A Bregman Method for Structure Learning on Sparse Directed Acyclic Graphs

Manon Romain
CNRS, ENS Paris & Inria

Alexandre d'Aspremont
CNRS & ENS Paris

Abstract

We develop a Bregman proximal gradient method for structure learning on linear structural causal models. While the problem is non-convex, has high curvature and is in fact NP-hard, Bregman gradient methods allow us to neutralize at least part of the impact of curvature by measuring smoothness against a highly nonlinear kernel. This allows the method to make longer steps and significantly improves convergence. Each iteration requires solving a Bregman proximal step which is convex and efficiently solvable for our particular choice of kernel. We test our method on various synthetic and real data sets.

1 INTRODUCTION

Estimating directed acyclic graphs (DAGs) from observational data is a problem of rising importance in machine learning, with applications in biology (Sachs et al., 2005), genomics (Hu et al., 2018), economics (Imbens, 2019), time-series analysis (Malinsky and Spirtes, 2018) and causal inference (Pearl, 2009; Peters et al., 2017). More precisely, given n random variables $X_1 \ldots, X_n$, we seek to learn the structure of a Structural Causal Model (SCM), written

$$X_j = f_j(X_{\mathrm{PA}(j)}, \epsilon_j),$$

where ϵ_j is random noise and $\mathrm{PA}(j)$ are the parents of X_j, the subset of $\{X_1, \ldots, X_n\}$ on which X_j depends. This set of dependencies naturally forms a graph where $\{X_j\}_{j=1\ldots,n}$ are nodes, and edges are directed from nodes in $\mathrm{PA}(j)$ to X_j. In this context, we assume the graph G is acyclic, see e.g., (Peters et al., 2017, Def. 6.2).

Equivalently, we could describe our problem in the framework of Bayesian networks, where we search for a compact factorization of the joint distribution. Using the chain rule, we write,

$$P(X_1 \ldots, X_n) = \prod_{j=1}^{n} P(X_j \mid X_{\mathrm{PA}(j)}),$$

where $\mathrm{PA}(j)$ are the parents of X_j: a subset of $\{X_1, \ldots, X_{j-1}\}$ on which X_j depends. Unlike with SCMs, the underlying graph G is here naturally acyclic. Structure learning then consists in learning an adequate permutation of the order of variables and the sets $\{\mathrm{PA}(j)\}_{j=1\ldots,n}$. The combinatorial nature of this task makes it NP-hard (Chickering et al., 2004). In cases where the graph is identifiable, structure learning of the Bayesian network is equivalent to learning actual causal relationships from observational data, we refer the reader to (Pearl, 2009; Peters et al., 2017) for a more complete discussion.

Here, we focus on the well-studied (and simpler) case of *linear structural causal models* of the form,

$$X_j = \sum_{k=1}^{n} \beta_{jk} X_k + \epsilon_j, \quad \text{for } j = 1 \ldots, n, \quad (1)$$

where *parents* of X_j are $\mathrm{PA}(j) = \{X_k : \beta_{jk} \neq 0\}$, $\beta_{jj} = 0$ and $(\epsilon_j)_{j=1\ldots,n}$ are mutually independent, centered i.e. $\mathbb{E}(\epsilon_j) = 0$ and independent from variables in $\mathrm{PA}(j)$. We do not assume that the noise is Gaussian, and aim for a model that works across noise distributions. Structure learning here means searching for the weighted adjacency matrix $W^* = (\beta_{jk})$ of the directed acyclic graph G, hence constraining our optimization variable to be in the space of DAGs.

The structure learning problems we tackle here are non-convex and penalizing to impose acyclicity also gives them high curvature. While nonconvexity is somewhat unavoidable, we focus on taming curvature, which severely limits the performance of classical gradient methods by forcing them to take short steps. Bregman gradient algorithms along the lines of (Birnbaum et al., 2011; Bauschke et al., 2016; Lu et al., 2018) extend projected gradient methods using a Bregman proximal step instead of a classical projection

with respect to a norm, which pushes much of the curvature in the proximal step. Implicitly, using a highly nonlinear kernel allows the method to form a better local model of the function, so the method behaves very much like gradient descent in well conditioned settings, taking longer steps. In practice then, provided certain relative smoothness conditions (Bauschke et al., 2016; Bolte et al., 2018) are satisfied and the Bregman projection step can be solved efficiently (which is the critical part), Bregman gradient methods allow us to neutralize part of the curvature by measuring smoothness against a nonlinear kernel and improve convergence.

1.1 Related Work

Structure Learning methods historically divide into constraint-based methods that test for conditional independence relations and score-based methods that optimize a variety of heuristics.

Constraint-based   In this approach, we test multiple conditional dependencies. Finding two conditionally independent variables $X \perp\!\!\!\perp Y \mid Z$ means all paths from X to Y are of form $X \rightarrow \cdots \rightarrow Z \rightarrow \cdots \rightarrow Y$ or $X \leftarrow \cdots \leftarrow Z \rightarrow \cdots \rightarrow Y$ (Koller and Friedman, 2009). Under restrictive hypotheses (such as faithfulness), a graph can be constructed from such relationships. A popular example of this approach is the PC algorithm (Kalisch and Bühlmann, 2007).

Score-based   In a typical score-based method, a *discrete* scoring function is optimized over the space of DAGs. Scoring functions can be penalized likelihood such as BIC or coming from a Bayesian approach like BDeu (Heckerman et al., 1995). Greedy-hill climbing is popular to optimize such scores. GES (Chickering et al., 2004) reduces the search space to Markov equivalences classes. Van de Geer and Bühlmann (2013) use the similar ℓ_0 penalized maximum likelihood estimator proven to be consistent in high dimensions, under favorable assumptions. Those approaches usually require a form of faithfulness which is a restrictive assumption as shown by Uhler et al. (2013). Finally, some *hybrid* methods alternate between score optimization and constraint-based updates (Raskutti and Uhler, 2013). However the combinatorial nature of the DAG space makes structure learning computationally challenging. In this work, we will use a particular penalty to impose DAG structure to the graph.

NOTEARS   Zheng et al. (2018) proposed a novel smooth characterisation of the acyclicity of an adjacency matrix $W \in \mathbb{R}^{n \times n}$,

$$\mathbf{Tr} \exp(W \circ W) - n = 0,$$

where $\exp(\cdot)$ is the matrix exponential. This allows solving a continuous optimization problem over the whole space $\mathbb{R}^{n \times n}$ subject to the acyclicity constraint, instead of a discrete DAG problem, hence use off-the-shelf solvers for smooth non-convex optimization. Unfortunately, while the function $\mathbf{Tr} \exp$ is convex on the space of symmetric positive definite matrices $\mathbb{S}^n$, this property does not extend to $\mathbb{R}^{n \times n}$. New methods have used this constraint in association with several popular deep learning methods (Yu et al., 2019; Ng et al., 2019a,b; Lachapelle et al., 2020; Ng et al., 2020). With the exception of the original NOTEARS, all listed methods were designed for GPUs and require substantial computational power.

Contributions   Based on penalty terms derived in, e.g., Zheng et al. (2018), we use a non-convex Bregman composite optimization framework, to produce a more efficient algorithm for structure learning in linear SCMs. Our choice of Bregman kernel means that the method behaves as a better conditioned gradient method, and that each iteration of the Bregman proximal gradient method requires solving a convex optimization subproblem. We demonstrate the empirical effectiveness of a soft DAG constraint and competitive results even in high-dimensional settings ($m \ll n$) for a fraction of the time taken by NOTEARS.

