Structure learning in polynomial time: Greedy algorithms, Bregman information, and exponential families

Anonymous Author(s)
Affiliation
Address
email

Abstract

Greedy algorithms have long been a workhorse for learning graphical models, and more broadly for learning statistical models with sparse structure. In the context of learning directed acyclic graphs, greedy algorithms are popular despite their worst-case exponential runtime. In practice, however, they are very efficient. We provide new insight into this phenomenon by studying a general greedy score-based algorithm for learning DAGs. Unlike edge-greedy algorithms such as the popular GES and hill-climbing algorithms, our approach is vertex-greedy and requires at most a polynomial number of score evaluations. We then show how recent polynomial-time algorithms for learning DAG models are a special case of this algorithm, thereby illustrating how these order-based algorithms can be rigourously interpreted as score-based algorithms. This observation suggests new score functions and optimality conditions based on the duality between Bregman divergences and exponential families, which we explore in detail. Explicit sample and computational complexity bounds are derived. Finally, we provide extensive experiments suggesting that this algorithm indeed optimizes the score in a variety of settings.

1 Introduction

Learning the structure of a graphical model from data is a notoriously difficult combinatorial problem with numerous applications in machine learning, artificial intelligence, and causal inference as well as scientific disciplines such as genetics, medicine, and physics. Owing to its combinatorial structure, greedy algorithms have proved popular and efficient in practice. For undirected graphical models (e.g. Ising, Gaussian) in particular, strong statistical and computational guarantees exist for a variety of greedy algorithms [20, 21]. These algorithms are based on the now well-known forward-backward greedy algorithm [22, 47], which has been applied to a range of problems beyond graphical models including regression [47], multi-task learning [43], and atomic norm regularization [36].

Historically, the use of the basic forward-backward greedy scheme for learning directed acyclic graphical (DAG) models predates some of this work, dating back to the classical greedy equivalence search [GES, 8] algorithm. Since its introduction, GES has become a gold-standard for learning DAGs, and is known to be asymptotically consistent under certain assumptions such as faithfulness and score consistency [8, 27]. Both of these assumptions are known to hold for certain parametric families [16], however, extending GES to distribution-free settings has proven difficult. Furthermore, although GES is in practice extremely efficient and has been scaled up to large problem sizes [35], it lacks polynomial-time guarantees. An important problem in this direction is the development of provably polynomial-time, consistent algorithms for DAG learning in general settings.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

In this paper, we revisit greedy algorithms for learning DAGs with an eye towards these issues. We propose a greedy algorithm for this problem—distinct from GES—and study its computational and statistical properties. In particular, it requires at most a polynomial number of score evaluations and provably recovers the correct DAG for properly chosen score functions. Furthermore, we illustrate its intimate relationship with existing order-based algorithms, providing a link between these existing approaches and classical score-based approaches. Along the way, we will see how the analysis itself suggests a family of score functions based on the Bregman information [2], which are well-defined without specific distributional assumptions.

Contributions At a high-level, our goal is to understand what kind of finite-sample and complexity guarantees can be provided for greedy score-based algorithms in general settings. In doing so, we aim to provide deeper insight into the relationships between existing algorithms. Our main contributions can thus be outlined as follows:

- A generic greedy forward-backward scheme for optimizing score functions defined over DAGs. Unlike existing *edge*-greedy algorithms that greedily add or remove edges, our algorithm is *vertex*-greedy, i.e. it greedily adds vertices in a topological sort.
- We show how several existing order-based algorithms from the literature are special cases of this algorithm, for properly defined score functions. Thus, we bring these approaches back under the umbrella of score-based algorithms.
- We introduce a new family of score functions derived from the Bregman information, and analyze the sample and computational complexity of our greedy algorithm for this family of scores.
- We explore the optimization landscape of the resulting score functions, and provide evidence that not only does our algorithm provably recover the true DAG, it does so by globally optimizing a score function.

The last claim is intriguing: It suggests that it is possible to globally optimize certain Bayesian network scores in polynomial-time. In other words, despite the well-known fact that global optimization of Bayesian networks scoring functions is NP-hard [7, 9], there may be natural assumptions under which these hardness results can be circumvented. This is precisely the case, for example, for undirected graphs: In general, learning Markov random fields is NP-hard [41], even for trees [13], but special cases such as Gaussian graphical models [3, 26] and Ising models [4, 24, 45] can be learned efficiently. Nonetheless, we emphasize that these results are merely empirical, and a proof of this fact beyond the linear case remains out of reach.

Previous work The literature on BNSL is vast, so we focus this review on related work involving score-based and greedy algorithms. For a broad overview of BNSL algorithms, see the recent survey [19] or the textbooks [34, 40]. The current work is closely related to and inspired by generic greedy algorithms such as [20–22, 36, 43, 47]. Existing greedy algorithms for score-based learning include GES [8], hill climbing [6, 42], and A* search [46]. In contrast to these greedy algorithms are global algorithms that are guaranteed to find a global optimum such as integer programming [11, 12] and dynamic programming [29, 37, 38]. More recently, a series of algorithms based around the idea of *order search*—i.e. first searching for a topological sort—have led to significant breakthroughs, most notable of which are finite-sample and strong polynomial-time guarantees [5, 15, 17, 18, 30]. It will turn out that many of these algorithms are special cases of the greedy algorithm we propose; we revisit this interesting topic in Section 3.1.

