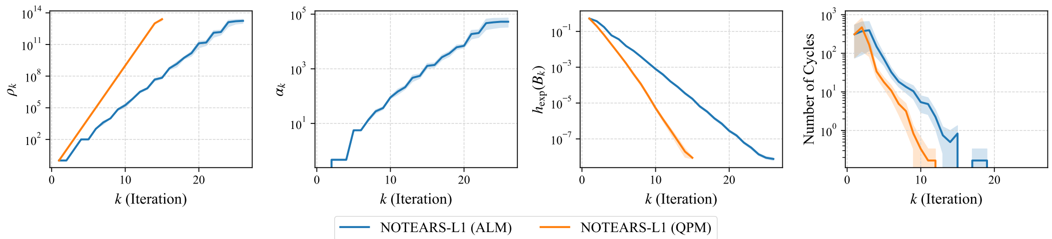
The function $h_{\text{bin}}(B)$ follows a similar proof and is omitted, by noting that

$$\nabla_B h_{\text{bin}}(B) = [(I + cB \circ B)^{d-1}]^\mathsf{T} \circ 2dcB.$$

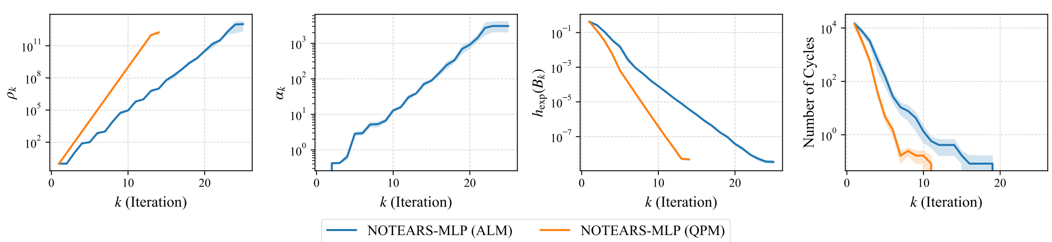
□

C Supplementary Experiment Results

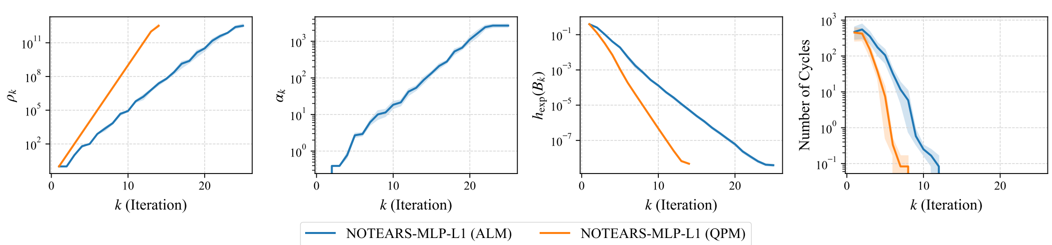
This section provides further results for Section 4.1 and 4.2, focusing on the variants of NOTEARS with nonlinear models and ℓ_1 penalty; see Figure 5 and 6.



(a) Optimization process of NOTEARS-L1 on the linear DAG model.

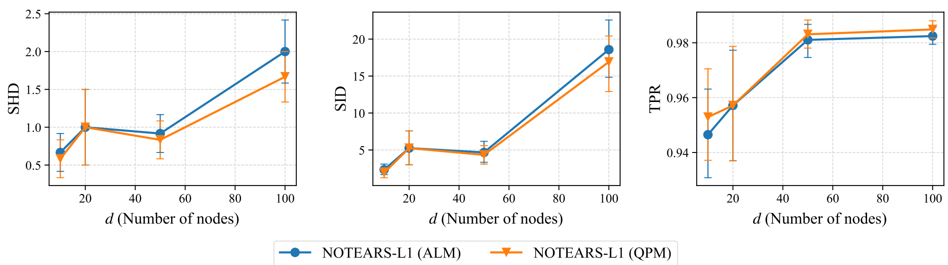


(b) Optimization process of NOTEARS-MLP on the nonlinear DAG model with GP.

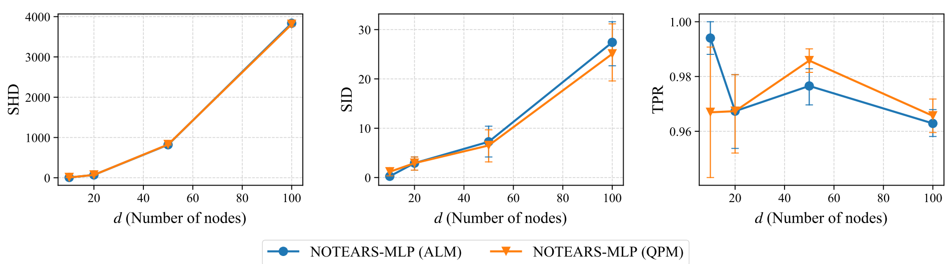


(c) Optimization process of NOTEARS-MLP-L1 on the nonlinear DAG model with GP.

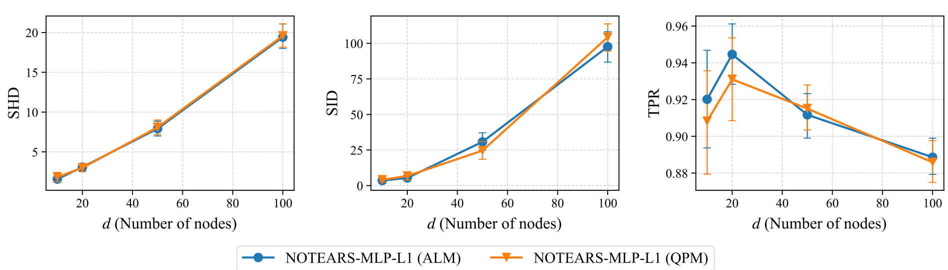
Figure 5: Optimization process of the continuous constrained formulation using ALM and QPM. Each data point corresponds to the k-th iteration. Shaded area denotes standard errors over 12 trials.



(a) Empirical results of NOTEARS-L1 on the linear DAG model.



(b) Empirical results of NOTEARS-MLP on the nonlinear DAG model with GP.



(c) Empirical results of NOTEARS-MLP-L1 on the nonlinear DAG model with GP.

Figure 6: Empirical results of the continuous constrained formulation using ALM and QPM. Lower is better, except for TPR. Error bars denote standard errors over 12 trials.