The paper is organized as follows. We first introduce the problem of structure learning with DAG penalties and after explaining the general theory of the Bregman proximal gradient methods, we show how our task fits into this framework. We then demonstrate the effectiveness and numerical performance of our approach on several synthetic data sets and show intuitive results on real data sets.

Notations   We use $\|\cdot\|$ to denote the Frobenius norm on matrices and unusually $\|\cdot\|_1$ is the sum of absolute values of all the matrix coefficients i.e. $\|W\|_1 = \sum_{i,j} |W_{ij}|$ (this make notations more compact).

2   STRUCTURE LEARNING PROBLEM

The linear structure learning problem is formulated as follows. Let X be a $m \times n$ matrix of m i.i.d. observations from a linear structural causal model (1). We can write (1) in matrix form

$$X = XW^* + E, \tag{2}$$

where $W^* = (\beta_{jk}) \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix of the underlying directed acyclic graph G and $E = (\epsilon_j^{(i)}) \in \mathbb{R}^{m \times n}$ are $m \times n$ independent noise sam-

ples. We use the least-squares loss

$$\ell(W; X) = \frac{1}{m}\|X(\mathbf{I} - W)\|^2, \qquad (3)$$

however, everything that follows applies to any differentiable convex loss $\ell : \mathbb{R}^{n \times n} \to \mathbb{R}$. With finite samples and in high-dimensions ($m \ll n$), the regularized least-squares estimator provably recovers the correct support with high probability, both in the Gaussian case (Aragam et al., 2015) and in the non-Gaussian case (Loh and Wainwright, 2013). To enforce sparsity, we use the ℓ_1 penalty on the coefficients of W as a convex relaxation of ℓ_0, which keeps the Bregman proximal mapping convex.

It is known that $\Theta = (\mathbb{E}(X^T X))^{-1}$, the precision matrix (inverse covariance matrix), can be written,

$$\Theta = (I - W^*)\Omega^{-1}(I - W^*)^T,$$

where $\Omega \in \mathbb{R}^{n \times n}$ is the diagonal matrix with diagonal equal to the variances of $(\epsilon_j)_{j=1...,n}$. This fact has been used by (Loh and Bühlmann, 2014) to first estimate the moralized graph using the precision matrix and then use that information to restrict the search to DAGs matching the moralized graph uncovered.

Identifiability Shimizu et al. (2006) proved that if $(\epsilon_j)_{j=1...,n}$ are jointly independent and non-Gaussian distributed with strictly positive density, the graph is identifiable from the joint distribution. Peters et al. (2014) extended this result to Gaussian errors with equal variances. We refer the reader to, e.g., Peters et al. (2017) for a more complete discussion.

DAG Penalty We will first assume the (β_{jk}) coefficients are positive as our method is simpler in that case and generalize from there to negative edge weights. As introduced in NOTEARS (Zheng et al., 2018), we will use the smooth characterization of acyclicity recalled in the following proposition.

Proposition 2.1 *A positive weighted adjacency matrix W represents an acyclic graph if and only if $\mathbf{Tr}(I + \alpha W)^n = n$.*

Proof. Note that W has a cycle of length $k \geq 1$ starting at i if $[W^k]_{ii} > 0$. Since every cycle can be reduced to a cycle of length less or equal to n,

$$\begin{aligned}\mathbf{Tr}(I + \alpha W)^n &= \sum_i \sum_{k=0}^n \binom{n}{k} \alpha^k [W^k]_{ii} \\ &= n + \sum_i \sum_{k=1}^n \binom{n}{k} \alpha^k [W^k]_{ii} \\ &\geq n,\end{aligned}$$

with equality if and only if W is acyclic. $\qquad\square$

Note that this choice of constraint is arbitrary, we could have equivalently chosen the constraint

$\mathbf{Tr}\, P(W) = n$ for any polynomial P with strictly positive coefficients and $P(0) = 1$. As in Yu et al. (2019), we chose the form in Proposition 2.1, as its factored form allows for simpler expressions.

Instead of a hard penalty, as in NOTEARS, we use a regularization term here, as in (Ng et al., 2020). Overall, we seek to solve

$$\begin{aligned}\text{min.} \quad & \frac{1}{m}\|X(I - W)\|^2 + \lambda\|W\|_1 + \mu\,\mathbf{Tr}(I + \alpha W)^n \\ \text{s.t.} \quad & W \geq 0,\end{aligned}$$

$$(4)$$

in the variable $W \in \mathbb{R}^{n \times n}$, given samples $X \in \mathbb{R}^{m \times n}$, $\lambda \geq 0$ and $\mu \geq 0$ control sparsity and DAG regularization respectively. The DAG regularization term in this problem has a high curvature and slows down convergence of classical gradient methods. In the next section, we look at how to efficiently solve this problem (locally) by measuring smoothness against a well-chosen, highly nonlinear kernel.

Handling Negative Coefficients Proposition 2.1 holds only for positive edge weights. Zheng et al. (2018) alleviate this issue by squaring the adjacency matrix elementwise. However, squaring hurts regularity and we use a different approach here, writing W the adjacency matrix as the difference of its positive and negative parts i.e. $W = W^+ - W^-$ where $W^+ = \max(W, 0)$ and $W^- = \max(-W, 0)$.

Consider a new graph $\hat{G}$ obtained from G by replacing every edge weight β_{jk} by $|\beta_{jk}|$. Note that it's straightforward to see that $\hat{G}$ is acyclic if and only if G is acyclic. Moreover, the weighted adjacency matrix of $\hat{G}$ is $W^+ + W^-$.

The optimization problem on G with acyclicity of $\hat{G}$ is now very similar to the positive case,

$$\begin{aligned}\text{min.} \quad & \mu\,\mathbf{Tr}\left(I + \alpha(W^+ + W^-)\right)^n \\ & + \frac{1}{m}\|X(\mathbf{I} - W^+ + W^-)\|^2 \\ & + \lambda\|W^+\|_1 + \lambda\|W^-\|_1 \\ \text{s. t.} \quad & W^+, W^- \geq 0,\end{aligned}$$

$$(5)$$

where $W^+, W^- \in \mathbb{R}^{n \times n}$.

We prove that ambiguous edges i.e.where the weight is ill-defined cannot exist at critical points of the objective,

Lemma 2.2 *No pair $W^+, W^- \in \mathbb{R}_+^{n \times n}$ with indexes (j, k) such that both W_{jk}^+ and W_{jk}^- are non zeros are local minima of the objective of problem (5).*

Proof. Without loss of generality, assume $W_{jk}^+ \geq W_{jk}^- > 0$, let $\tilde{W}^+$ be the same matrix as W^+ except at (j, k) where $\tilde{W}_{jk}^+ = W_{jk}^+ - W_{jk}^-$ and similarly with $\tilde{W}^-$ where $\tilde{W}_{ij}^- = 0$, then $\tilde{W}^+, \tilde{W}^- \geq 0$,

$\tilde{W}^+ - \tilde{W}^- = W^+ - W^-$ and $\|\tilde{W}^\pm\|_1 < \|W^\pm\|_1$. Moreover, $\tilde{W}^\pm \leq W^\pm$ elementwise and the DAG regularization term is increasing in every entry of the matrix. So $(\tilde{W}^+, \tilde{W}^-)$ has a strictly better objective than (W^+, W^-). □

This means that the acyclicity constraints on $\hat{G}$ and G are completely equivalent.