2 Background

Let $X = (X_1, \ldots, X_d)$ be a random vector with distribution $\mathcal{D}$. The goal of structure learning is to find a DAG $W = (V, E)$, also called a Bayesian network (BN), for the joint distribution $\mathcal{D}$. Traditionally, there have been two dominant approaches to Bayesian network structure learning (BNSL): Constraint-based and score-based. In constraint-based algorithms such as the PC [39] and MMPC [44] algorithms, tests of conditional independence are used to identify the structure of a DAG via exploitation of d-separation in DAG models. Score-based algorithms such as GES [8] define an objective function over DAGs such as the likelihood or a Bayesian posterior, and seek to optimize this score.

To formalize this, denote the space of DAGs on d nodes by DAG and let $S : \mathsf{DAG} \to \mathbb{R}$ be a score function. Intuitively, S assigns to each DAG W a "score" $S(W)$ that evaluates the fit between W and $\mathcal{D}$. In the sequel, we assume without loss of generality that the goal is to minimize the score:

$$\min_{W \in \mathsf{DAG}} S(W). \tag{1}$$

Although this is an NP-hard combinatorial optimization problem, we can ask whether or not it is possible to design score functions S which can be optimized efficiently, and whose minimizers are close to W. In order for this problem to be well-posed, there must be a *unique* W that we seek; namely, W must be identifiable from $\mathcal{D}$. The problem of identifiability will be taken up further in Section 4, where it will be connected to the choice of score function. For now, our primary goal is to solve the problem (1).

Exact algorithms Some of the earliest exact methods for score-based learning ("exact" meaning a genuine *global* minimizer of (1) is returned) relied on the following basic idea [28, 31, 37, 38]: Use dynamic programming to search for optimal sinks in W, remove these sinks, and recursively find optimal sinks in the resulting subgraph. In doing so, a topological sort of W can be learned, and from this sort, the optimal DAG can be easily learned. In other words, once the topological sort is known, finding the corresponding DAG is relatively easy. In the sequel, we refer to the problem of finding the topological sort of W as *order search*. Unfortunately, searching for optimal sinks involves computing $d2^{d-1}$ local scores, which is both time and memory intensive.

Poly-time algorithms Recently, a third family of algorithms based on applying the idea of order search has led to significant breakthroughs in our understanding of this problem [5, 15, 17, 18, 30]. Most notably, unlike the exact algorithms described above, these algorithms run in polynomial-time. The key distinction between these algorithms and exact algorithms is the clever exploitation of specific distributional (e.g. moments) or structural properties (e.g. linearity) of $\mathcal{D}$, and as a result do not optimize a specific score function. In contrast, exact algorithms apply to *any* score S, and do not require any distributional assumptions.

Motivation It is tempting to want to draw connections between exact algorithms and poly-time algorithms: After all, they both rely on the same fundamental principle of order search. In this paper, we explore this connection from the perspective of *greedy optimization*. In particular, we will show how existing polynomial-time algorithms are special cases of a generic greedy forward-backward search algorithm for solving (1) under specific choices of S, and show how this leads to new insights for this problem. We *do not* prove that this algorithm exactly solves (1) (save for the exceptional case of linear models), however, we provide empirical evidence to support this idea in principle. Since the score-based learning problem is NP-hard, this is of course not possible without additional assumptions.

Notation Let n be the number of samples we observe. Each sample is a vector of the form $X = (X_1, \ldots, X_d)$ on d variables. In this paper, W is used for DAGs and the vertex set is $[d] = \{1, 2, \ldots, d\}$. Naturally, we match vertex i to the variable X_i. We denote the set of parents of a vertex i with $\mathrm{pa}_W(i)$, dropping the subscript when it's clear from context. We will also abuse notation and use W for the adjacency matrix of the graph W as well. Let $W^{(i)}$ denote the ith column of W, whose nonzero entries are precisely at the set of parents of vertex i. Let $W[j,k]$ denote the (j,k)th entry of the matrix.

3 The GFBS algorithm

In this section, we will describe the greedy algorithm in a general framework assuming the score S is arbitrary. In subsequent sections, we will specialize to particular models or scores, as necessary.

For an edge e of W, denote by W^{-e} the matrix W with the entry corresponding to e zeroed out. For any set of vertices J and vertex $i \notin J$, denote by $W[J \to i]$ the matrix W where the ith column $W^{(i)}$ is replaced by the indicator vector of J. That is,

$$W^{-e}[j,k] = \begin{cases} W[j,k] & \text{if } (j,k) \neq e, \\ 0 & \text{otherwise,} \end{cases} \qquad W[J \to i][j,k] = \begin{cases} W[j,k] & \text{if } k \neq i, \\ 1 & \text{if } k = i \text{ and } j \in J \\ 0 & \text{if } k = i \text{ and } j \notin J \end{cases}$$

Algorithm 1: Greedy Forward-Backward Search

Input: Dataset X, tolerance parameter $\gamma \geq 0$
Output: $\arg\min_{\text{DAG } W} S(W)$

1 $W = \emptyset$ `// n-vertex graph with no edges`
2 $T = []$ `// The ordering`
`// Forward phase`
3 **for** $iter = 1$ *to* d **do**
4 $\quad i = \arg\min_{i \notin T} S(W[T \to i])$ `// Minimize jump in score`
5 $\quad T.append(i)$
`// Backward phase`
6 **for** *edge* e *in* W **do**
7 $\quad$ **if** $S(W^{-e}) - S(W) \leq \gamma$ **then**
8 $\quad\quad W = W^{-e}$ `// Delete the edge` e
9 **return** W `// Guaranteed to be a DAG`

In Algorithm 1, we outline a general framework based on greedy forward-backward search to recover W. We denote this by GFBS for short. Crucially, in contrast to traditional greedy algorithms for structure learning, GFBS is *vertex-greedy*: Instead of greedily adding edges to W, GFBS greedily adds *vertices* to first build up a topological sort T of W. Specifically, Line 4 in the algorithm greedily finds the next vertex i to add to the ordering, by comparing the score changes if we set the parents of i to be the vertices already in the ordering. Conceptually, this step is one of the most important differences from GES which adds edges one at a time. We make this distinction clear in Appendix A.

It is worth emphasizing that the output of GFBS is guaranteed to be a DAG. Furhtermore, in practice, Algorithm 1 can be adapted to different settings as follows:

1. Suppose the score is decomposable; i.e. $S(W) = \sum_{i \leq d} S_i(W^{(i)})$ for functions S_i, an example of which would be the least-squares loss. Let v_T denote the indicator vector of T. Then, in each iteration of the forward phase, in line 4, we have

$$\arg\min_{i \notin T} S(W[T \to i]) = \arg\min_{i \notin T} S_i(v_T)$$

   Therefore, this step could be made efficient by only computing the local scores S_i as opposed to the global score S.

2. In the backward phase, we could also process the edges in batches. As we explore in Section 5, in certain cases, this allows us to prove sample complexity upper bounds.

Computational complexity The running time of GFBS is a polynomial in d and the time needed to compute the score $S(\cdot)$. More specifically, GFBS requires $O(d^2)$ score evaluations (compared to $O(d2^d)$ for exact algorithms). Evidently, a key computational concern is the complexity of evaluating the score in the first place. For many models such as linear, generalized linear, and exponential family models, this computation can be carried out in $\text{poly}(n, d)$ time, which implies that GFBS on the whole runs in polynomial time. For nonparametric models, this computation may no longer be polynomial-time, but the total number of score evaluations is still $O(d^2)$. In particular, GFBS always enjoys an exponential speedup over exact algorithms.

Comparison to GES In the supplement (see Appendix A), we exhibit linear Gaussian SEMs and illustrate how GES differs from GFBS for the least squares score as well as the traditional Gaussian BIC score. We first examine a folklore model where we show that their outputs sometimes differ. We also exhibit a model where they always differ. The key takeaway is that GFBS really is a distinct algorithm from GES.

3.1 Connection to equal variance SEM

An important line of work starting with [17] has shown that the assumption of equal variances in a linear Gaussian SEM [32] leads directly to an efficient, order-based algorithm. A similar idea in the

setting of so-called quadratic variance function (QVF) DAGs was explored in [30]. In this section, we show that the equal-variance algorithm of [5], Algorithm 1, is a special case of GFBS.

Define a score function as follows:

$$S_{\mathrm{LS}}(W) = \sum_{i=1}^{d} \mathbb{E}\,\mathrm{var}(X_i \mid \mathrm{pa}_W(i)). \tag{2}$$

A few comments on this score function are in order:

1. The only assumption needed on X for this score to be well-defined is that $\mathbb{E}XX^T$ is well-defined, i.e. $X_i \in L^2$ for each i.
2. When X satisfies a linear structural equation model $X = W^T X + z$, minimizing S_{LS} is equivalent to minimizing the least-squares loss $\sum_{i=1}^{d}(X_i - \langle W^{(i)}, X\rangle)^2$. Loh and Bühlmann [23] have shown that when $\mathrm{cov}(z) = \sigma^2 I$, the unique global minimizer of the least-squares loss is the so-called *equal variance SEM*.
3. More generally, for nonlinear models, we have

$$S_{\mathrm{LS}}(W) = \min_{g_1,\ldots,g_d \sim W} \sum_{i=1}^{d} \mathbb{E}(X_i - g_i(X))^2, \tag{3}$$

   where $g_1, \ldots, g_d \sim W$ indicates that for each i, g_i depends only on the variables in $\mathrm{pa}_W(i)$. In other words, the minimum is taken over all functions $g_1, \ldots, g_d$ that respect the dependency structure implied by W. In this case, g_i is essentially $\mathbb{E}[X_i \mid \mathrm{pa}_W(i)]$.
4. We can use (3) to define an empirical score in the obvious way given i.i.d. samples. Alternatively, the residual variance $\mathbb{E}\,\mathrm{var}(X_i \mid \mathrm{pa}_W(i))$ can be replaced with any estimator of the residual variance.

The GFBS algorithm consists of two phases: A forward phase and a backward phase. Our claim is that the forward phase of GFBS is identical to the equal-variance algorithm from [5]:

Proposition 3.1. *After the forward phase, the ordering T returned by GFBS (Algorithm 1) is the same as the ordering returned by the top-down equal-variance algorithm from Chen et al. [5].*

Corollary 3.2. *Assume the linear SEM $X = W^T X + z$ with $\mathrm{cov}(z) = \sigma^2 I$ under the score function* (2)*. Then GFBS returns a global minimizer of the problem* (1)*.*

Proposition 3.1 will immediately follow from a more general statement which we prove in Theorem 4.6. An intriguing question is to what extent this observation extends to *nonlinear* models such as additive noise models: While we do not have a proof, our experiments in Section 6 suggest something along these lines is true.