Parameter estimation We mostly focus on structure learning here, i.e. estimation of the support of the adjacency matrix and disregard the problem of parameter estimation i.e. learning the actual values of $(\beta_{jk})_{j,k=1...,n}$. However, support estimation is the challenging task: once a correct graph G is known, parameter estimation in the regularized least square setting is a convex problem.

3 BREGMAN GRADIENT METHODS

Problems (4) and (5) are non-convex and the DAG penalty makes them highly nonlinear. Under more flexible relative smoothness assumptions introduced by Bolte et al. (2018), the performance and analysis of Bregman proximal gradient methods (Birnbaum et al., 2011; Bauschke et al., 2016; Lu et al., 2018) becomes much closer to that of classical gradient methods. In practice, Bregman gradient methods allow us to neutralize at least part of the effect of nonlinearities by measuring smoothness against a more nonlinear kernel than that produced by norms. Here, we use a variant of these methods introduced by Dragomir et al. (2019), that uses dynamical step size for faster convergence.

We first briefly recall the structure of Bregman gradient methods in the relative smoothness setting (aka the NoLips algorithm), as described in (Bolte et al., 2018). Let E be a Euclidean vector space endowed with an inner product $\langle \cdot, \cdot \rangle$. The method aims at solving non-convex composite minimization problems of the form

$$\min_{x \in \bar{C}} f(x) + g(x), \qquad (6)$$

where C is a nonempty convex open set in $\mathbb{R}^n$, f is a proper C^1 function with $\mathbf{dom} f \cap C \neq \emptyset$ and g is a proper and lower semicontinuous function.

3.1 Kernels, Bregman Divergences & Relative Smoothness

Let $h : E \to \mathbb{R}$ be a differentiable strictly convex function, which is called the *distance kernel*. It generates the *Bregman distance*

$$D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle. \qquad (7)$$

Note that D_h is generally asymmetric, therefore it is not a proper distance and is sometimes referred to as a *Bregman divergence*. However, we can deduce from the strict convexity of h that D_h enjoys a distance-like separation property: $D_h(x, x) = 0$ and $D_h(x, y) > 0$ for $x \neq y$.

We are now ready to define the notion of relative smoothness, also called L-smooth adaptability in Bolte et al. (2018).

Definition 3.1 (Relative smoothness) *We say that a differentiable function $f : E \to \mathbb{R}$ is L-smooth relatively to the distance kernel h if there exists $L > 0$ such that*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + L D_h(x, y), \\ \text{(RelSmooth)}$$

for all $x, y \in E$.

There are several convenient ways of checking the relative smoothness assumption. In the differentiable case, an equivalent definition reads as follows.

Definition 3.2 (Relative smoothness II) *We say that a C^∞ function $f : \mathbb{R}^{p \times p} \to \mathbb{R}$ is L-smooth relatively to the distance kernel h, if for all $W \in \mathbb{R}^{p \times p}$,*

$$\nabla^2 f(W) \preceq L \nabla^2 h(W). \qquad (8)$$

Note that if $h(x) = \frac{1}{2}\|x\|^2$ is the quadratic kernel, then $D_h(x, y) = \frac{1}{2}\|x-y\|^2$ and (RelSmooth) is equivalent to Lipschitz continuity of the gradient. By using different kernels, such as logarithms or power functions, it is possible to show that (RelSmooth) holds for functions that do not have a Lipschitz continuous gradient — see e.g., (Bauschke et al., 2016; Bolte et al., 2018).

Bregman Gradient Method Now that we are equipped with a non-Euclidean geometry generated by h, we can define the Bregman proximal gradient map with step size γ as follows

$$T_\gamma(x) \triangleq \qquad (9)$$
$$\underset{u \in C}{\operatorname{argmin}} \left\{ g(u) + \langle \nabla f(x), u - x \rangle + \frac{1}{\gamma} D_h(u, x) \right\}.$$

The Bregman gradient method then simply iterates this mapping as in Algorithm 1.

Algorithm 1 Bregman Gradient Method

Require: A function h such that (RelSmooth) holds with relative Lipschitz constant L, and step size $0 < \gamma \leq \frac{1}{L}$.
  Initialize $x_0 \in C$.
  for k = 1,2,... **do**
    $x^k \in T_\gamma(x^{k-1})$
  end for

If h is the squared Euclidean norm, we recover the projected gradient algorithm. In general, this of course assumes that the Bregman proximal map $T_\gamma(x)$ is simple to compute. We will see that in our case here, solving the iteration map is simply a convex problem.

It can be easily be proved that, under the relative smoothness condition and with $\gamma \in (0, \frac{1}{L})$, the sequence $\{f(x^k)\}_{k \geq 0}$ is nonincreasing. Convergence towards a critical point of the problem (6) is established in (Bolte et al., 2018) under additional assumptions (boundedness of the sequence and Kurdyka-Lojasiewicz property) which will hold in our case.

3.2 Bregman Method For Structure Learning

In the case where W^* is assumed positive, we can write problem (4) in the form (6) with

$$
\begin{aligned}
f(W) &= \mu \operatorname{\mathbf{Tr}}(\mathbf{I} + \alpha W)^n \\
g(W) &= \frac{1}{m}\|X(I - W)\|^2 + \lambda\|W\|_1,
\end{aligned} \tag{10}
$$

two C^∞ functions on $(0, +\infty)^{n \times n}$ that satisfy the Bregman Gradient methods' assumptions.

Relative Smoothness We choose a kernel which resembles our penalty function,

$$
h(W) = \mu(n-1)(1 + \alpha\|W\|)^n, \tag{11}
$$

a C^∞, convex function. We need to define the following convex subspace

$$
C_\alpha = \left\{ W \in \mathbb{R}_+^{n \times n} \text{ such that } \sum_{ij} W_{ij} \geq \frac{n}{(n-2)\alpha} \right\},
$$

and get the following result.

Theorem 3.3 *The DAG penalty f is 1-smooth relatively to h on the convex space C_α.*

The proof is left in the supplementary material. Note that we need to make the extra hypothesis that $W^* \in C_\alpha$, this assumption is not very restrictive as we can choose α, so this only impacts the regularity constants.

In the general case, where W^* takes on both positive and negative values, we change our DAG penalty function to:

$$
f(W^+, W^-) = \mu \operatorname{\mathbf{Tr}}(\mathbf{I} + \alpha(W^+ + W^-))^n, \tag{12}
$$

a function with similar properties than in the positive case. We then define

$$
h(W^+, W^-) = \mu(n-1)(1 + \alpha\|W^+ + W^-\|)^n, \tag{13}
$$

a C^∞, convex function as our kernel, and the convex space

$$
\begin{aligned}
C_\alpha^+ &= \Big\{ W^+, W^- \in \mathbb{R}_+^{n \times n} \text{ s.t.} \\
&\quad \sum_{ij}[W^+ + W^-]_{ij} \geq \frac{n}{(n-2)\alpha} \Big\}.
\end{aligned}
$$

Theorem 3.4 *$f : \left(\mathbb{R}_+^{n \times n}\right)^2 \to \mathbb{R}$ is 1-smooth relatively to h on C_α^+.*

The proof, a natural extension of the proof in the positive case, is available to the reader in the supplementary material. The Bregman proximal gradient map then writes very similarly to (14).