4 Bregman scores and identifiability via Bregman information

Motivated by the connection between GFBS, global optimality, and the least squares loss, in this section we establish a nice connection between the greedy algorithm and exponential families via the well-known duality between Bregman divergences and partition functions in exponential families [2]. This can then be used to prove identifiability and recovery guarantees for GFBS.

Bregman divergences and information Let $\phi : \mathbb{R} \to \mathbb{R}$ be a strictly convex, differentiable function. Let $d_\phi(x, y) = \phi(x) - \phi(y) - (x - y)\phi'(y)$ be the Bregman divergence associated with ϕ and let $I_\phi(\mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}[d_\phi(x, \mu)]$ be the associated Bregman information. The Bregman-divergence is a general notion of distance that generalizes squared Euclidean distance, logistic loss, Itakuro-Saito distance, KL-divergence, Mahalanobis distance and generalized I-divergences, among others [2]. The Bregman-information of a distribution is a measure of randomness of the distribution, that's associated with ϕ. Among others, it generalizes the variance, the mutual-information and the Jensen-Shannon divergence of Gaussian processes [2]. See Appendix B for a brief review of this material and a basic treatment of Legendre duality, which will be used in the next section.

4.1 Bregman score functions, duality, and exponential families

By replacing the least squares loss in (1) with a Bregman divergence d_ϕ, we obtain the following score function, which we call a *Bregman score*:

$$S_\phi(W) = \sum_i \mathbb{E}_X d_\phi(X_i, \mathbb{E}[X_i \mid \mathrm{pa}_W(i)]) = \sum_i \min_{g_1,\ldots,g_d \sim W} \mathbb{E}_X d_\phi(X_i, g_i(\mathrm{pa}_W(i))) \tag{4}$$

Before we study the behaviour of GFBS on Bregman scores, it is worth taking a moment to interpret this score function. To accomplish this, let us define the notion of an exponential random family DAG:

Definition 4.1. *A DAG W and a distribution $\mathcal{D}$ define an exponential random family (ERF) DAG if (a) $\mathcal{D}$ is Markov with respect to W, and (b) The local conditional probabilities come from an exponential family, i.e. $\mathbb{P}(X_i \mid \mathrm{pa}_W(i)) \sim \mathrm{ERF}(g_i, \psi_i)$, where ψ_i is the log-partition function of an exponential family with mean function $g_i(\mathrm{pa}_W(i))$.*

Since $\mathrm{ERF}(g_i, \psi_i)$ parametrizes a conditional distribution, its mean parameter g_i is a function instead of vector, which explains our choice of notation. By the Markov property, any choice of local exponential family $\mathrm{ERF}(g_i, \psi_i)$ gives a well-defined joint distribution. The following lemma makes explicit the relationship between Bregman scores, exponential family DAGs, and the Bregman information. Let ϕ^* denote the Legendre dual of ϕ.

Lemma 4.2. *Let ϕ be a strictly convex, differentiable function and let $\psi := \phi^*$. Then*

$$S_\phi(W) = \sum_{i \le d} \mathbb{E}[I_\phi(X_i | \mathrm{pa}_W(i))] = -\sum_{i \le d} \mathbb{E}_X \log p_{g_i,\psi}(X_i \mid \mathrm{pa}_W(i)) - C(X) \tag{5}$$

where $C(X)$ depends only on X and not the underlying DAG W and $p_{g_i,\psi}$ is the density of an $\mathrm{ERF}(g_i, \psi)$ model.

The proof of this lemma, which can be found in Appendix C, follows from the well-known correspondence between Bregman divergences and exponential families, given by the dual map $\phi \mapsto \phi^*$: Given a Bregman divergence ϕ, there is a corresponding exponential family whose log-partition function is given by ϕ^* [2] and vice versa.

Importantly, Lemma 4.2 shows that the Bregman score S_ϕ is equivalent to the expected negative log-likelihood of an exponential family DAG whose local conditional probabilities all have the same log-partition function ψ. This means that minimizing the Bregman score can be naturally associated to maximizing the expected log likelihood of such a model. Similar observations had also been made and used in prior works on PCA [10], clustering [2] and learning theory [14].

4.2 Identifiability via Bregman information

Motivated by the connection between exponential family DAGs with the same local log-partition maps, in this section, we state our main assumption that generalizes the equal variance assumptions from prior works.

First, we will need a mild assumption on W that's of similar flavor to causal minimality, but with respect to the Bregman-information we are looking at. This is a natural assumption that will hold in most settings. Denote $\mathcal{A}_W(i)$ to be the non-descendants of i in the graph W,

Assumption 4.3. *For all $i \le d$ and all subsets $Y \subseteq \mathcal{A}_W(i)$ such that $\mathrm{pa}(i) \not\subseteq Y$, $\mathbb{E}[I_\phi(X_i|Y)] > \mathbb{E}[I_\phi(X_i| \mathrm{pa}(i))]$.*

This assumption essentially asserts that no edge in W is superfluous with respect to the distribution on X. Now, we state our main assumption.