Bregman Prox Since g is a convex function, computing the solution $T_\gamma(W_k)$ of the Bregman proximal gradient map means solving a convex minimization problem, written

$$
\min_{W \in C_\alpha} \left\{ g(W) + \langle \nabla f(W_k), W - W_k \rangle + \frac{1}{\gamma} D_h(W, W_k) \right\},
$$

which is again

$$
\min_{W \in C_\alpha} \left\{ \frac{1}{m}\|X(I-W)\|^2 + \lambda\|W\|_1 + \langle \nabla_k, W \rangle + \frac{1}{\gamma} h(W) \right\}. \tag{14}
$$

where $\gamma > 0$ and $\nabla_k = \nabla f(W_k) - \frac{1}{\gamma}\nabla h(W_k)$. In our case, this means minimizing a sum of convex functions with linear constraints, and can therefore be solved efficiently using off-the-shelf convex optimization solvers, e.g., ECOS (Domahidi et al., 2013), MOSEK (ApS, 2019).

Having properly defined the functions and corresponding kernels as in (6), we can now apply algorithm 1. As a final step to our method, we threshold the output matrix W to get a binary adjacency matrix, to zero out negligible coefficients and because hard thresholding has been proven to reduce false discovery rate (Wang et al., 2016).

4 EXPERIMENTS

We compare our method against GES (Chickering, 2002), LiNGAM (Shimizu, 2014), CAM (Bühlmann et al., 2014), CCDr (Aragam and Zhou, 2015) and NOTEARS (Zheng et al., 2018). Notably, it seems that LiNGAM fails to run on ill-conditioned covariance matrices. With the exception of NOTEARS, we use the implementation available as part of the Causal Discovery Toolbox by Kalainathan and Goudet (2019). However, we only report results against the following methods which perform better on the given tasks:

- **NOTEARS**[1](Zheng et al., 2018): we use the most recent code version that was updated to use a DAG constraint in form $\operatorname{\mathbf{Tr}}(\mathbf{I}+\alpha W \circ W)^n - n = 0$ (Yu et al., 2019) instead of $\operatorname{\mathbf{Tr}} e^{W \circ W} - n = 0$ in the original paper.
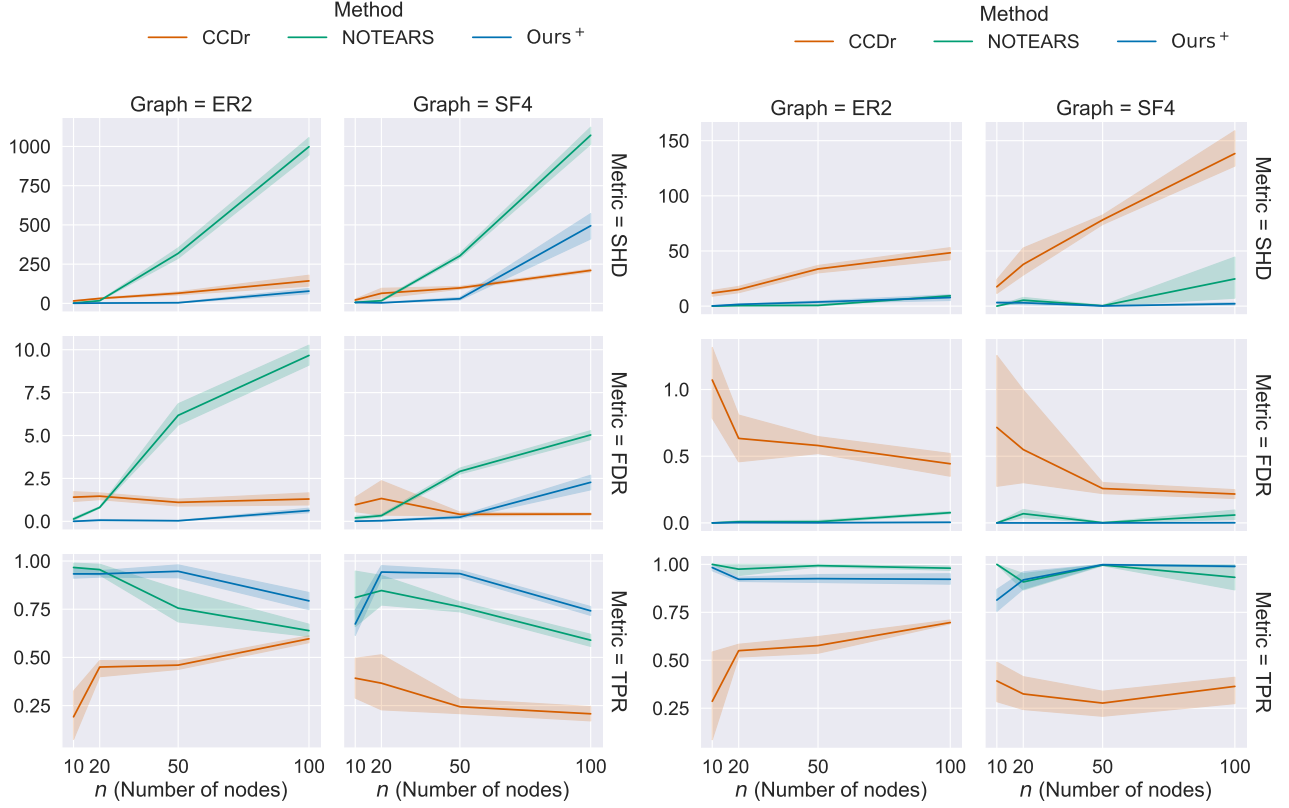
---

[1]https://github.com/xunzheng/notears

Figure 1: Performance for $m = 50$ (left) and $m = 200$ (right) lower is better except for TPR. The method proposed here performs consitently well compared to the two other.

- **CCDr** (Aragam et al., 2015): they use a concave penalty, interpolation of the ℓ_0 and ℓ_1 penalty and reparametrize the Gaussian likelihood estimator into a convex objective. Finally they use coordinate descent on the obtained objective.

We did not test methods such as Yu et al. (2019); Ng et al. (2019a,b); Lachapelle et al. (2020); Ng et al. (2020) that run on GPU and necessitate substantial computational power.

Metrics To compare the output of our model to the ground truth graph in synthetic examples, we let TP be the number of correctly detected edges and distinguish three error sources: M counts the missing edges compared to the skeleton, E counts the extra ones and R the reversed edges from the ground truth directed graph. The most standard metric in Structure Learning is Structural Hamming Distance (SHD), the number of additions, deletions, reversals to go from our output graph to the true graph, hence SHD $= M+E+R$. Other interesting metrics including False Discovery Rate (FDR) with FDR $= (E + R)/p$

where p is the true number of edges and True Positive Rate (TPR) where TPR $=$ TP$/p$.

Negative Weights In our experiments, we notice that, even though the ground truth adjacency matrix W^* contains both positive and negative coefficients (half in expectation), both NOTEARS and our algorithm, even when they perfectly recover the support, estimate all parameters to be positive. As an illustration, in our experiments, the proportion of correctly predicted edges by NOTEARS that were given a positive weight is $99.4(\pm1.5)\%$ — the minimum being 94.7%. Moreover, our algorithm restricted to positive coefficients i.e.solving problem (4) performs better than the general algorithm — even when the graph has negative weighted edges. Therefore we present results with this algorithm denoted Ours$^+$. We can't fully explain this phenomenon at this point.

4.1 Synthetic Datasets

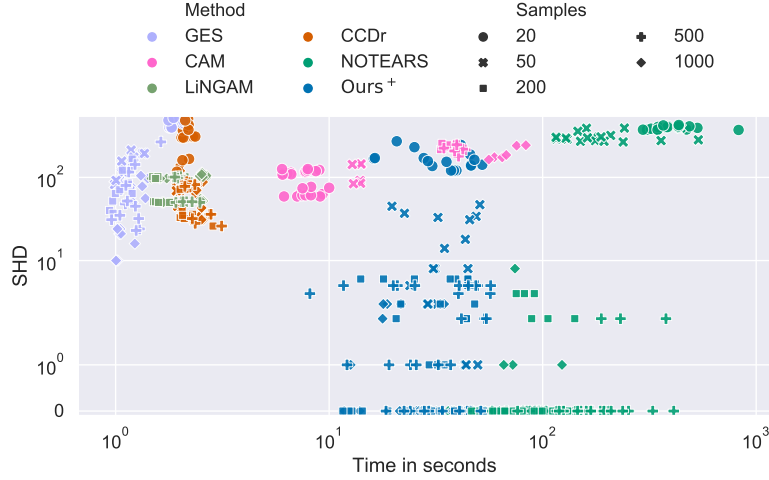We choose an experimental setup similar to the one of Zheng et al. (2018). Our datasets vary on 5 dif-

Manon Romain, Alexandre d'Aspremont

Figure 2: Performance vs. runtime for $n = 50$: for every dataset (with number of samples m, mean degree k, noise type and random seed), we plot Structural Hamming Distance (SHD) against run time of the corresponding algorithm.
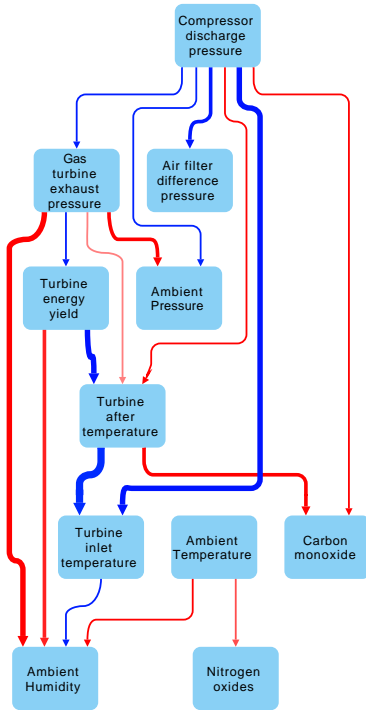

Figure 3: Graph learned on the Turbine dataset: color hue represents coefficient sign (blue is positive, red is negative); color intensity represents edge weight

ferent points: number of nodes, sparsity, graph type, noise type and number of samples. We generate G from one of two types of random graphs: Erdös-Rényi (ER) or scale-free (SF). We sample graphs with kn ($k = 2, 4$) edges on average and denote the corresponding graph ERk/SFk. Given G, we assign random uniform weights to the edges in a fixed range. Then we generate $m \in [20, 50, 100, 200, 500, 1000]$ i.i.d. samples from the distribution entailed by the graph with one of three noise distributions: Gaussian, Exponential or Gumbel noise.

The results presented on Figure 1 are aggregated over the three different noise types, several parameter choices (λ) and two random seeds. More detailed results (split by noise type and on ER4/SF2) can be found in Supplementary.

From $m = 200$ samples onward, both NOTEARS and our method learn the graph to quasi-perfection (SHD < 10) whereas CCDr stagnates to a sizable error level even with more data. On the other hand, when less data is available, CCDr becomes very competitive while NOTEARS has significant SHD but our method still behaves reasonably well, especially on sparser datasets such as ER2.

Interestingly, we can see on Figure 2 that the performance of our method increases steadily given more samples, unlike CCDr, while being approximately 10 times faster than NOTEARS, with better performance at a fixed number of samples. GES, LiNGAM and CAM are all much faster but with much worst SHD results, failing to get below 10 errors on datasets containing around 50 or 100 edges ($n = 50$, $k = 2, 4$).
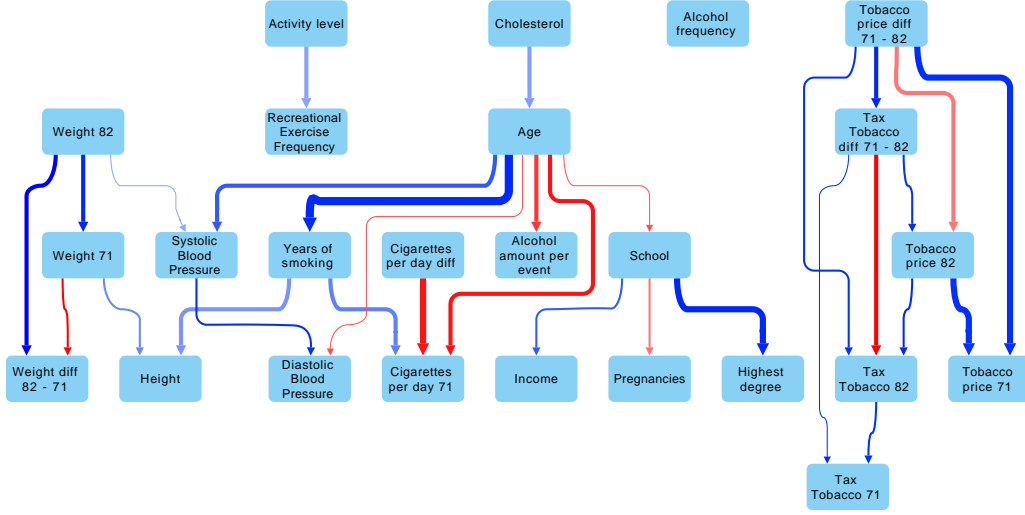
Figure 4: Graph learned on the NHEFS dataset: color hue represents coefficient sign (blue is positive, red is negative); width represents reliability from cross-validation; color intensity represents edge weight averaged over the 5-fold cross validation

4.2 Real Data

We run our algorithm on two real datasets.

Gas Turbine This dataset is composed of 36733 instances of 11 sensor measures from a Turkish gas turbine.[2] Results can be seen on Figure 3 and show key variables influencing emissions.

NHEFS The NHANES I Epidemiologic Follow-up Study (NHEFS) is a clinical study that followed a cohort from 1971-75 yo 1982 (Hernán and Robins, 2020). Data include the initial examination, the 1982 follow up and data about their environment at both times, e.g., tobacco prices.

The dataset included missing data, which we inferred using the `IterativeImputer` of `scikit-learn`, which regresses the missing values using all the remaining variables. We selected numerical and ordinal variables only, and normalized them. Our preprocessed dataset is now composed of $m = 1629$ patients with $n = 25$ measurements each. We ran 5−fold cross validation and report the graph where edges weights are averaged over splits (absolute weight shown as color intensity, sign as color) and frequency of the corresponding edge is shown as the edge width. The output of our algorithm is shown in Figure 4. We observe some intuitive connections such as a positive weight `school` $\rightarrow$ `highest degree`. The algorithm, which is unsupervised, also managed to group together all variables related to tobacco prices.