Assumption 4.4 (Equal Bregman-information upon conditioning)**.** *Assume that for a constant $\tau > 0$,*

$$\mathbb{E}[I_\phi(X_i| \mathrm{pa}(i))] = \mathbb{E}_w[I_\phi(X_i| \mathrm{pa}(i) = w)] = \tau \text{ for all } i \le n$$

where $\mathrm{pa}(i)$ are the parents of i in the underlying DAG W.

Example 4.5 (Special case of ANMs)**.** *Suppose we are working with an ANM. That is, there is a DAG W such that for all $i \le d$, $X_i = f_i(\mathrm{pa}(i)) + \epsilon_i$ for some function f_i, where ϵ_i are jointly independent noise variables. Then, the above assumption says that there is a constant $\tau \ge 0$ such that for all i, $I_\phi(\epsilon_i) = \tau$. When $\phi(x) = x^2$, this is the well-known equal variance assumption.*

We are now ready to state our main theorem.

Theorem 4.6. *Consider a model satisfying Assumption 4.3 and Assumption 4.4. Under the Bregman score $S_\phi(W)$, the GFBS algorithm with tolerance parameter $\gamma = 0$ will output the true model.*

As stated, the theorem holds for the population setting. The case of finite samples is studied in detail in Section 5, where we prove the same result given sufficient samples.

Corollary 4.7. *A model satisfying Assumption 4.3 and the Equal Bregman-information Assumption 4.4 is identifiable.*

We defer the proof of the main theorem to the supplement, where we prove it for an even more general class of functionals that subsume the Bregman-information. Here, we make the following remarks regarding this proof.

1. The proof is actually shown for general functionals for which "conditioning drops value". Therefore, we don't need to only work with Bregman-information and we can instead work with many uncertainty measures of distributions that have this property. This is useful, for example, to show that non-Bregman-type models such as the QVF model from [30] are identifiable using our framework.
2. A similar proof could be adapted for other functionals of distributions that measure the randomness or uncertainty of the distribution. One class of examples could be generalized entropies [1] such as the Shannon entropy, Rényi entropy or the Tsallis entropy. We leave this for future work.

Remark 4.8. *An important reason why our algorithm is efficient is because in line 4 of Algorithm 1, we only compute a single score for each vertex not in the ordering so far. This works especially nicely with the Bregman score, precisely because conditioning with respect to more variables only lowers the Bregman information of a variable, as is exploited to prove the theorems above.*

A natural score function for non-parametric multiplicative models We study multiplicative noise models of the form $X_i = f(\mathrm{pa}(i))\epsilon_i$ from the perspective of the framework built so far. Examples of such models include growth models from economics and biology [25]. More specifically, we choose $\phi(x) = -\log x$ for which the Bregman divergence d_ϕ is the *Itakuro-Saito* distance commonly used in the Signal and Speech processing community. The associated Bregman score is the *Itakuro-Saito* score given by

$$S_\phi(W) = \sum_{i \le d} (\mathbb{E} \log \mathbb{E}[X_i | \, \mathrm{pa}(i)] - \mathbb{E}[\log X_i]).$$

Interestingly, the equal Bregman-information assumption reduces purely to an assumption about the noise variables, akin to the equal variance assumption in the case of additive noise models. This suggests that for multiplicative models, the Itakuro-Saito score is naturally motivated from the perspective of identifiability. This gives a new insight into the applicability of score-based learning for multiplicative models, with theoretical foundations in our analysis. For details, see Appendix E.

5 Sample complexity

To derive a sample complexity bound for GFBS, we first need to compute the Bregman score S_ϕ; due to decomposability and (5), this reduces to estimating the Bregman information I_ϕ. Let the samples be denoted $(X_1^{(t)}, X_2^{(t)}, \ldots, X_d^{(t)})$ for $t = 1, 2, \ldots, n$. Denote the Bregman information of X_i conditioning on a set A with conditional mean plugged in as (after some calculation)

$$S(X_i \mid A) := \mathbb{E}[I_\phi(X_i \mid A)] = \mathbb{E}\phi(X_i) - \mathbb{E}\phi(\mathbb{E}(X_i \mid A)) \tag{6}$$

for some strictly convex, differentiable function ϕ. To estimate this quantity, we can first apply nonparametric regression to estimate $f_{iA} := \mathbb{E}(X_i \mid A)$ and then take the sample mean:

$$\widehat{S}(X_i \mid A) = \frac{1}{n} \sum_{t \le n} \phi\big(X_i^{(t)}\big) - \frac{1}{n} \sum_{t \le n} \phi\big(\widehat{f}_{iA}(A^{(t)})\big). \tag{7}$$

To show convergence rate of this estimator, we will need some some regularity conditions on f_{iA} and ϕ. First, we recall the definition of the Hölder class of functions:

Definition 5.1. *For any $r = (r_1, \cdots, r_d)$, $r_i \in \mathbb{N}$, let $|r| = \sum_i r_i$ and $D^r = \frac{\partial^{|r|}}{\partial x_1^{r_1} \cdots \partial x_d^{r_d}}$. The Hölder class $\Sigma(s, L)$ is the set of functions satisfying*

$$|D^r f(x) - D^r f(y)| \leq L \,|\, x - y \,|^{\,s-r}$$

for all r such that $|r| \leq s$ and $x, y \in \mathbb{R}^d$.