5 CONCLUSION AND FUTURE WORK

Using penalties to enforce acyclicity and thanks to an appropriate choice of kernel, we develop a Bregman proximal gradient method for structure learning on linear structural causal models with good regularity properties, for which each Bregman proximal step amounts to solving a convex quadratic program. This allows the method to make longer steps and significantly improves convergence. The method has relatively low complexity and is uniformly competitive with existing algorithms on various synthetic data sets. We test it on two real data sets where it produces intuitive DAG structures.

Acknowledgements

---

[2] Available here

References

ApS, M. (2019). *MOSEK Fusion API for Python 9.2.26.*

Aragam, B., Amini, A. A., and Zhou, Q. (2015). Learning directed acyclic graphs with penalized neighbourhood regression.

Aragam, B. and Zhou, Q. (2015). Concave penalized estimation of sparse gaussian bayesian networks. *Journal of Machine Learning Research*, 16(1):2273–2328.

Bauschke, H. H., Bolte, J., and Teboulle, M. (2016). A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348.

Birnbaum, B., Devanur, N. R., and Xiao, L. (2011). Distributed algorithms via gradient descent for fisher markets. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 127–136.

Bolte, J., Sabach, S., Teboulle, M., and Vaisbourd, Y. (2018). First order methods beyond convexity and lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM Journal on Optimization*, 28(3):2131–2151.

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

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

Chickering, D. M., Heckerman, D., and Meek, C. (2004). Large-sample learning of bayesian networks is np-hard. In *J. Mach. Learn. Res.*

Domahidi, A., Chu, E., and Boyd, S. (2013). Ecos: An socp solver for embedded systems. In *2013 European Control Conference (ECC)*, pages 3071–3076. IEEE.

Dragomir, R.-A., d'Aspremont, A., and Bolte, J. (2019). Quartic first-order methods for low rank minimization. *arXiv preprint arXiv:1901.10791.*

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243.

Hernán, M. A. and Robins, J. M. (2020). *Causal Inference: What If.* Boca Raton: Chapman & Hall, CRC.

Hu, P., Jiao, R., Jin, L., and Xiong, M. (2018). Application of causal inference to genomic analysis: Advances in methodology. *Frontiers in Genetics*, 9:238.

Imbens, G. (2019). Potential Outcome and Directed Acyclic Graph Approaches to Causality: Relevance for Empirical Practice in Economics. NBER Working Papers 26104, National Bureau of Economic Research, Inc.

Kalainathan, D. and Goudet, O. (2019). Causal discovery toolbox: Uncover causal relationships in python.

Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(Mar):613–636.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques.* MIT press.

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

Loh, P.-L. and Bühlmann, P. (2014). High-dimensional learning of linear causal networks via inverse covariance estimation. *Journal of Machine Learning Research*, 15.

Loh, P.-L. and Wainwright, M. J. (2013). Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*, pages 476–484.

Lu, Y., Fan, Y., Lv, J., and Noble, W. S. (2018). Deeppink: reproducible feature selection in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 8676–8686.

Malinsky, D. and Spirtes, P. (2018). Causal structure learning from multivariate time series in settings with unmeasured confounding. In *Proceedings of 2018 ACM SIGKDD Workshop on Causal Discovery*, pages 23–47.

Ng, I., Fang, Z., Zhu, S., Chen, Z., and Wang, J. (2019a). Masked gradient-based causal structure learning.

Ng, I., Ghassami, A., and Zhang, K. (2020). On the role of sparsity and dag constraints for learning linear dags. *ArXiv*, abs/2006.10201.

Ng, I., yu Zhu, S., Chen, Z., and Fang, Z. (2019b). A graph autoencoder approach to causal structure learning. *ArXiv*, abs/1911.07420.

Pearl, J. (2009). *Causality: Models, Reasoning and Inference.* Cambridge University Press, USA, 2nd edition.

Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of Causal Inference: Foundations and Learning Algorithms.* MIT Press, Cambridge, MA, USA.

Peters, J., Mooij, J. M., Janzing, D., and Schölkopf, B. (2014). Causal discovery with continuous additive noise models. *J. Mach. Learn. Res.*, 15:2009–2053.

Raskutti, G. and Uhler, C. (2013). Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1):e183.

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

Shimizu, S. (2014). Lingam: Non-gaussian methods for estimating causal structures. *Behaviormetrika*, 41(1):65–98.

Shimizu, S., Hoyer, P. O., Hyvärinen, A., and Kerminen, A. (2006). A linear non-gaussian acyclic model for causal discovery. *J. Mach. Learn. Res.*, 7:2003–2030.

Uhler, C., Raskutti, G., Bühlmann, P., and Yu, B. (2013). Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, 41(2):436–463.

Van de Geer, S. and Bühlmann, P. (2013). ℓ_0 penalized maximum likelihood for sparse directed acyclic graphs. *The Annals of Statistics*, 41(2):536–567.

Wang, X., Dunson, D., and Leng, C. (2016). No penalty no tears: Least squares in high-dimensional linear models. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of Machine Learning Research*, volume 48, pages 1814–1822, New York, New York, USA. PMLR.

Yu, Y., Chen, J., Gao, T., and Yu, M. (2019). Daggnn: Dag structure learning with graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*.

Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. (2018). Dags with no tears: Continuous optimization for structure learning. In *NeurIPS*.

A Bregman Method for Structure Learning on Sparse Directed Acyclic Graphs

1 Dynamic NoLips

In this work, we use the dynamic variant of NoLips of **?** that uses more aggressive step size for faster convergence,

---
Algorithm 1 Dynamic NoLips

Require: A function h such that f is smooth relatively to h, initial step size $\gamma_0 > 0$ and maximal step size $\gamma_{\max}$.

    Initialize $W \in C$ such that $\Psi(W) < \infty$.
    $\gamma \leftarrow \gamma_0$
    repeat
      repeat
        $W^+ \leftarrow T_\gamma(W)$
        if the decrease condition (1) is not verified **then**
          $\gamma \leftarrow \gamma/2$
        end if
      until decrease condition (1) is verified
      $W \leftarrow T_\gamma(W)$
      $\gamma \leftarrow \min(2\gamma, \gamma_{\max})$
    until convergence criterion

The sufficient decrease condition is, for $W^+ = T_\gamma(W)$,

$$f(W^+) \leq f(W) + \langle \nabla f(W), W^+ - W \rangle + \frac{1}{\gamma} D_h(W^+, W). \tag{1}$$

? proves that this modified version of NoLips allows for larger step sizes while maintaining convergence guaranties. We refer the reader to their work for further information and proofs.

As a convergence criterion, we use convergence of the L_2 error, i.e.for $\ell_k = \frac{1}{m}\|X(\mathbf{I} - W_k)\|^2$ being the L_2 error at step k, we stop when $\left|\frac{\ell_k - \ell_{k-1}}{\ell_{k-1}}\right| \leq \tau$, with an user-defined τ.

2 Proofs of relative smoothness

This section provides proofs for theorems 3.3 and 3.4 that were left out in the main paper.