Assumption 5.2. *Suppose for all i and ancestor sets A of i, $f_{iA} \in \Sigma(s, L)$. And suppose $\phi(X_i)$, $\phi(f_{iA})$ and $\phi'(f_{iA})$ all have finite second moments.*

Denote $\mathcal{A}_W(i)$ to be the non-descendants of i in graph W, then the following lemma says that we have a uniform estimator for the Bregman score:

Lemma 5.3. *Suppose the Bregman score and the conditional expectations satisfy Assumption 5.2. Using the estimator defined in* (7) *yields*

$$\min_{i \in [d], A \subseteq \mathcal{A}_W(i)} \mathbb{P}\left(\left| \widehat{S}(X_i \,|\, A) - S(X_i \,|\, A) \right| \leq t \right) \geq 1 - \frac{\delta_n^2}{t^2}$$

where $\delta_n^2 = C(n^{\frac{-2s}{2s+d}} + n^{-1})$ for some constant C.

Using this estimator, we can bound the sample complexity of the forward pass of GFBS as follows:

Theorem 5.4 (Forward phase of GFBS)**.** *Suppose the BN satisfies the identifiability condition in Theorem 4.6 and assumptions in Lemma 5.3, denote the gap*

$$\Delta := \min_{\substack{i \in [d], A \subseteq \mathcal{A}_W(i) \\ \mathrm{pa}(i) \not\subseteq A}} S(X_i \,|\, A) - \tau > 0$$

Let the ordering returned by the first phase of GFBS to be $\widehat{\pi} = (\widehat{\pi}_1, \cdots, \widehat{\pi}_d)$. If the sample size

$$n \gtrsim \left(\frac{d^2}{\Delta^2 \epsilon} \right)^{\frac{2s+d}{2s} \vee 1}$$

then $\mathbb{P}(\widehat{\pi}$ is a valid ordering$) \geq 1 - \epsilon$.

The causal minimality Assumption 4.3 is equivalent to stating $\Delta > 0$. Theorem 5.4 shows that Δ in fact controls the hardness of the estimation, which is the gap between the minimum Bregman information when all parents are conditioned on and when some parents are missing.

In this section, to obtain strong bounds on sample complexity, we modify the backward phase of GFBS to be as follows:

Definition 5.5. . *For each $\widehat{\pi}_{j+1}$, we find it's parents from $\widehat{A}_j$ in the following way, estimate $S(X_{\widehat{\pi}_{j+1}} \,|\, \widehat{A}_j)$ and $S(X_{\widehat{\pi}_{j+1}} \,|\, \widehat{A}_j \setminus i)$ for $i \in \widehat{A}_j$. Then, set*

$$\widehat{\mathrm{pa}}(\widehat{\pi}_{j+1}) = \widehat{A}_j \setminus \left\{ i \in \widehat{A}_j \,\middle|\, |\widehat{S}(X_{\widehat{\pi}_{j+1}} \,|\, \widehat{A}_j) - \widehat{S}(X_{\widehat{\pi}_{j+1}} \,|\, \widehat{A}_j \setminus i)| \leq \gamma \right\}. \tag{8}$$

This says that we keep an edge $(i, \widehat{\pi}_{j+1})$ depending on it's influence on the local score at the vertex $\widehat{\pi}_{j+1}$. If the influence is low, then we discard that edge. For our analysis to work, we process these low-influence edges in batches grouped according to the vertices they are oriented towards.

Theorem 5.6 (Backward phase of GFBS)**.** *Suppose the same conditions and sample size in Theorem 5.4 holds, using the backward phase defined in* (8) *with $\gamma = \Delta/2$ guarantees $\mathbb{P}(\widehat{W} = W) \geq 1 - \epsilon$.*

Proofs can be found in Appendix F in the supplement.

6 Experiments

We conduct experiments to show the performance of GFBS on optimizing the Bregman score. We compare GFBS with existing score-based DAG learning algorithms: Gobnilp [11], NOTEARS [49], and GDS [32]. The implementation of these algorithms and data generating process are detailed in Appendix H. Although previous works have evaluated the structure learning performance of special cases of GFBS such as equal variances, we also include these comparisons in the appendix for completeness. Also, in Appendix G, we investigate the performance of GFBS on models which violate the identifiability Assumption 4.4.

- *Choice of* ϕ. To show the generality of the Bregman score (4), we investigate two convex functions to define the score: $\phi_1(x) = x^2$ and $\phi_2(x) = -\log x$. They correspond to sum of residual variances and sum of residual Itakuro-Saito (IS) distances respectively.
- *Graph type*. We generate three types of graphs: Markov chains (MC), Erdös-Rényi (ER) graphs, Scale-Free (SF) graphs with different expected number of edges. We let the expected number of edges scale with d, e.g. ER-2 stands for Erdös-Rényi with $2d$ edges.
- *Model type*. We simulate the data as $X_i = f_i(\mathrm{pa}(i)) + Z_i$ or $X_i = f_i(\mathrm{pa}(i)) \times Z_i$ for different ϕ's, where Z_i is independently sampled from some distribution such that Assumption 4.4 is satisfied. Then we consider the following forms of the parental functions f_i: linear (LIN), sine (SIN), additive Gaussian process (AGP), and non-additive Gaussian process (NGP).

The main objective of these experiments is to evaluate the performance of these algorithms in optimizing the score: For this, it is necessary to compute the globally optimal score as a benchmark, which is computationally intensive. For this, we use Gobnilp [11]. As a result, our experiments are restricted to $d = 5$. The results are shown in Figure 1. As expected, GFBS returns a near-globally optimal solution in most cases when the sample size is large. Due to finite-sample errors, in some cases (notably on the IS score), GFBS returns a slightly higher score due to the backward phase, which allows the score to increase slightly in favour of sparser solutions. At a technical level, the issue is that the score does not distinguish I-maps from minimal I-maps, and this is exacerbated on finite samples. Better regularization and parameter tuning should resolve this, which we leave to future work. Nonetheless, the close alignment between GFBS and the globally optimal score suggest that GFBS—and hence the equal variance algorithm—is truly minimizing the score.

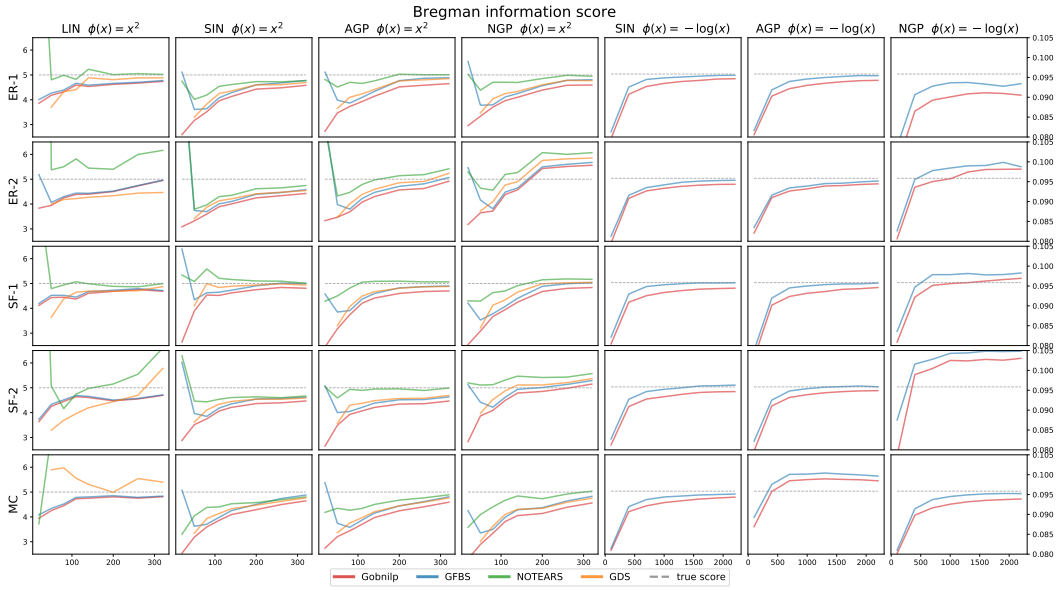

Figure 1: Score of output DAG vs. sample size n for GFBS and 3 other algorithms. Left four columns: $\phi_1(x) = x^2$ and Z_i is t-distribution with variance 1; Right three columns: $\phi_2(x) = -\log(x)$ and Z_i is uniform distribution in $[1, 2]$. The two sets of columns have different Y-axis scales. The grey dashed line is the score of the true graph.

Broader impacts

Learning graphical models has important applications in causal inference, which is useful for mitigating bias in ML models. At the same time, causal models can be easily misinterpreted and provide a false sense of security, especially when they are subject to finite-sample errors. One additional potential negative impact from this line of work is the environmental cost of training large causal models, which can be expensive and time-consuming.

References

[1] J. M. Amigó, S. G. Balogh, and S. Hernández. A brief review of generalized entropies. *Entropy*, 20(11):813, 2018.

[2] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty. Clustering with bregman divergences. *Journal of machine learning research*, 6(10), 2005.

[3] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

[4] G. Bresler. Efficiently learning ising models on arbitrary graphs. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 771–782, 2015.

[5] W. Chen, M. Drton, and Y. S. Wang. On causal discovery with an equal-variance assumption. *Biometrika*, 106(4):973–980, 09 2019. ISSN 0006-3444. doi: 10.1093/biomet/asz049. URL `https://doi.org/10.1093/biomet/asz049`.

[6] D. Chickering, D. Geiger, and D. Heckerman. Learning bayesian networks: Search methods and experimental results. In *proceedings of fifth conference on artificial intelligence and statistics*, pages 112–128, 1995.

[7] D. M. Chickering. Learning Bayesian networks is NP-complete. In *Learning from data*, pages 121–130. Springer, 1996.

[8] D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2003.

[9] D. M. Chickering, D. Heckerman, and C. Meek. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.

[10] M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal components analysis to the exponential family. In *Nips*, volume 13, page 23, 2001.

[11] J. Cussens. Bayesian network learning with cutting planes. *arXiv preprint arXiv:1202.3713*, 2012.

[12] J. Cussens, D. Haws, and M. Studenỳ. Polyhedral aspects of score equivalence in bayesian network structure learning. *Mathematical Programming*, 164(1-2):285–324, 2017.

[13] S. Dasgupta. Learning polytrees. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 134–141, 1999.

[14] J. Forster and M. K. Warmuth. Relative expected instantaneous loss bounds. *Journal of Computer and System Sciences*, 64(1):76–102, 2002.