2.1 Positive case - Proof of Theorem 3.3

Recall that in this case, for $W \in \mathbb{R}_+^{n \times n}$,

$$\begin{aligned} f(W) &= \mu \, \mathbf{Tr}(\mathbf{I} + \alpha W)^n \\ h(W) &= \mu(n-1)(1 + \alpha\|W\|)^n \end{aligned} \tag{2}$$

We will prove,

Theorem 2.1 (3.3 in main paper) *In the set* $\mathcal{M}_\alpha = \left\{ W \in \mathbb{R}_+^{n \times n} \text{ such that } \|W\| \geq \frac{1}{(n-2)\alpha} \right\}$, f *is 1-smooth relatively to* h *i.e.,*

$$\left| \nabla^2 f(W)[H, H] \right| \leq \nabla^2 h(W)[H, H].$$

Proof. Note that both f and h are C^∞ on $\mathcal{M}_\alpha$. Note that $f(W) = \mu \sum_{k=0}^n \binom{n}{k} \alpha^k \, \mathbf{Tr}(W^k)$, using this form and Taylor series, it is easier to get to:

$$\tfrac{1}{2} \nabla^2 f(W)[H, H] = \mu \sum_{k=2}^n \binom{n}{k} \alpha^k \, \mathbf{Tr}(X_k H), \tag{3}$$

where $X_k = \sum_{j=0}^{k-2} (j+1) W^j H W^{k-2-j}$.

So:

$$|\nabla^2 f(W)[H, H]| \le \mu n(n-1)\alpha^2 (1 + \alpha \|W\|)^{n-2} \|H\|^2 \tag{4}$$

Moreover for $W \in \mathcal{M}_\alpha$, and $\tilde{W} = \frac{W}{\|W\|}$, we have:

$$
\begin{aligned}
\nabla^2 h(W)[H, H] \quad &= \mu n(n-1)(1 + \alpha \|W\|)^{n-2} \left[(n-1)\alpha^2 (\mathbf{Tr}\, \tilde{W}^T H)^2 + \alpha \frac{1 + \alpha \|W\|}{\|W\|} \left(\|H\|^2 - (\mathbf{Tr}\, \tilde{W}^T H)^2 \right) \right] \\
&= \mu n(n-1)(1 + \alpha \|W\|)^{n-2} \Big[\underbrace{\left((n-1)\alpha^2 - \alpha \frac{1 + \alpha \|W\|}{\|W\|} \right)}_{\ge 0} (\mathbf{Tr}\, \tilde{W}^T H)^2 + \underbrace{\alpha \frac{1 + \alpha \|W\|}{\|W\|}}_{\ge \alpha} \|H\|^2 \Big] \\
&\ge \mu n(n-1)\alpha^2 (1 + \alpha \|W\|)^{n-2} \|H\|^2
\end{aligned}
\tag{5}
$$

So we proved that $|\nabla^2 f(W)[H, H]| \le \nabla^2 h(W)[H, H]$. $\qquad \square$

Now, note that using the inequality $\|x\|_1 \le \sqrt{n} \|x\|_2$ for $x \in \mathbb{R}^n$, the set

$$C_\alpha = \left\{ W \in \mathbb{R}_+^{n \times n} \text{ such that } \|W\|_1 \ge \tfrac{n}{(n-2)\alpha} \right\},$$

is a convex subset of $\mathcal{M}_\alpha$.

2.2   General case - Proof of Theorem 3.4

Let $f : \mathbb{R}^{n \times n} \to \mathbb{R}$ and $h : \mathbb{R}^{n \times n} \to \mathbb{R}$ be the functions of the positive case (see Eq. (2)) and the function sum $s : A, B \to A + B$.

In this case:

$$
\begin{aligned}
\hat{f}(W^+, W^-) \quad &= \mu \, \mathbf{Tr}\, (\mathbf{I} + \alpha(W^+ + W^-))^n = f \circ s(W^+, W^-) \\
\hat{h}(W^+, W^-) \quad &= \mu (n-1)(1 + \alpha \|W^+ + W^-\|)^n = h \circ s(W^+, W^-)
\end{aligned}
\tag{6}
$$

We will prove the following lemma that allows to adapt the proof of the positive case to the general case.

Lemma 2.2 *For all C^∞ functions $\hat{f} : \left(\mathbb{R}_+^{n \times n} \right)^2 \to \mathbb{R}$ and $f : \mathbb{R}_+^{n \times n} \to \mathbb{R}$ such that $\hat{f} = f \circ s$, for all $W \in \left(\mathbb{R}_+^{n \times n} \right)^2$ and $H \in (\mathbb{R}^{n \times n})^2$, we have:*

$$
\begin{aligned}
\langle \nabla \hat{f}(W), H \rangle \quad &= \langle \nabla f(s(W)), s(H) \rangle \\
\nabla^2 \hat{f}(W)[H, H] \quad &= \nabla^2 f(s(W))[s(H), s(H)]
\end{aligned}
\tag{7}
$$

Proof. Composition with a linear operator. $\qquad \square$

So in our case, for all $W \in \left(\mathbb{R}_+^{n \times n} \right)^2$ such that $\|s(W)\| \ge \frac{1}{(n-2)\alpha}$, and all $H \in (\mathbb{R}^{n \times n})^2$, we have:

$$|\nabla^2 \hat{f}(W)[H, H]| = |\nabla^2 f(s(W))[s(H), s(H)]| \le \nabla^2 h(s(W))[s(H), s(H)] = \nabla^2 \hat{h}(W)[H, H] \tag{8}$$

where $\hat{h} = h \circ s$. So $\hat{f}$ is 1-smooth relatively to $\hat{h}$ on $\mathcal{M}_\alpha^+ = \left\{ W \in \left(\mathbb{R}_+^{n \times n} \right)^2 \mid s(W) \in \mathcal{M}_\alpha \right\}$, so on the convex subset $C_\alpha^+ = \left\{ W \in \left(\mathbb{R}_+^{n \times n} \right)^2 \mid s(W) \in C_\alpha \right\}$.

3 Experimental details & further results

3.1 Synthetic datasets

We found $\alpha = 0.1/n$ and $\mu = 100$ to be working well and used it in all our synthetic data experiments. We used $\lambda \in [0, 10^{-6}, 10^{-4}]$ and averaged results for both NOTEARS and our algorithm, however the results are not very sensitive to the choice of ℓ_1 penalty. We used $\tau = 10^{-7}$ across experiments.

Figures 1 and 2 show that both NOTEARS and our algorithm seem to perform similarly with varying noise type and graph architecture, however, CCDr seems to perform significantly better when graphs are sparser ($k = 2$).