[15] M. Gao, Y. Ding, and B. Aragam. A polynomial-time algorithm for learning nonparametric causal graphs. *arXiv preprint arXiv:2006.11970*, 2020.

[16] D. Geiger, D. Heckerman, H. King, and C. Meek. Stratified exponential families: Graphical models and model selection. *Annals of Statistics*, pages 505–529, 2001.

[17] A. Ghoshal and J. Honorio. Learning identifiable gaussian bayesian networks in polynomial time and sample complexity. In *Advances in Neural Information Processing Systems 30*, pages 6457–6466. 2017.

[18] A. Ghoshal and J. Honorio. Learning linear structural equation models in polynomial time and sample complexity. In A. Storkey and F. Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1466–1475, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

[19] C. Glymour, K. Zhang, and P. Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

[20] A. Jalali, C. Johnson, and P. Ravikumar. On learning discrete graphical models using greedy methods. *arXiv preprint arXiv:1107.3258*, 2011.

[21] C. Johnson, A. Jalali, and P. Ravikumar. High-dimensional sparse inverse covariance estimation using greedy methods. In *Artificial Intelligence and Statistics*, pages 574–582. PMLR, 2012.

[22] J. Liu, J. Ye, and R. Fujimaki. Forward-backward greedy algorithms for general convex smooth functions over a cardinality constraint. In *International Conference on Machine Learning*, pages 503–511. PMLR, 2014.

[23] P.-L. Loh and P. Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *Journal of Machine Learning Research*, 15:3065–3105, 2014.

[24] A. Y. Lokhov, M. Vuffray, S. Misra, and M. Chertkov. Optimal structure and parameter learning of ising models. *Science advances*, 4(3):e1700791, 2018.

[25] A. W. Marshall and I. Olkin. *Life distributions*, volume 13. Springer, 2007.

[26] S. Misra, M. Vuffray, and A. Y. Lokhov. Information theoretic optimal learning of gaussian graphical models. In *Conference on Learning Theory*, pages 2888–2909. PMLR, 2020.

[27] P. Nandy, A. Hauser, and M. H. Maathuis. High-dimensional consistency in score-based and hybrid structure learning. *arXiv preprint arXiv:1507.02608*, 2018.

[28] S. Ott and S. Miyano. Finding optimal gene networks using biological constraints. *Genome Informatics*, 14:124–133, 2003.

[29] S. Ott, S. Imoto, and S. Miyano. Finding optimal models for small gene networks. In *Pacific symposium on biocomputing*, volume 9, pages 557–567. Citeseer, 2004.

[30] G. Park and G. Raskutti. Learning quadratic variance function (QVF) dag models via overdispersion scoring (ODS). *The Journal of Machine Learning Research*, 18(1):8300–8342, 2017.

[31] E. Perrier, S. Imoto, and S. Miyano. Finding optimal bayesian network given a super-structure. *Journal of Machine Learning Research*, 9(Oct):2251–2286, 2008.

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

[33] J. Peters and P. Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2014.

[34] J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.

[35] J. Ramsey, M. Glymour, R. Sanchez-Romero, and C. Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, pages 1–9, 2016.

[36] N. Rao, P. Shah, and S. Wright. Forward–backward greedy algorithms for atomic norm regularization. *IEEE Transactions on Signal Processing*, 63(21):5798–5811, 2015.

[37] T. Silander and P. Myllymaki. A simple approach for finding the globally optimal bayesian network structure. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, 2006.

[38] A. P. Singh and A. W. Moore. Finding optimal bayesian networks by dynamic programming. 2005.

[39] P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72, 1991.

[40] P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search*, volume 81. The MIT Press, 2000.

[41] N. Srebro. Maximum likelihood bounded tree-width markov networks. *Artificial intelligence*, 143(1):123–138, 2003.

[42] M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning bayesian networks. In *Uncertainty in Artifical Intelligence (UAI)*, 2005.

[43] L. Tian, P. Xu, and Q. Gu. Forward backward greedy algorithms for multi-task learning with faster rates. In *UAI*, 2016.

[44] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

[45] M. Vuffray, S. Misra, A. Y. Lokhov, and M. Chertkov. Interaction screening: Efficient and sample-optimal learning of ising models. *arXiv preprint arXiv:1605.07252*, 2016.

[46] C. Yuan and B. Malone. Learning optimal Bayesian networks: A shortest path perspective. *J. Artif. Intell. Res.(JAIR)*, 48:23–65, 2013.

[47] T. Zhang. Adaptive forward-backward greedy algorithm for sparse learning with linear models. *Advances in Neural Information Processing Systems*, 21:1921–1928, 2008.

[48] X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing. Dags with no tears: Continuous optimization for structure learning. *arXiv preprint arXiv:1803.01422*, 2018.

[49] X. Zheng, C. Dan, B. Aragam, P. Ravikumar, and E. Xing. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. PMLR, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In L65-66, L116-117, we clarify that we do not actually prove global optimality except in a special case, and explain how our results provide empirical evidence
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Broader Impacts statement. We expect the possible negative consequences to be minimal, as this submission is theoretical in nature.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See for instance Assumption 4.3, Assumption 4.4, Assumption 5.2
   (b) Did you include complete proofs of all theoretical results? [Yes] See supplementary materials
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Appendix H
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In Section 6
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In Appendix H
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We use synthetic data that we generate

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We use synthetic data that we generate

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]