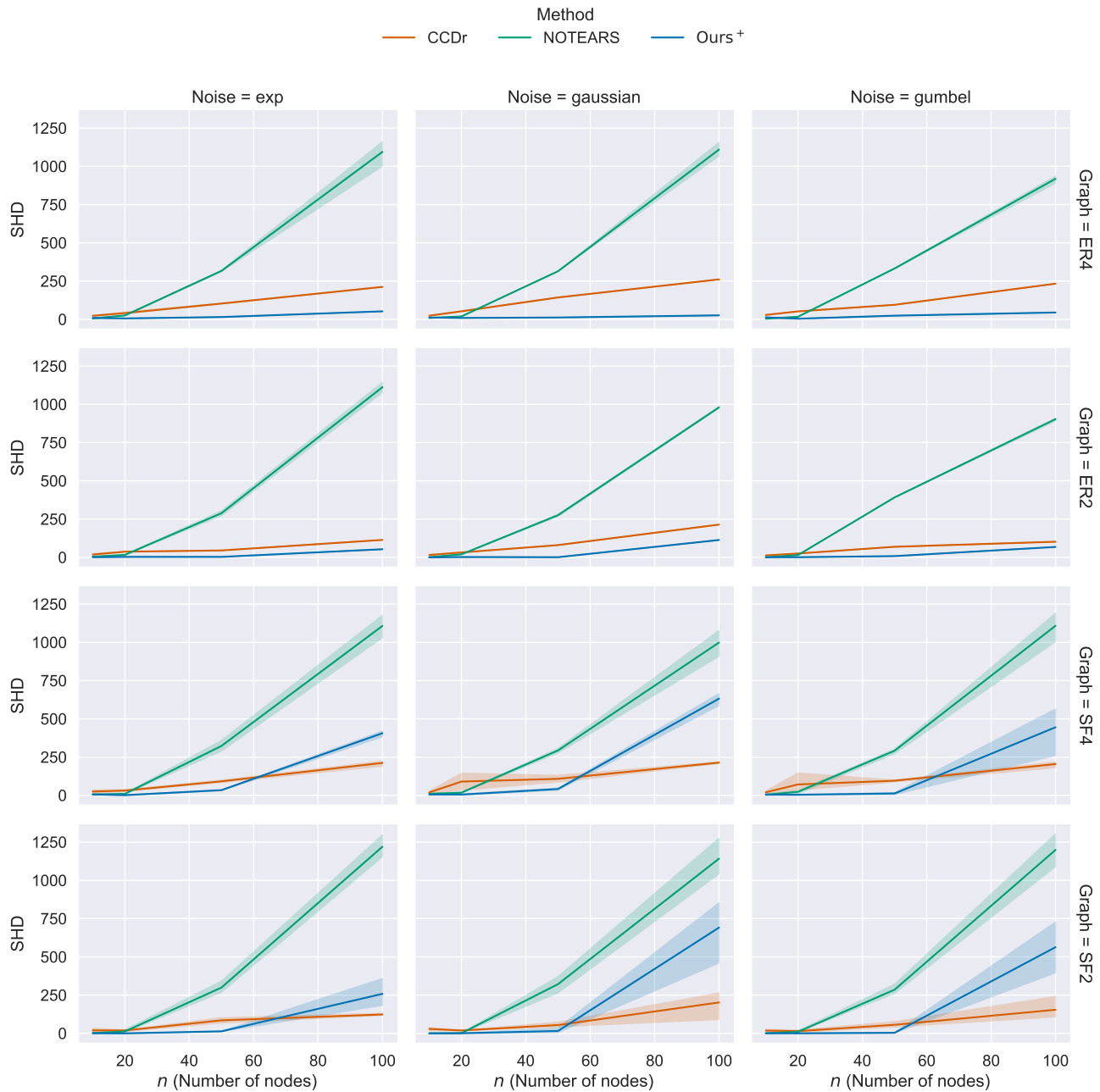A short Python demonstration is available as part of the supplementary material.

Figure 1: Structural Hamming Distance (SHD) on 12 different types of datasets (graph type, mean degree and noise type differ) for $m = 50$
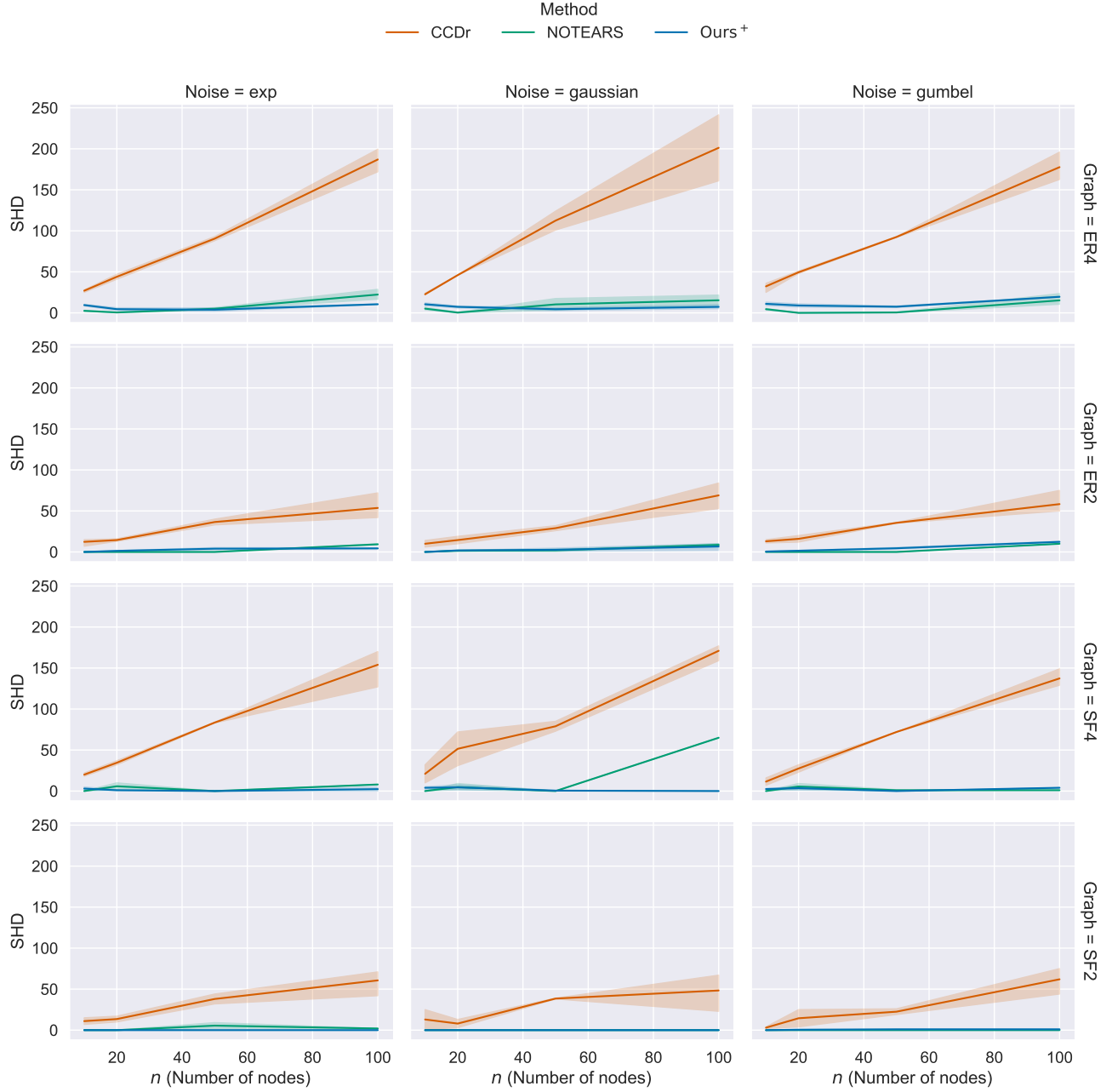
Figure 2: Structural Hamming Distance (SHD) on 12 different types of datasets (graph type, mean degree and noise type differ) for $m = 200$